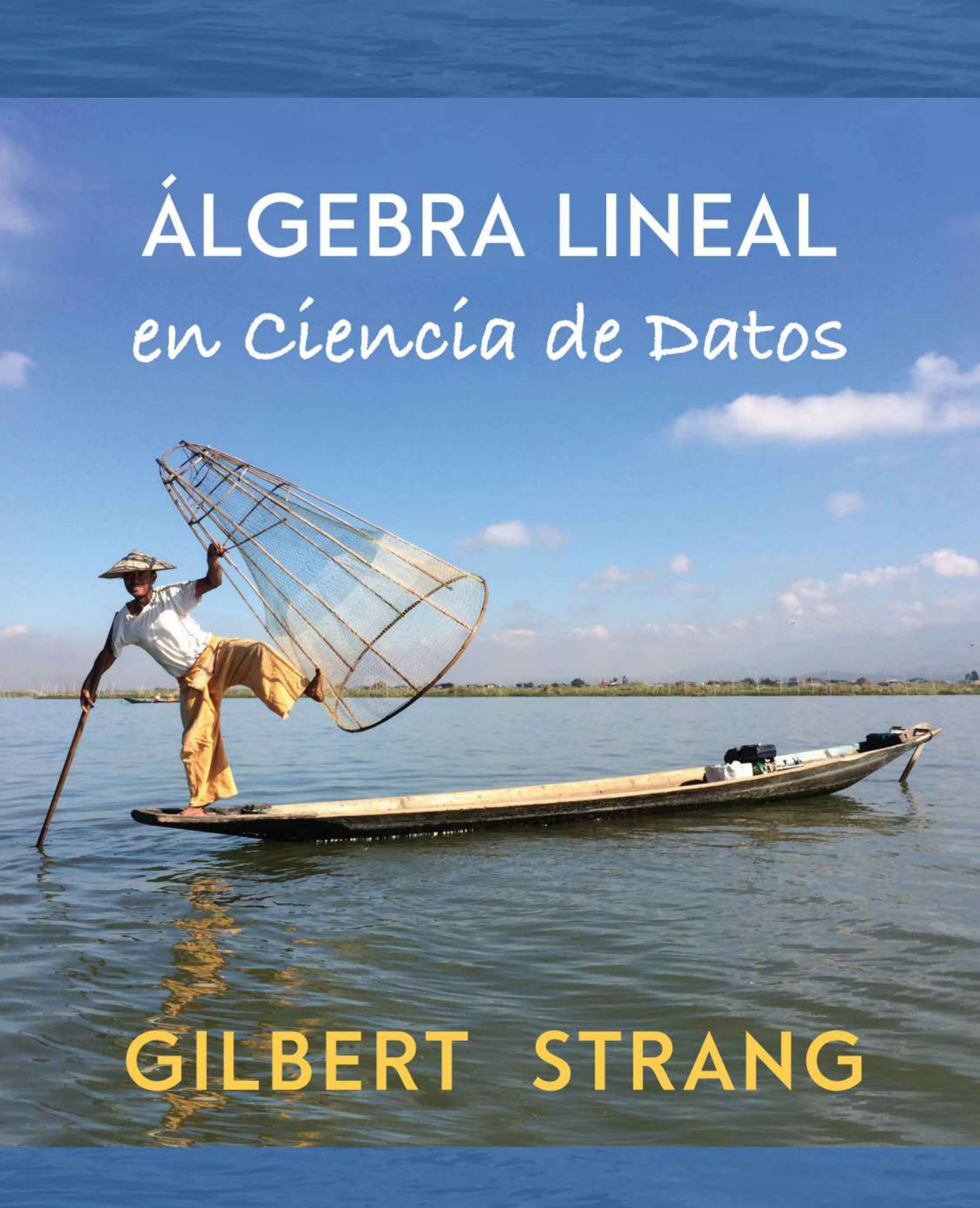
ÁLGEBRA LINEAL

en Ciencia de Datos



GILBERT STRANG

ÁLGEBRA LINEAL EN CIENCIA DE DATOS

GILBERT STRANG

Massachusetts Institute of Technology

Traducido al español por

Liliana Forzani, Cecilia Morin, Pedro Morin

Facultad de Ingeniería Química

Universidad Nacional del Litoral, Santa Fe, Argentina

9 8 7 6 5 4 3 2 1

Todos los derechos reservados. Toda reproducción parcial o total, por cualquier medio mecánico, electrónico u otro, queda prohibida sin el consentimiento escrito de SIAM. Toda traducción queda prohibida salvo expreso consentimiento del autor o de Wellesley - Cambridge Press.

Linear Algebra and Learning from Data ©2019 por Gilbert Strang.
Edición en español ©2022 por Gilbert Strang. Traducido del original en inglés.

ISBN 978-1-61197-738-7

Otros libros de Gilbert Strang

Lecture Notes for Linear Algebra	ISBN 978-1-7331466-4-7
Linear Algebra for Everyone	ISBN 978-1-7331466-3-0
Linear Algebra and Learning from Data	ISBN 978-0-6921963-8-0
Introduction to Linear Algebra, Fifth Edition	ISBN 978-0-9802327-7-6
Computational Science and Engineering	ISBN 978-0-9614088-1-7
Differential Equations and Linear Algebra	ISBN 978-0-9802327-9-0
Calculus, Third Edition	ISBN 978-0-9802327-5-2
Essays in Linear Algebra	ISBN 978-0-9802327-6-9
Algorithms for Global Positioning with Kai Borre	ISBN 978-0-9802327-3-8
An Analysis of the Finite Element Method, Second Edition with George Fix	ISBN 978-0-9802327-8-3
Wavelets and Filter Banks with Truong Nguyen	ISBN 978-0-9614088-7-9
Introduction to Applied Mathematics	ISBN 978-0-9614088-0-0

Wellesley - Cambridge Press
Box 812060
Wellesley MA 02482 USA
www.wellesleycambridge.com

Este libro está disponible como libro electrónico en Google Play.
Para más información:
algebralinealencienciadedatos@gmail.com

El sitio web de la versión original en inglés de este libro es **math.mit.edu/learningfromdata**
El enlace lleva al material del curso 18.065 y videoconferencias en YouTube y OCW

La fotografía de la portada muestra una red neuronal en el Lago Inle. Fue tomada en Birmania.
A partir de esa foto, Lois Sellers diseñó y creó la portada.
La foto de **playground.tensorflow.org** fue un regalo de su creador Daniel Smilkov.

Álgebra lineal está incluida en el sitio del MIT OpenCourseWare **ocw.mit.edu**
Aquí se pueden encontrar las lecciones grabadas del curso de álgebra lineal 18.06 y 18.06 SC

Aprendizaje profundo y redes neuronales

El álgebra lineal, la probabilidad/estadística y la optimización son los pilares matemáticos del aprendizaje automático o maquinal. Esos capítulos serán previos a los de la arquitectura de una red neuronal. Pero nos parece útil comenzar con la siguiente descripción del objetivo : *construir una función que clasifique los datos de entrenamiento correctamente, así puede generalizarse a datos de prueba aún no vistos.*

Para que esa afirmación tenga sentido, necesitamos saber más sobre esta función de aprendizaje. Ese es el propósito de estas tres páginas : dar una guía para todo lo que sigue.

Las variables independientes de la función F son vectores o matrices o a veces tensores: una entrada $\boldsymbol{v}$ para cada muestra de entrenamiento. Para el problema de identificación de dígitos manuscritos, cada entrada será una imagen (una matriz de píxeles). Buscamos clasificar cada una de esas imágenes como un número del 0 al 9. Esos diez números son los resultados posibles de la función de aprendizaje. En este ejemplo, la función F aprende a qué debe prestar atención para clasificar las imágenes.

El conjunto MNIST contiene 70 000 dígitos manuscritos. Entrenamos una función de aprendizaje usando una parte de ese conjunto. Creamos la función asignándole pesos a diferentes píxeles en la imagen. El gran problema de optimización (el corazón del cálculo) es elegir los pesos para que la función seleccione el resultado correcto $0, 1, 2, 3, 4, 5, 6, 7, 8,$ o 9. ¡ Y no pedimos perfección ! (Uno de los peligros del aprendizaje profundo es el *sobreajuste de los datos.*)

Luego validamos la función eligiendo muestras MNIST no utilizadas para el entrenamiento y aplicando la función para clasificar esos datos de prueba. Competiciones a lo largo de los años han conducido a importantes mejoras de los resultados en los datos de pruebas. Actualmente, las redes de convolución menos del 1 % de error. De hecho, son las competencias sobre datos conocidos como MNIST las que produjeron grandes mejoras en la estructura de F. Esa estructura está basada en la arquitectura de una red neuronal subyacente.

Funciones de aprendizaje lineales y no lineales

Las entradas son las muestras $\boldsymbol{v}$, las respuestas son las clasificaciones calculadas $\boldsymbol{w} = F(\boldsymbol{v})$. La función de aprendizaje más simple sería lineal : $\boldsymbol{w} = A\boldsymbol{v}$. Las entradas en la matriz A son los pesos a aprender : no es tan difícil. Con frecuencia la función también aprende un *vector de sesgo* $\boldsymbol{b}$, entonces $F(\boldsymbol{v}) = A\boldsymbol{v} + \boldsymbol{b}$. Esta función es "*afín*". Las funciones afines pueden aprenderse rápido, pero por sí solas son demasiado simples.

Más precisamente, la linealidad es un requerimiento muy limitante. Si el MNIST usara números romanos, entonces II podría estar a mitad de camino entre I y III (como lo exige la linealidad). Pero, ¿ qué es estar a mitad de camino entre I y XIX ? Ciertamente las funciones afines $A\boldsymbol{v}+\boldsymbol{b}$ no siempre son suficientes.

La no linealidad podría obtenerse elevando al cuadrado las componentes del vector de entrada $\boldsymbol{v}$. Ese paso podría ayudar a separar un círculo de un punto en su interior : algo que las funciones lineales no pueden hacer. Pero la construcción de F la tendencia es utilizar las funciones con gráficos en forma de S, llamadas "sigmoides". Es destacable que se logra un gran progreso al insertar estas funciones en forma de S estándares y no lineales entre las matrices A y B produciendo $A(S(B\boldsymbol{v}))$. Eventualmente se descubrió que las funciones logísticas S suavemente curvadas podían ser reemplazadas por la muy sencilla función rampa ahora llamada $\mathbf{ReLU}(\boldsymbol{x}) = \mathbf{max}(\mathbf{0}, \boldsymbol{x})$. Los gráficos de estas *"funciones de activación"* no lineales R se presentan en la Sección VII.1.

Redes neuronales y la estructura de $F(v)$

Las funciones que producen aprendizaje profundo tienen la forma $F(\boldsymbol{v}) = L(R(L(R(\ldots(L\boldsymbol{v})))))$. Esto es una *composición* de funciones afines $L\boldsymbol{v} = A\boldsymbol{v} + \boldsymbol{b}$ con funciones no lineales R, que actúan sobre cada componente del vector $L\boldsymbol{v}$. Las matrices A y los vectores de sesgo $\boldsymbol{b}$ son los **pesos en la función de aprendizaje $\boldsymbol{F}$**. Son las A y los $\boldsymbol{b}$ que deben aprenderse a partir de los datos de entrenamiento, para que las respuestas $F(\boldsymbol{v})$ sean (casi) correctos. Luego F puede aplicarse a nuevas muestras de la misma población. Si los pesos (las A y los $\boldsymbol{b}$) están bien elegidos, las respuestas $F(v)$ correspondientes a los datos de prueba no vistos serán precisos. Más capas en la función F producirán típicamente resultados más precisos en $F(\boldsymbol{v})$.

Hablando correctamente, $F(\boldsymbol{x}, \boldsymbol{v})$ depende de las entradas $\boldsymbol{v}$ y los pesos $\boldsymbol{x}$ (todas las A y los $\boldsymbol{b}$). Las respuestas $\boldsymbol{v_1} = \mathrm{ReLU}(A_1\boldsymbol{v} + \boldsymbol{b_1})$ del primer paso conforman la **primera capa oculta de nuestra red neuronal**. La red completa comienza con la capa de entrada $\boldsymbol{v}$ y termina con la capa de salida $\boldsymbol{w} = F(\boldsymbol{v})$. La parte afín $L_k(\boldsymbol{v}_{k-1}) = A_k\boldsymbol{v}_{k-1} + \boldsymbol{b}_k$ de cada paso usa los pesos calculados A_k y $\boldsymbol{b_k}$.

Todos esos pesos juntos se eligen en la optimización gigante de aprendizaje profundo :

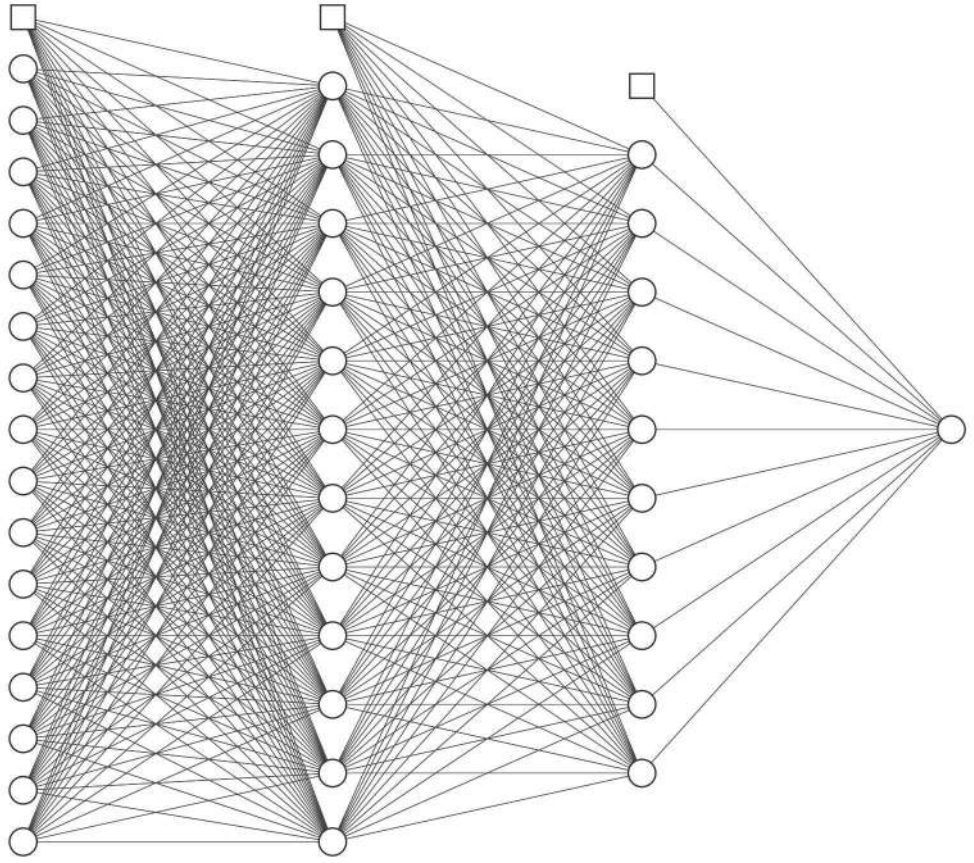
Elegir pesos $\boldsymbol{A_k}$ y $\boldsymbol{b_k}$ que minimicen la pérdida total sobre todas las muestras de entrenamiento.

La pérdida total es la suma de las pérdidas individuales en cada muestra. La función de pérdida de cuadrados mínimos tiene la forma conocida $||F(\boldsymbol{v}) - \text{verdadero valor de salida}||^2$. Generalmente utilizar cuadrados mínimos no da la mejor función de pérdida para aprendizaje profundo.

Una entrada $\boldsymbol{v}$ = **Una salida $\boldsymbol{w} = 2$**

He aquí una imagen de una red neuronal, que muestra la estructura de $F(\boldsymbol{v})$. La capa de entrada contiene las muestras de entrenamiento $\boldsymbol{v} = \boldsymbol{v}_0$. La respuesta es su clasificación $\boldsymbol{w} = F(\boldsymbol{v})$. Para un aprendizaje perfecto, $\boldsymbol{w}$ será un dígito (correcto) del 0 al 9. **Las capas ocultas agregan profundidad a la red**. Es esa profundidad la que permite que la función compuesta F sea tan existosa en el aprendizaje profundo. De hecho, la cantidad de pesos A_{ij} y b_j en la red neuronal suele ser mayor a la cantidad de muestras de entrenamiento $\boldsymbol{v}$.

Esta es una red pre-alimentada totalmente conectada. Para las imágenes, una red neuronal *de convolución* (**CNN**, por su sigla en inglés) suele ser apropiada y los pesos suelen estar compartidos : las diagonales de las matrices A son constantes. El aprendizaje profundo funciona increíblemente bien, cuando la arquitectura es correcta.



Muestra de entrada Capa oculta Capa oculta Salida

Cada diagonal en esta red neuronal representa un peso a aprender vía optimización. Las aristas que salen de los cuadrados contienen vectores de sesgo $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3$. Los demás pesos están en A_1, A_2, A_3.

Prefacio y agradecimientos

Mi más profundo agradecimiento al profesor Raj Rao Nadakuditi de la Universidad de Michigan. En su año sabático en 2017, Raj trajo su curso EECS 551 al MIT. Voló hasta Boston todas las semanas para enseñar el curso 18.065. Gracias a Raj, los estudiantes pudieron tomar un curso nuevo. Dictaba las clases con computadoras, asignaba tareas y los exámenes estaban prohibidos.

Esto era Álgebra lineal para señales y datos, y estaba viva. Se inscribieron 140 estudiantes del MIT. Alan Edelman enseñó el poderoso lenguaje *Julia*, y yo expliqué los cuatro subespacios fundamentales y la descomposición en valores singulares. Las clases de laboratorio de Michigan involucraban al rango y a la SVD y aplicaciones. Le pedíamos a la clase *razonamiento computacional.*

Ese curso funcionó, incluso la primera vez. No tocó un tema importante : **aprendizaje profundo**. Con esto me refiero a la emoción de crear una función de aprendizaje en una red neuronal, con capas ocultas y las funciones de activación no lineales que las hacen tan poderosas. El sistema se entrena a sí mismo con datos que han sido correctamente clasificados de antemano. La optimización de los pesos revela características importantes como la forma de una letra, los bordes de una imagen, la sintaxis de una oración y los detalles que identifican una señal. Esas características reciben los pesos más grandes *sin sobreajustar los datos* y aprendiéndolo todo. Así, los datos de prueba no vistos de una población similar pueden ser identificados por tener las mismas características.

Los algoritmos que hacen todo esto están mejorando constantemente. Mejor dicho, los *están mejorando*. Esta es la contribución de los informáticos, ingenieros, biólogos, lingüistas, matemáticos y, sobre todo, de los optimizadores: quienes pueden optimizar los pesos para minimizar los errores y también quienes creen que el aprendizaje profundo puede mejorar nuestras vidas.

Este libro fue escrito por las siguientes razones :

1. Para organizar métodos e ideas centrales de la **ciencia de datos**.
2. Para ver cómo el lenguaje del **álgebra lineal** expresa estas ideas.
3. Sobre todo, para mostrar cómo **explicar y enseñar** esas ideas, a uno mismo o a una clase.

Sin duda aprendí que los proyectos son mucho mejores que los exámenes. Los estudiantes hacen sus propias preguntas y escriben sus propios programas. De ahora en más, **¡ proyectos !**

Álgebra lineal y cálculo

El lector ya conocerá las dos áreas centrales de la matemática universitaria : álgebra lineal y cálculo. Para aprendizaje profundo, es el álgebra lineal lo que más importa. Calculamos "pesos" que eligen las características más importantes de los datos de entrenamiento y esos pesos van en matrices. La forma de la función de aprendizaje está descripta en la página iv. Luego, el cálculo nos indica en qué *dirección movernos*, para poder mejorar los pesos $\boldsymbol{x}_k$.

Del cálculo, necesitamos las derivadas parciales (y no las integrales) :

$$\textbf{Reducir el error } \boldsymbol{L(x)} \textbf{ desplazándose desde } \boldsymbol{x_k} \textbf{ a } \boldsymbol{x_{k+1} = x_k - s_k \nabla L.}$$

Ese símbolo $\boldsymbol{\nabla L}$ representa las primeras derivadas de $L(\boldsymbol{x})$. Como está el signo menos, $\boldsymbol{x}_{k+1}$ *desciende desde* $\boldsymbol{x}_k$ en el gráfico de $L(\boldsymbol{x})$. El tamaño del paso $\boldsymbol{s_k}$ (también llamado velocidad de aprendizaje) decide qué tan lejos moverse. Se visualiza cuál es la idea básica : reducir la función de pérdida $L(\boldsymbol{x})$ moviéndonos en la dirección de mayor decrecimiento. $\boldsymbol{\nabla L = 0}$ en los mejores pesos $\boldsymbol{x}^*$.

La complicación es que el vector $\boldsymbol{x}$ representa miles de pesos. Entonces debemos calcular miles de derivadas parciales de L. Y L es una función complicada que depende de varias capas de $\boldsymbol{x}$, como así también de los datos. Así que necesitamos la regla de la cadena para calcular $\boldsymbol{\nabla L}$.

La presentación del Capítulo VI recordará resultados esenciales del cálculo multivariable.

Por contraste, *el álgebra lineal está por todas partes en el mundo del aprendizaje a partir de datos*. ¡ Este es el tema a saber ! Los primeros capítulos de este libro son esencialmente un curso de álgebra lineal aplicada, la teoría básica y su uso para realizar cálculos. Puedo intentar describir cómo ese enfoque (de las ideas que necesitamos) se compara con cursos de álgebra lineal anteriores. Son muy diferentes, lo que significa que hay cosas muy útiles para aprender.

Curso básico

1. Eliminación para resolver $\boldsymbol{Ax = b}$
2. Operaciones matriciales, inversas y determinantes
3. Espacios de vectores y subespacios
4. Independencia, dimensión, rango de una matriz
5. Autovalores y autovectores

Si un curso consiste principalmente en aprender definiciones, eso no es álgebra lineal en acción. Un curso más avanzado hace uso del álgebra. Las definiciones tienen un propósito, y también este libro.

Curso más avanzado

1. $\boldsymbol{Ax = b}$ en todos los casos : sistema cuadrado y sistemas rectangulares.
2. Factorización de A en $\boldsymbol{LU}$, en $\boldsymbol{QR}$, en $\boldsymbol{U\Sigma V^T}$ y en $\boldsymbol{CMR}$: *columnas por filas*.
3. *Los cuatro subespacios fundamentales* : dimensiones, ortogonalidad y buenas bases.
4. Diagonalización de A por autovectores y por vectores singulares a izquierda y a derecha.
5. Aplicaciones : grafos, convoluciones, iteraciones, covarianzas, proyecciones, filtros, redes, imágenes, matrices de datos.

El álgebra lineal se encuentra en el centro del aprendizaje automático o maquinal y debemos estar allí.

Se necesitaba un libro para el curso 18.065. Se inició durante el primer dictado en 2017, y una primera versión salió para el dictado en 2018. Reconozco felizmente que este libro le debe su existencia a Ashley C. Fernandes. Ashley recibe páginas escaneadas desde Boston y envía nuevas secciones desde Mumbai, listo para más trabajo. Este es nuestro séptimo libro juntos y estoy muy agradecido.

Los estudiantes fueron generosos en ayudar con ambas clases, especialmente William Loucks, Claire Khodadad, Alex LeNail y Jack Strang. El proyecto de Alex dio lugar a su código online **alexlenail.me/NN−SVG/** para dibujar redes neuronales (un ejemplo aparece en la página v). El proyecto de Jack en **http://www.teachyourmachine.com** aprende a reconocer números y letras escritas a mano por el usuario : disponible para experimentar. Ver Sección VII.2.

El cuerpo docente y el personal del MIT han brindado una ayuda generosa y muy necesitada :

Suvrit Sra dio una fantástica conferencia sobre el método del descenso por gradiente estocástico (ahora un video en 18.065)

Alex Postnikov explicó cuándo la completación de una matriz puede resultar en una de rango uno (Sección IV.8)

Tommy Poggio mostró a su clase cómo el aprendizaje profundo se generaliza a nuevos datos

Jonathan Harmon, Tom Mullaly y Liang Wang contribuyeron en este libro todos los días

Las ideas llegaron de todos lados y poco a poco fueron completando este libro.

Los contenidos de este libro

Este libro busca explicar las matemáticas de las que depende la ciencia de datos : *álgebra lineal, optimización, probabilidad y estadística*. Los pesos en la función de aprendizaje van en matrices. Esos pesos se optimizan con el método de "descenso por gradiente estocástico". Esa palabra, estocástico, (= aleatorio) es una señal de que el éxito depende de la probabilidad y no de la certeza. La ley de los grandes números se extiende a la ley de las grandes funciones : si la arquitectura está bien diseñada y los parámetros están bien calculados, hay altas probabilidades de éxito.

Tenga en cuenta que este no es un libro sobre cómo realizar cálculos, o codificación, o software. Muchos libros desarrollan bien esas áreas. Uno de nuestros favoritos es *Hands-On Machine Learning* (2017) de Aurélien Géron (publicaod por O'Reilly). Y la ayuda online, como la de Tensorflow, Keras, MathWorks, Caffe y muchos más, son contribuciones importantes a la ciencia de datos.

El álgebra lineal tiene una maravillosa variedad de matrices : simétricas, ortogonales, triangulares, de banda, permutaciones, proyecciones y circulantes. En mi experiencia, las *matrices simétricas definidas positivas* S son las mejores. Tienen autovalores positivos λ y autovectores ortogonales $\boldsymbol{q}$. Son combinaciones $S = \lambda_1 \boldsymbol{q}_1 \boldsymbol{q}_1^{\mathrm{T}} + \lambda_2 \boldsymbol{q}_2 \boldsymbol{q}_2^{\mathrm{T}} + \cdots$ de proyecciones de rango uno sencillas $\boldsymbol{q}\boldsymbol{q}^{\mathrm{T}}$ sobre sus autovectores. Y si $\lambda_1 \geq \lambda_2 \geq \ldots$ entonces $\lambda_1 \boldsymbol{q}_1 \boldsymbol{q}_1^{\mathrm{T}}$ es la parte más informativa de S. Para una matriz de covarianza muestral, esa parte contiene la mayor varianza.

Capítulo I En el tiempo que hemos vivido, el paso más importante ha sido extender esas ideas que funcionaban en las matrices simétricas a todas las matrices. Ahora necesitamos **dos conjuntos de vectores singulares, los u y los v**. Los valores singulares σ reemplazan a los autovalores λ. La descomposición $A = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^{\mathrm{T}} + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^{\mathrm{T}} + \cdots$ sigue valiendo (esta es la SVD). Con σ decrecientes, esas componentes de rango uno de A todavía están ordenadas en orden de importancia . Ese "Teorema de Eckart-Young" sobre A complementa lo que hace tiempo sabemos sobre la matriz simétrica $A^{\mathrm{T}}A$: para rango k, detengámonos en $\sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^{\mathrm{T}}$.

II Las ideas en el Capítulo I se convierten en algoritmos en el Capítulo II. Para matrices razonablemente grandes, los σ, los $\boldsymbol{u}$ y los $\boldsymbol{v}$ se pueden calcular. Para matrices muy grandes, usamos la aleatorización : tomar muestras de las columnas y de las filas. Para una gran clase de matrices grandes esto funciona bien.

III-IV El Capítulo III se enfoca en matrices de rango bajo y el Capítulo IV, en muchos ejemplos importantes. Buscamos propiedades que hagan que los cálculos sean muy rápidos (en III) o muy útiles (en IV). La *matriz de Fourier* es fundamental en todos aquellos problemas con coeficientes constantes (no cambian con la posición). La transformada discreta es super rápida debido a la **FFT :** la Transformada rápida de Fourier.

V El Capítulo V explica, de la manera más sencilla posible, los temas de estadística que necesitamos. Las ideas principales son siempre **media y varianza :** el *promedio* y la *dispersión* alrededor de ese promedio. Generalmente podemos reducir la media a cero con un simple desplazamiento. El verdadero problema es reducir la varianza (la incertidumbre). Para vectores, matrices y tensores aleatorios, ese problema es más profundo. Sabemos que el *álgebra lineal de la estadística* es esencial para el aprendizaje automático.

VI El Capítulo VI presenta dos clases de problemas de optimización. Primero los problemas lindos de programación lineal y cuadrática y teoría de juegos. La dualidad y los puntos de ensilladura son ideas clave. Pero los objetivos del aprendizaje profundo y de este libro son otros : *problemas muy grandes con una estructura que sea lo más sencilla posible*. "La derivada igual a cero" es todavía la ecuación fundamental. Las derivadas segundas que hubiera usado el método de Newton son demasiado numerosas y demasiado complicadas de calcular. Suele ser incluso imposible usar todos los datos (cuando realizamos un paso descendente para reducir la función de pérdida). Por eso elegimos solo un minilote de datos de entrada en cada paso del descenso por gradiente estocástico.

El éxito del aprendizaje a gran escala proviene del maravilloso hecho de que la aleatoriedad frecuentemente produce confiabilidad cuando hay miles o millones de variables.

VII El Capítulo VII comienza con la arquitectura de una red neuronal. Una capa de entrada está conectada con capas ocultas y finalmente con la capa de salida. Para los datos de entrenamiento, los vectores de entrada $\boldsymbol{v}$ son conocidos. También las respuestas correctas son conocidas (frecuentemente $\boldsymbol{w}$ es la clasificación correcta de $\boldsymbol{v}$). **Optimizamos los pesos x en la función de aprendizaje $\boldsymbol{F}$ para que $\boldsymbol{F}(x, v)$ esté cerca de w para casi todos los datos de entrenamiento v.**

Luego se aplica $\boldsymbol{F}$ a los *datos de prueba*, tomados de la misma población que los datos de entrenamiento. Si $\boldsymbol{F}$ aprendió lo que necesita (*sin sobreajustes* : no queremos ajustar 100 puntos con un polinomio de grado 99), el error en los datos de prueba también será bajo. El sistema reconoce imágenes y habla. Traduce entre distintas lenguas. Puede seguir diseños como ImageNet o AlexNet, ganadores de importantes competencias. Una red neuronal derrotó al campeón mundial de Go.

La función F generalmente es *lineal a trozos*, los pesos van en las multiplicaciones matriciales. Cada neurona en cada capa oculta también tiene una "función de activación" no lineal. La función rampa **ReLU**$(x) =$ (**máximo de 0 y** x) es ahora la favorita.

Existen cada vez más formas de diseñar las capas que conforman $F(x, v)$. Comenzamos con capas *completamente conectadas*: todas las neuronas de la capa n están conectadas con todas las neuronas de la capa $n+1$. Generalmente las *redes neuronales de convolución* (CNN por sus siglas en inglés) son mejores, repiten los mismos pesos en todos los píxeles de una imagen : una construcción muy importante. Otras capas son diferentes. Una capa de *pooling* reduce la dimensión. El *abandono* o *dropout* excluye aleatoriamente neuronas. La *normalización por lotes* restablece la media y la varianza. Todos estos pasos crean una función que ajusta muy bien a los datos de entrenamiento. Así $F(x, v)$ está lista para usarse.

Agradecimientos

Por sobre todo, aprovecho esta oportunidad para agradecer a tantos amigos generosos por su apoyo :

Pawan Kumar, Leonard Berrada, Mike Giles y Nick Trefethen de Oxford
Ding-Xuan Zhou y Yunwen Lei de Hong Kong
Alex Townsend y Heather Wilber de Cornell
Nati Srebro y Srinadh Bhojanapalli de Chicago
Tammy Kolda, Thomas Strohmer, Trevor Hastie y Jay Kuo de California
Bill Hager, Mark Embree y Wotao Yin, por su ayuda con el Capítulo III
Stephen Boyd y Lieven Vandenberghe, por sus excelentes libros
Alex Strang, por crear las mejores figuras, y más
Ben Recht de Berkeley, especialmente.

Sus artículos, correos electrónicos, conferencias y consejos fueron maravillosos.

EL ALFABETO MATRICIAL

A	Cualquier matriz	Q	Matriz ortogonal
C	Matriz circulante	R	Matriz triangular superior
C	Matriz de columnas	R	Matriz de filas
D	Matriz diagonal	S	Matriz simétrica
F	Matriz de Fourier	S	Matriz de covarianza muestral
I	Matriz identidad	T	Tensor
L	Matriz triangular inferior	U	Matriz triangular superior
L	Matriz laplaciana	U	Vectores singulares a izquierda
M	Matriz de mezcla	V	Vectores singulares a derecha
M	Matriz de Markov	X	Matriz de autovectores
P	Matriz de probabilidad	Λ	Matriz de autovalores
P	Matriz de proyección	Σ	Matriz de valores singulares

Videoconferencias : OpenCourseWare ocw.mit.edu y YouTube (**Math 18.06 y 18.065**)
Introduction to Linear Algebra (5th ed) by Gilbert Strang, Wellesley-Cambridge Press
Sitios web del libro : **math.mit.edu/linearalgebra** y **math.mit.edu/learningfromdata**

Índice general

Aprendizaje profundo y redes neuronales iii

Prefacio y agradecimientos vi

Parte I : Aspectos importantes de Álgebra Lineal 1

I.1 Multiplicación $\boldsymbol{Ax}$ usando las columnas de $\boldsymbol{A}$ 2
I.2 Multiplicación matriz-matriz $\boldsymbol{AB}$ 9
I.3 Los cuatro subespacios fundamentales 14
I.4 Eliminación y $\boldsymbol{A = LU}$ 21
I.5 Matrices ortogonales y subespacios 29
I.6 Autovalores y autovectores 36
I.7 Matrices simétricas y definidas positivas 44
I.8 Valores singulares y vectores singulares en la SVD 56
I.9 Componentes principales y la matriz más cercana de rango bajo 71
I.10 Cocientes de Rayleigh y autovalores generalizados 81
I.11 Normas de vectores, de funciones y de matrices 88
I.12 Factorización de matrices y tensores : positivos y ralos 97

Parte II : Cálculos con matrices grandes 113

II.1 Algebra lineal numérica 115
II.2 Cuadrados mínimos : cuatro enfoques 124
II.3 Tres bases para el espacio columna 138
II.4 Álgebra lineal aleatorizada 146

Parte III: Rango bajo y compressed sensing **159**

III.1 Cambios en A^{-1} a partir de cambios en A 160

III.2 Autovalores entrelazados y señales de rango bajo 168

III.3 Valores singulares que decaen rápido 178

III.4 Algoritmos de partición para $\ell^2 + \ell^1$ 184

III.5 Compressed Sensing y completación de matrices 195

Parte IV: Matrices especiales **203**

IV.1 Transformadas de Fourier : discreta y continua 204

IV.2 Matrices de desplazamiento y matrices circulantes 213

IV.3 El producto de Kronecker $A \otimes B$ 221

IV.4 Transformadas de senos y cosenos a partir de sumas de Kronecker 228

IV.5 Matrices de Toeplitz y filtros invariantes de desplazamiento 232

IV.6 Grafos y laplacianos y las leyes de Kirchhoff 239

IV.7 Conglomerados usando métodos espectrales y k-medias 245

IV.8 Completando matrices de rango uno 255

IV.9 El problema ortogonal de Procrustes 257

IV.10 Matrices de distancias 259

Parte V: Probabilidad y Estadística **263**

V.1 Media, varianza y probabilidad 264

V.2 Distribuciones de probabilidad 275

V.3 Momentos, cumulantes y desigualdades de la Estadística 284

V.4 Matrices de covarianza y probabilidades conjuntas 294

V.5 Gaussiana multivariada y mínimos cuadrados ponderados 304

V.6 Cadenas de Markov 311

Parte VI: optimización 321

VI.1 Problemas de mínimos : convexidad y método de Newton 324

VI.2 Multiplicadores de Lagrange = derivadas del costo 333

VI.3 Programación lineal, teoría de juegos y dualidad 338

VI.4 Descenso por gradiente hacia el mínimo 344

VI.5 Descenso por gradiente estocástico y ADAM 359

Parte VII : Aprendizaje con datos 371

VII.1 La construcción de redes neuronales profundas 375

VII.2 Redes neuronales convolucionales 387

VII.3 Retropropagación y la regla de la cadena 397

VII.4 Hiperparámetros : las decisiones cruciales 407

VII.5 El mundo del aprendizaje automático 413

Libros sobre aprendizaje automático 416

Autovalores y valores singulares : rango uno 417

Códigos y algoritmos de álgebra lineal numérica 418

Contar parámetros en factorizaciones básicas 419

Índice de autores 420

Índice 423

Índice de símbolos 432

Parte I

Aspectos importantes del Álgebra Lineal

I.1 **Multiplicación Ax usando las columnas de A**

I.2 **Multiplicación matriz-matriz AB**

I.3 **Los cuatro subespacios fundamentales**

I.4 **Eliminación y $A = LU$**

I.5 **Matrices ortogonales y subespacios**

I.6 **Autovalores y autovectores**

I.7 **Matrices simétricas y definidas positivas**

I.8 **Valores singulares y vectores singulares en la SVD**

I.9 **Componentes principales y la matriz más cercana de rango bajo**

I.10 **Cociente de Rayleigh y autovalores generalizados**

I.11 **Normas de vectores, de funciones y de matrices**

I.12 **Factorización de matrices y tensores: positivos y ralos**

Parte I : Aspectos importantes del Álgebra Lineal

La parte I de este libro es una introducción seria al álgebra lineal aplicada. Si el lector no tiene mucha experiencia o esta no es reciente (en esta importante área de la matemática), por favor, *no la siga a las apresuradas*. Empezamos multiplicando $A\boldsymbol{x}$ y AB usando las columnas de la matriz A. Puede parecer una formalidad, pero en realidad, es fundamental.

Me referiré a cinco problemas básicos que se estudian en este capítulo.

$A\boldsymbol{x} = \boldsymbol{b}$	$A\boldsymbol{x} = \lambda\boldsymbol{x}$	$A\boldsymbol{v} = \sigma\boldsymbol{u}$	**Mininizar** $\|\|A\boldsymbol{x}\|\|^2/\|\|\boldsymbol{x}\|\|^2$	**Factorizar la matriz** $\boldsymbol{A}$

Cada uno de estos problemas parece una pregunta meramente computacional :

Encuentre $\boldsymbol{x}$ Encuentre $\boldsymbol{x}$ y $\boldsymbol{\lambda}$ Encuentre $\boldsymbol{v}$, $\boldsymbol{u}$, y $\boldsymbol{\sigma}$ Factorice $A =$ **columnas** por **filas**

Podrá notar que nuestro objetivo es *entender* (y no tanto resolver). Queremos saber, en primer lugar, si $A\boldsymbol{x} = \boldsymbol{b}$ tiene una solución $\boldsymbol{x}$. "¿ Está el vector $\boldsymbol{b}$ en el espacio columna de A ?" Esa palabrita, "espacio", nos llevará por un largo camino, y muy productivo, como bien podrá observar.

El problema de autovalores $A\boldsymbol{x} = \lambda\boldsymbol{x}$ es muy diferente. No hay un vector $\boldsymbol{b}$, solo miramos la matriz A. Queremos las direcciones de los vectores $\boldsymbol{x}$ tales que $A\boldsymbol{x}$ **tenga la misma dirección que** $\boldsymbol{x}$. Así, a lo largo de esa recta habrán desaparecido todas las interconexiones complicadas de A. El vector $A^2\boldsymbol{x}$ es directamente $\lambda^2\boldsymbol{x}$. La matriz e^{At} (de una ecuación diferencial) aplicada a $\boldsymbol{x}$ es $\boldsymbol{x}$ por $e^{\lambda t}$. Podemos resolver cualquier problema lineal cuando conocemos todos los $\boldsymbol{x}$ y λ.

La ecuación $A\boldsymbol{v} = \sigma\boldsymbol{u}$ es similar y a la vez diferente. Ahora tenemos **dos vectores** $\boldsymbol{v}$ **y** $\boldsymbol{u}$. Nuestra matriz A es probablemente rectangular y llena de datos. ¿Qué parte de esa matriz de datos es importante ? La *descomposición en valores singulares* (SVD, por sus siglas en inglés) encuentra sus términos más simples $\sigma\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}$. Esas piezas son matrices (columna $\boldsymbol{u}$ por fila $\boldsymbol{v}^{\mathrm{T}}$). Toda matriz se construye a partir de estos bloques ortogonales. **La ciencia de datos se encuentra con el álgebra lineal en la SVD.**

Hallar esos términos $\sigma\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}$ es el propósito del Análisis de Componentes Principales (PCA).

La minimización y la factorización expresan problemas aplicados fundamentales. Nos llevan a esos vectores singulares $\boldsymbol{v}$ y $\boldsymbol{u}$. Calcular el mejor $\widehat{\boldsymbol{x}}$ vía cuadrados mínimos y la componente principal $\boldsymbol{v}_1$ en PCA es el problema algebraico de *ajustar los datos*. No presentaremos códigos, que se encuentran en internet, nos centraremos en explicar ideas.

Cuando entienda los espacios columna, los espacios nulos, los autovectores y los vectores singulares, estará listo para aplicarlos de numerosas maneras diferentes : mínimos cuadrados, transformadas de Fourier, LASSO en estadística, y descenso estocástico por gradientes en *deep learning* con redes neuronales.

I.1. Multiplicación Ax usando las columnas de A

Esperamos que usted ya sepa un poco de álgebra lineal. Es un campo maravilloso y más útil para más personas que el cálculo (en nuestra humilde opinión). Pero aun así en los cursos de álgebra lineal a la antigua falta información esencial e importante. La primera sección de este libro trata sobre la multiplicación *matriz-vector* $A\boldsymbol{x}$, el espacio columna de una matriz y el rango.

Siempre usaremos ejemplos para que sea claro.

Ejemplo 1 Multiplique A por x usando las tres filas de A. Luego use las dos columnas :

$$\textbf{Por filas} \quad \begin{bmatrix} 2 & 3 \\ 2 & 4 \\ 3 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2x_1 + 3x_2 \\ 2x_1 + 4x_2 \\ 3x_1 + 7x_2 \end{bmatrix} = \begin{matrix} \text{productos internos} \\ \text{de las filas} \\ \text{con } \boldsymbol{x} = (x_1, x_2) \end{matrix}$$

$$\textbf{Por columnas} \quad \begin{bmatrix} 2 & 3 \\ 2 & 4 \\ 3 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1 \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix} + x_2 \begin{bmatrix} 3 \\ 4 \\ 7 \end{bmatrix} = \begin{matrix} \text{combinación} \\ \text{de las columnas} \\ \boldsymbol{a}_1 \text{ y } \boldsymbol{a}_2 \end{matrix}$$

Podrá observar que ambas formas dan lugar al mismo resultado. La primera (una fila a la vez) produce tres productos internos, también conocidos como *productos punto* debido a la notación :

$$\textbf{fila} \cdot \textbf{columna} = (\mathbf{2},\ \mathbf{3}) \cdot (\boldsymbol{x}_1,\ \boldsymbol{x}_2) = \mathbf{2}\boldsymbol{x}_1 + \mathbf{3}\boldsymbol{x}_2 \tag{1}$$

Esta es la manera de calcular las tres componentes de $A\boldsymbol{x}$ por separado. Se usa para calcular pero no para entender. Es de *bajo nivel*. Entenderlo es de *alto nivel*, y requiere usar vectores.

El enfoque vectorial interpreta $A\boldsymbol{x}$ como una *combinación lineal* de $\boldsymbol{a}_1$ y $\boldsymbol{a}_2$. ¡ Esta es la operación fundamental del álgebra lineal ! Una combinación lineal de $\boldsymbol{a}_1$ y $\boldsymbol{a}_2$ tiene dos pasos :

(1) Multiplicar las columnas $\boldsymbol{a}_1$ y $\boldsymbol{a}_2$ por los *escalares* x_1 y x_2.

(2) Sumar los vectores $x_1\boldsymbol{a}_1 + x_2\boldsymbol{a}_2 = A\boldsymbol{x}$.

Por lo tanto $A\boldsymbol{x}$ es una combinación lineal de las columnas de A. Esto es fundamental.

Este razonamiento nos lleva al **espacio columna** de A. La idea es tomar **todas las combinaciones** de las columnas de A. Están permitidos todos los números reales x_1 y x_2, el espacio contiene $A\boldsymbol{x}$ para todos los vectores $\boldsymbol{x}$. De esta forma obtenemos una cantidad infinita de vectores $A\boldsymbol{x}$. Esos resultados podemos verlos geométricamente.

En nuestro ejemplo, cada $A\boldsymbol{x}$ es un vector en un espacio tridimensional. Ese espacio 3D se llama $\mathbf{R}^3$. (La $\mathbf{R}$ representa los números reales. Los vectores con tres componentes complejas se encuentran en el espacio $\mathbf{C}^3$.) Nos quedaremos con los vectores reales y nos hacemos la siguiente pregunta

¿ Qué subconjunto de $\mathbf{R}^3$ se obtiene con todas las combinaciones $A\boldsymbol{x} = x_1\boldsymbol{a}_1 + x_2\boldsymbol{a}_2$?

Respuesta : todas estas combinaciones producen un **plano**. Este plano contiene la recta completa en la dirección de $\boldsymbol{a}_1 = (2, 2, 3)$ ya que cada vector $x_1\boldsymbol{a}_1$ está incluido en el mismo. El plano también incluye la recta en la dirección de $\boldsymbol{a}_2$ con todos los vectores $x_2\boldsymbol{a}_2$. Además incluye la *suma* de cualquier vector de la primera recta con cualquier vector de la segunda recta. **Esta suma completa un plano infinito que contiene las dos rectas** pero no llena todo el espacio tridimensional $\mathbf{R}^3$.

Definición **Las combinaciones de las columnas** completan el **espacio columna** de A.

En este caso, el espacio columna es un plano. Ese plano incluye al punto cero $(0,0,0)$ que se produce cuando $x_1 = x_2 = 0$. El plano incluye a $(5,6,10) = \boldsymbol{a}_1 + \boldsymbol{a}_2$ y a $(-1,-2,-4) = \boldsymbol{a}_1 - \boldsymbol{a}_2$. Toda combinación $x_1\boldsymbol{a}_1 + x_2\boldsymbol{a}_2$ se encuentra en este espacio columna. ¡ Con probabilidad 1 **no** incluye al punto aleatorio **rand**$(3,1)$! ¿ Qué puntos están en el plano ?

> $\boldsymbol{b} = (\boldsymbol{b_1}, \boldsymbol{b_2}, \boldsymbol{b_3})$ está en el espacio columna de $\boldsymbol{A}$
> **exactamente en los casos en que** $\boldsymbol{Ax} = \boldsymbol{b}$ tiene una solución $(\boldsymbol{x_1}, \boldsymbol{x_2})$

Una vez que entienda esta afirmación, comprenderá al espacio columna $\mathbf{C}(A)$: la solución $\boldsymbol{x}$ muestra cómo expresar el lado derecho $\boldsymbol{b}$ como una combinación $x_1\boldsymbol{a}_1 + x_2\boldsymbol{a}_2$ de las columnas. Para algunos $\boldsymbol{b}$ esto es imposible porque no están en el espacio columna.

Ejemplo 2 $\boldsymbol{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ no está en $\mathbf{C}(A)$. $A\boldsymbol{x} = \begin{bmatrix} 2x_1 + 3x_2 \\ 2x_1 + 4x_2 \\ 3x_1 + 7x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ no tiene solución.

Las primeras dos ecuaciones dan $x_1 = \frac{1}{2}$ y $x_2 = 0$. Luego la ecuación 3 falla : $3\left(\frac{1}{2}\right) + 7(0) = \mathbf{1.5}$ (**no 1**). Esto quiere decir que $\boldsymbol{b} = (1,1,1)$ no está en el espacio columna (el plano de $\boldsymbol{a}_1$ y $\boldsymbol{a}_2$).

Ejemplo 3 ¿ Cuáles son los espacios columna de $A_2 = \begin{bmatrix} 2 & 3 & 5 \\ 2 & 4 & 6 \\ 3 & 7 & 10 \end{bmatrix}$ y $A_3 = \begin{bmatrix} 2 & 3 & 1 \\ 2 & 4 & 1 \\ 3 & 7 & 1 \end{bmatrix}$?

Solución. El espacio columna de A_2 es el mismo plano que antes. La nueva columna $(5,6,10)$ es la suma de la columna 1 más la columna 2. Entonces $\boldsymbol{a}_3 =$ columna 3 ya está en el plano y no agrega nada nuevo. Al incluir esta columna "*dependiente*" no nos extendemos más allá del plano original.

El espacio columna de A_3 es todo el espacio tridimensional $\mathbf{R}^3$. El Ejemplo 2 nos muestra que la nueva tercera columna $(1,1,1)$ no está en el plano $\mathbf{C}(A)$. Nuestro espacio columna $\mathbf{C}(A_3)$ ha crecido. Pero no hay un espacio intermedio entre un plano y todo un espacio 3D. Visualice el plano x-y y un tercer vector (x_3, y_3, z_3) fuera del plano (es decir, un vector con $z_3 \neq 0$). Se combinan para producir **todo vector en** $\mathbf{R}^3$.

He aquí una lista de todos los espacios columna posibles dentro de $\mathbf{R}^3$, de dimensión 0, 1, 2, 3 :

Subespacios de $\mathbf{R}^3$
- El **vector cero** (0, 0, 0) por sí solo
- Una **recta** con todos los vectores $x_1\boldsymbol{a}_1$
- Un **plano** con todos los vectores $x_1\boldsymbol{a}_1 + x_2\boldsymbol{a}_2$
- **El espacio completo $\mathbf{R}^3$** con todos los vectores $x_1\boldsymbol{a}_1 + x_2\boldsymbol{a}_2 + x_3\boldsymbol{a}_3$

En esa lista necesitamos que los vectores $\boldsymbol{a}_1$, $\boldsymbol{a}_2$, $\boldsymbol{a}_3$ sean **independientes**, es decir, la única combinación de ellos que da el vector cero es $0\boldsymbol{a}_1 + 0\boldsymbol{a}_2 + 0\boldsymbol{a}_3$. Entonces $\boldsymbol{a}_1$ por sí solo produce una recta; $\boldsymbol{a}_1$ y $\boldsymbol{a}_2$ producen un plano; $\boldsymbol{a}_1$, $\boldsymbol{a}_2$ y $\boldsymbol{a}_3$ producen todo vector $\boldsymbol{b}$ en $\mathbf{R}^3$. ¡El vector cero esta en todos los subespacios! En el lenguaje del álgebra lineal:

- Tres columnas independientes en $\mathbf{R}^3$ producen una **matriz invertible** : $AA^{-1} = A^{-1}A = I$.
- $A\boldsymbol{x} = 0$ implica $\boldsymbol{x} = (0,0,0)$. Entonces $A\boldsymbol{x} = \boldsymbol{b}$ tiene solo una solución $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$.

Ahora se comprende la idea de las columnas de una matriz invertible de n por n : sus combinaciones llenan el espacio columna: **todo** $\mathbf{R}^n$. Necesitábamos esas ideas y ese lenguaje para poder continuar.

Columnas independientes y el rango de A

Al terminar de escribir lo anterior, pensé que esta corta sección estaba completa, pero estaba *equivocado*. Con muy poco esfuerzo, se puede encontrar una **base** para el espacio columna de A, podemos **factorizar** A en C por R, y podemos demostrar el **primer gran teorema** del álgebra lineal. Usted *verá* el rango de una matriz y las dimensiones de un subespacio.

Para esto hace falta entender la **independencia**. El objetivo es crear una matriz C cuyas columnas vengan directamente de A, pero no incluyan ninguna columna que sea una combinación de columnas previas. Las columnas de C (tantas como sea posible) serán *independientes*. Aquí tenemos una construcción natural de C a partir de las n columnas de A:

Si la columna 1 de A no es completamente cero, colóquela en la matriz C.

Si la columna 2 de A no es múltiplo de la columna 1, colóquela en C.

Si la columna 3 de A no es una combinación de las columnas 1 y 2, colóquela en C. *Prosiga.*

Al final, C tendrá r columnas ($r \leq n$).

Las columnas de C constituirán una *base* para el espacio columna de A.

Las columnas excluidas son combinaciones de esas columnas básicas que conforman C.

Una **base** de un subespacio es un conjunto completo de vectores independientes: **todos los vectores del espacio son combinaciones de los vectores base**. Los siguientes ejemplos le ayudarán a entenderlo.

Ejemplo 4 Si $A = \begin{bmatrix} 1 & 3 & 8 \\ 1 & 2 & 6 \\ 0 & 1 & 2 \end{bmatrix}$ entonces $C = \begin{bmatrix} 1 & 3 \\ 1 & 2 \\ 0 & 1 \end{bmatrix}$ $n = 3$ columnas en A, $\boldsymbol{r = 2}$ columnas en C

La columna 3 de A es $2(\text{columna } 1) + 2(\text{columna } 2)$. La excluimos de la base C.

Ejemplo 5 Si $A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{bmatrix}$ entonces $C = A$. $n = 3$ columnas en A, $\boldsymbol{r = 3}$ columnas en C

Esta matriz A es invertible. El espacio columna es todo $\mathbf{R}^3$. Las tres columnas de A quedan en C.

Ejemplo 6 Si $A = \begin{bmatrix} 1 & 2 & 5 \\ 1 & 2 & 5 \\ 1 & 2 & 5 \end{bmatrix}$ entonces $C = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ $n = 3$ columnas en A, $\boldsymbol{r = 1}$ columna en C

El valor $\boldsymbol{r}$ es el **rango** de A. También es el rango de C, **cuenta las columnas independientes**. Es cierto que podríamos habernos movido de derecha a izquierda en A, comenzando por la *última* columna. Esto no modificaría la cuenta final r. *La base es diferente, pero la cantidad de vectores es siempre la misma*. Ese número r es la **dimensión** del espacio columna de A y de C (que son los mismos espacios).

El rango de una matriz es la dimensión del espacio columna.

La matriz C se conecta con A por medio de una tercera matriz, R: $\boldsymbol{A = CR}$. Sus tamaños son (m por n)=(m por r) (r por n). La *factorización de* A se puede ver en el Ejemplo 4 :

$$\boxed{\boldsymbol{A} = \begin{bmatrix} 1 & 3 & 8 \\ 1 & 2 & 6 \\ 0 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \end{bmatrix} = \boldsymbol{CR}} \tag{2}$$

Cuando C multiplica a la primera columna $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ de R, se obtiene la columna 1 de C y de A.

Cuando C multiplica a la segunda columna $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ de R, se obtiene la columna 2 de C y de A.

Cuando C multiplica a la tercera columna $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$ de R, se obtiene $2(\text{columna1}) + 2(\text{columna2})$.

Esto es la columna 3 de A. Lo único que hay que hacer es colocar los valores correctos en R. Combinaciones de columnas de C producen las columnas de A. Luego $A = CR$ almacenará esta información como una multiplicación matricial. De hecho, R es una matriz famosa en álgebra lineal :

$$\boldsymbol{R} = \textbf{rref}(\boldsymbol{A}) = \textbf{forma reducida por filas de } \boldsymbol{A} \text{ (sin las filas nulas).}$$

El Ejemplo 5 tiene $C = A$ y luego $R = I$ (matriz identidad). El Ejemplo 6 solo tiene *una* columna en C, y luego solo una fila en R:

$$A = \begin{bmatrix} 1 & 2 & 5 \\ 1 & 2 & 5 \\ 1 & 2 & 5 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 5 \end{bmatrix} = CR$$

Las tres matrices tienen $r = 1$
Rango por columnas = Rango por filas

El número de *columnas independientes* es igual al número de *filas independientes*

Este teorema del rango es cierto para toda matriz. ¡En álgebra lineal siempre trabajamos con columnas y filas! Las m filas contienen los mismos números a_{ij} que las n columnas pero los vectores son diferentes.

El teorema se demuestra usando la igualdad $A = CR$. Mirémoslo de otra manera, por filas en lugar de por columnas. La matriz R tiene r filas. **Multiplicar por $\boldsymbol{C}$ toma combinaciones de esas filas.** Como $A = CR$, obtenemos todas las filas de A a partir de las r filas de R. Esas r filas son independientes, entonces son la **base del espacio fila de** A. El espacio columna y el espacio fila de A tienen dimensión r, con r vectores base: las columnas de C y las filas de R, respectivamente.

Un momento : ¿ por qué las filas de R son independientes ? Observe una vez más el Ejemplo 4.

$$A = \begin{bmatrix} 1 & 3 & 8 \\ 1 & 2 & 6 \\ 0 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \end{bmatrix} \begin{matrix} \leftarrow \text{filas de } R \\ \leftarrow \text{independientes} \end{matrix}$$
$$\uparrow \quad \uparrow$$
unos y ceros

Son esos unos y ceros de R los que nos muestran que ninguna fila es combinación de las otras filas.

La gran factorización en la ciencia de datos es la **SVD** de A, que coincide con la CR cuando el primer factor C tiene r columnas *ortogonales* y el secundo factor R tiene r filas *ortogonales*.

Problemas de la Sección I.1

1 Dé un ejemplo donde una combinación lineal de tres vectores no nulos de $\mathbf{R}^4$ es el vector cero. Luego exprese su ejemplo como solución de $A\boldsymbol{x} = \mathbf{0}$. ¿ Qué tamaño tienen A, $\boldsymbol{x}$ y $\mathbf{0}$?

2 Suponga que una combinación de las columnas de A es igual a una combinación diferente de esas columnas. Expréselo como $A\boldsymbol{x} = A\boldsymbol{y}$. Encuentre dos combinaciones de las columnas de A que son iguales al vector cero (en lenguaje matricial: encuentre dos soluciones para $A\boldsymbol{z} = \mathbf{0}$).

3 (Práctica con subíndices) Los vectores $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n$ están en el espacio m-dimensional $\mathbf{R}^m$, y una combinación $c_1\boldsymbol{a}_1 + \cdots + c_n\boldsymbol{a}_n$ es el vector cero. Esta afirmación está escrita en notación de vectores.

(*a*) Escriba la afirmación en notación matricial. Use la matriz A con los vectores $\boldsymbol{a}$ en sus columnas y use el vector columna $\boldsymbol{c} = (c_1, \ldots, c_n)$.

(*b*) Escriba la afirmación en notación escalar. Use subíndices y notación de sumatoria para sumar. El vector columna $\boldsymbol{a}_j$ tiene componentes $a_{1j}, a_{2j}, \ldots, a_{mj}$.

4 Suponga que $A = \textbf{unos}(3, 3)$ es la matriz de 3 por 3 con un 1 en cada posición. Encuentre dos vectores independientes $\boldsymbol{x}$ e $\boldsymbol{y}$ que sean soluciones de $A\boldsymbol{x} = \mathbf{0}$ y $A\boldsymbol{y} = \mathbf{0}$. Escriba la primera ecuación $A\boldsymbol{x} = \mathbf{0}$ (con números) como una combinación de las columnas de A. ¿ Por qué piensa que no le pedí un tercer vector independiente tal que $A\boldsymbol{z} = \mathbf{0}$?

5 Las combinaciones lineales de $\boldsymbol{v} = (1, 1, 0)$ y $\boldsymbol{w} = (0, 1, 1)$ completan un plano en $\mathbf{R}^3$:

(a) Encuentre un vector $\boldsymbol{z}$ que sea perpendicular a $\boldsymbol{v}$ y a $\boldsymbol{w}$. Así $\boldsymbol{z}$ es perpendicular a todos los vectores $c\boldsymbol{v} + d\boldsymbol{w}$ en el plano : $(c\boldsymbol{v} + d\boldsymbol{w})^T\boldsymbol{z} = c\boldsymbol{v}^T\boldsymbol{z} + d\boldsymbol{w}^T z = 0 + 0$.

(b) Encuentre un vector $\boldsymbol{u}$ que no esté en el plano. Verifique que $\boldsymbol{u}^{\mathrm{T}}\boldsymbol{z} \neq 0$.

6 Si $(1, 1)$, $(4, 2)$, y $(1, 3)$ son los tres vértices de un paralelogramo, encuentre las tres posibilidades para el cuarto vértice. Grafique dos.

7 Describa el espacio columna de $A = \begin{bmatrix} \boldsymbol{v} & \boldsymbol{w} & \boldsymbol{v} + 2\boldsymbol{w} \end{bmatrix}$. Describa el espacio nulo de A: todos los vectores $\boldsymbol{x} = (x_1, x_2, x_3)$ solución de $A\boldsymbol{x} = \mathbf{0}$. Sume las *dimensiones* del plano (el espacio columna de A) y de la recta (el espacio nulo de A):

$$\textbf{dimensión del espacio columna} + \textbf{dimensión del espacio nulo} = \textbf{cantidad de columnas}$$

8 $\boldsymbol{A} = \boldsymbol{CR}$ es la representación de las columnas de A en la base formada por las columnas de C con coeficientes en R. Si $A_{ij} = j^2$ es de 3 por 3, escriba A, C y R.

9 Sea todo $\mathbf{R}^3$ el espacio columna de una matriz m por n. ¿ Qué puede decir sobre m ? ¿ Qué puede decir sobre n ? ¿ Qué puede decir sobre el rango r ?

10 Encuentre las matrices C_1 y C_2 que contengan columnas independientes de A_1 y A_2 :

$$A_1 = \begin{bmatrix} 1 & 3 & -2 \\ 3 & 9 & -6 \\ 2 & 6 & -4 \end{bmatrix} \qquad A_2 = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

11 Factorice cada una de esas matrices como $A = CR$. La matriz R contiene los números que multiplicados por las columnas de C recuperan las columnas de A.

Esta es una manera de interpretar la multiplicación de matrices : C **por cada columna de** R**.**

12 Halle una base para el espacio columna de A_1 y de A_2. ¿ Cuáles son las *dimensiones* de los espacios columna (el número de vectores independientes) ? ¿ Cuáles son los *rangos* de A_1 y A_2 ? ¿ Cuántas filas independientes tienen A_1 y A_2 ?

13 Encuentre una matriz A de 4 por 4 de rango 2. ¿ Qué tamaño tienen C y R ?

14 Suponga que dos matrices A y B tienen el mismo espacio columna.

(a) Muestre que los espacios fila pueden ser diferentes.

(b) Muestre que las matrices C (columnas básicas) pueden ser diferentes.

(c) ¿ Qué valor será el mismo para A y B ?

15 Si $A = CR$, la primera fila de A es una combinación de las filas de R. ¿ Qué parte de qué matriz tiene los coeficientes de esa combinación (los números que multiplican las filas de R para producir la fila 1 de A) ?

16 Las filas de R son una base para el espacio fila de A. *¿ Qué significa esta afirmación ?*

17 Para las siguientes matrices con bloques cuadrados, encuentre la factorización $A = CR$. ¿ Cuáles son los rangos de estas matrices ?

$$A_1 = \begin{bmatrix} \text{ceros} & \text{unos} \\ \text{unos} & \text{unos} \end{bmatrix}_{\mathbf{4 \times 4}} \qquad A_2 = \begin{bmatrix} A_1 \\ A_1 \end{bmatrix}_{\mathbf{8 \times 4}} \qquad A_3 = \begin{bmatrix} A_1 & A_1 \\ A_1 & A_1 \end{bmatrix}_{\mathbf{8 \times 8}}$$

18 Si $A = CR$, ¿ cuáles son los factores CR de la matriz $\begin{bmatrix} 0 & A \\ 0 & A \end{bmatrix}$?

19 La *eliminación* resta de la fila i un factor ℓ_{ij} multiplicado por la fila j : es una operación por filas. Muestre cómo esos pasos pueden reducir la matriz A del Ejemplo 4 a R (excepto que una fila de la matriz escalonada R sea cero). ¡El rango no cambiará!

$$A = \begin{bmatrix} 1 & 3 & 8 \\ 1 & 2 & 6 \\ 0 & 1 & 2 \end{bmatrix} \quad \rightarrow \quad \rightarrow \quad R = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix} = \mathbf{rref}\,(A).$$

Un nuevo comienzo y un nuevo libro

Escribir las páginas 2 a 5 me llevó a un nuevo comienzo para enseñar Álgebra Lineal. Pude ver que las ideas claves aparecen muy temprano, siempre que los ejemplos sean pequeñas matrices de enteros.

Permítame utilizar esta página para explicar estas ideas, porque permiten a los estudiantes interiorizarse rápidamente con el Álgebra Lineal. Las ideas me llevaron a un nuevo libro de texto llamado *Linear Algebra for Everyone* (Álgebra Lineal para todos). Diríjase a su sitio web para ver más : **math.mit.edu/everyone**. Las ideas claves son:

columnas independientes **espacio columna** **rango de matriz** **multiplicación $A = CR$**

Ejemplo de $A = CR$

$$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \end{bmatrix}$$

C tiene 2 columnas independientes de A
R tiene 2 filas independientes incluyendo una I
Columna 3 de $A = -1(\text{columna } 1) + 2(\text{columna } 2)$

Las tres matrices tienen rango $r = 2$. C contiene una base del espacio columna de A. R contiene una base del espacio fila de A. Ambos espacios son planos en el espacio tridimensional. $A = CR$ nos lleva a una forma relacionada con tres factores, que trata a las columnas y a las filas de manera simétrica :

$$\boldsymbol{A} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} 1 & 4 \\ 2 & 5 \end{bmatrix}^{\boldsymbol{-1}} \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \end{bmatrix}$$

2 columnas de A en C
2 filas de A en la nueva R
Invertir el solapamiento de 2 por 2

Esta es la primera vez que aparece la inversa, así que la explicamos aquí : $A \times A^{-1} =$ identidad I.

Inversa de una matriz de 2 por 2
No hay inversa si $ad = bc$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Este nuevo enfoque trabaja directamente con la multiplicación matricial CR. Produce combinaciones de las columnas de C. Y cuando miramos a *todas las combinaciones* de dos columnas, completamos un plano. Si la tercera columna de A no hubiera estado sobre ese plano, A habría sido invertible : rango 3. Por favor, visite la página web **math.mit.edu/everyone** del nuevo libro para ver las secciones 1.3 y 1.4.

En la página 11, a continuación de $A = CR$ presentaremos cinco grandes factorizaciones del Álgebra Lineal, comenzando con $A = LU$ para resolver n ecuaciones $A\boldsymbol{x} = \boldsymbol{b}$. Estas factorizaciones son los principios organizadores del Álgebra Lineal. Incluyen autovalores de matrices cuadradas y descomposición en valores singulares de todas las matrices. Estas factorizaciones tienen una visión profunda y exitosa de una matriz, mientras que C toma columnas directamente de A.

Cuando los números que están en A tienen un significado claro, pueden ser todos positivos o podemos tener muchos ceros, ¡ no es una mala idea preservar esas propiedades en C !

Recomendado : verifique que $\begin{bmatrix} 1 & 4 \\ 2 & 5 \end{bmatrix}^{\boldsymbol{-1}} \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \end{bmatrix}$ es igual a $\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \end{bmatrix}$

Las dos factorizaciones coinciden.

Altamente recomendado : cree una matriz A de 3 por 3 con rango 1. Factorice A como $A = CR$.

I.2. Multiplicación matriz-matriz AB

Los **productos internos** (*filas por columnas*) dan como resultado las entradas de $AB = C$:

$$\begin{array}{c}\textbf{fila 2 de } \boldsymbol{A} \times \\ \textbf{columna 3 de } \boldsymbol{B} \\ = \boldsymbol{c_{23}} \textbf{ en } \boldsymbol{C}\end{array} \quad \begin{bmatrix} \cdot & \cdot & \cdot \\ a_{21} & a_{22} & a_{23} \\ \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot & \cdot & b_{13} \\ \cdot & \cdot & b_{23} \\ \cdot & \cdot & b_{33} \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \boldsymbol{c_{23}} \\ \cdot & \cdot & \cdot \end{bmatrix} \tag{1}$$

Ese producto punto c_{23} = (fila 2 de A)•(columna 3 de B) es la suma de componentes a por componentes b:

$$\boldsymbol{c_{23}} = a_{21}\, b_{13} + a_{22}\, b_{23} + a_{23}\, b_{33} = \sum_{k=1}^{3} \boldsymbol{a_{2k}\, b_{k3}} \quad \text{y} \quad \boldsymbol{c_{ij}} = \sum_{k=1}^{n} \boldsymbol{a_{ik}\, b_{kj}}. \tag{2}$$

Así es como se suele calcular cada valor de $AB = C$. Pero hay otra manera.

La otra manera de multiplicar AB es **multiplicar las columnas de A por las filas de B.** ¡ Necesitamos visualizarlo ! Veamos un ejemplo numérico para comprender dos puntos clave : *una columna* $\boldsymbol{u}$ por *una fila* $\boldsymbol{v}^{\mathrm{T}}$ da como resultado una *matriz*. Concentrémonos primero en esa parte de AB. Esta matriz $\boldsymbol{uv}^{\mathrm{T}}$ es particularmente simple :

$$\textbf{Producto externo} \quad \boldsymbol{uv}^{\mathrm{T}} = \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 3 & 4 & 6 \end{bmatrix} = \begin{bmatrix} 6 & 8 & 12 \\ 6 & 8 & 12 \\ 3 & 4 & 6 \end{bmatrix} = \textbf{matriz de rango uno}$$

Una matriz de m por 1 (una columna $\boldsymbol{u}$) multiplicada por una matriz de 1 por p (una fila $\boldsymbol{v}$) da una matriz de m por p. Notemos qué tiene de especial una matriz de rango uno $\boldsymbol{uv}^{\mathrm{T}}$:

$$\text{Todas las columnas de } \boldsymbol{uv}^{\mathrm{T}} \text{ son múltiplos de } \boldsymbol{u} = \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix}$$

$$\text{Todas las filas de } \boldsymbol{uv}^{\mathrm{T}} \text{ son múltiplos de } \boldsymbol{v}^{\mathrm{T}} = \begin{bmatrix} 3 & 4 & 6 \end{bmatrix}$$

El espacio columna de $\boldsymbol{uv}^{\mathrm{T}}$ es unidimensional : *es la recta en la dirección de* $\boldsymbol{u}$. La dimensión del espacio columna (el número de columnas independientes) es el **rango de la matriz**, un número que es clave. Todas las matrices no nulas $\boldsymbol{uv}^{\mathrm{T}}$ tienen rango uno. Son la unidad elemental de cualquier matriz.

Notemos también que **el espacio fila de $\boldsymbol{uv}^{\mathrm{T}}$ es la recta en la dirección de $\boldsymbol{v}$.** Por definición, el espacio fila de cualquier matriz A es el espacio columna $\mathbf{C}(A^{\mathrm{T}})$ de su traspuesta A^{T}. De esta manera, podemos pensar solo en vectores columna. En el ejemplo, trasponemos $\boldsymbol{uv}^{\mathrm{T}}$ **(cambiamos filas por columnas)** para obtener la matriz $\boldsymbol{vu}^{\mathrm{T}}$:

$$(\boldsymbol{uv}^{\mathrm{T}})^{\mathrm{T}} = \begin{bmatrix} \mathbf{6} & 8 & 12 \\ \mathbf{6} & 8 & 12 \\ \mathbf{3} & 4 & 6 \end{bmatrix}^{\mathbf{T}} = \begin{bmatrix} \mathbf{6} & \mathbf{6} & \mathbf{3} \\ 8 & 8 & 4 \\ 12 & 12 & 6 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 6 \end{bmatrix} \begin{bmatrix} 2 & 2 & 1 \end{bmatrix} = \boldsymbol{vu}^{\mathrm{T}}.$$

Estamos viendo el ejemplo más claro posible del primer gran teorema del álgebra lineal :

Rango por filas=rango por columnas $\quad$ **r columnas independientes $\Leftrightarrow$ r filas independientes**

Una matriz no nula $\boldsymbol{uv}^{\mathrm{T}}$ tiene una columna linealmente independiente y una fila linealmente independiente. Todas las columnas son múltiplos de $\boldsymbol{u}$ y todas las filas son múltiplos de $\boldsymbol{v}^{\mathrm{T}}$. Para estas matrices el rango es $r = 1$.

AB = Suma de matrices de rango uno

Miraremos al producto AB como producto de las columnas de A por las filas de B. Sean $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n$ las n columnas de A. Para poder multiplicar AB, la matriz B debe tener n filas, que denotaremos por $\boldsymbol{b}_1^*, \boldsymbol{b}_2^*, \ldots, \boldsymbol{b}_n^*$. **El producto AB es la suma de las columnas $\boldsymbol{a}_k$ multiplicadas por las filas $\boldsymbol{b}_k^*$:**

Multiplicación columna-fila de matrices

$$AB = \begin{bmatrix} | & & | \\ \boldsymbol{a}_1 & \cdots & \boldsymbol{a}_n \\ | & & | \end{bmatrix} \begin{bmatrix} \text{—} & \boldsymbol{b}_1^* & \text{—} \\ & \vdots & \\ \text{—} & \boldsymbol{b}_n^* & \text{—} \end{bmatrix} = \underbrace{\boldsymbol{a}_1\,\boldsymbol{b}_1^* + \boldsymbol{a}_2\,\boldsymbol{b}_2^* + \cdots + \boldsymbol{a}_n\,\boldsymbol{b}_n^*}_{\textbf{suma de matrices de rango uno}}. \tag{3}$$

He aquí un ejemplo de 2 por 2 para mostrar los $n = 2$ términos (columna por fila) y su suma AB :

$$\begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 \\ 0 & 5 \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ \mathbf{3} \end{bmatrix} \begin{bmatrix} \mathbf{2} & \mathbf{4} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{5} \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 6 & 12 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 5 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 6 & 17 \end{bmatrix} \tag{4}$$

Podemos contar la cantidad de multiplicaciones de un número por un número. Cuatro multiplicaciones para obtener 2, 4, 6, 12. Cuatro más para 0, 0, 0, 5. Un total de $2^3 = 8$ multiplicaciones. Siempre hay $\boldsymbol{n^3}$ multiplicaciones cuando A y B son matrices de n por n. Y $\boldsymbol{mnp}$ multiplicaciones cuando $AB = (m \text{ por } n) \text{ por } (n \text{ por } p)$: n matrices de rango uno, cada una de esas matrices es de m por p.

El cálculo es el mismo usando el método tradicional de productos internos. Fila de A multiplicada por columna de B requiere de n multiplicaciones. Hacemos esto para cada número en AB : mp productos internos cuando AB es de m por p. El número total de multiplicaciones es nuevamente $\boldsymbol{mnp}$ cuando multiplico $(m \text{ por } n)$ por $(n \text{ por } p)$.

filas por columnas	$\boldsymbol{mp}$ productos internos,	$\boldsymbol{n}$ multiplicaciones cada uno :	$\boldsymbol{mnp}$
columnas por filas	$\boldsymbol{n}$ productos externos,	$\boldsymbol{mp}$ multiplicaciones cada uno :	$\boldsymbol{mnp}$

Si miramos más detenidamente, son exactamente las mismas multiplicaciones $a_{ik}b_{kj}$ en distintos órdenes. Aquí esta la prueba algebraica de que cada número c_{ij} en $C = AB$ es el mismo usando los productos externos en (3) o usando productos internos en (2) :

El elemento i, j de $\boldsymbol{a}_k\boldsymbol{b}_k^*$ es $a_{ik}b_{kj}$. Sumándolos llegamos a $c_{ij} = \sum_{k=1}^{n} a_{ik}b_{kj} =$ **fila i $\cdot$ columna j**.

Beneficios del punto de vista de columnas por filas

¿Por qué es esencial el método de los productos externos en ciencia de datos ? La respuesta corta es *que buscamos la parte importante de una matriz* A. En general no queremos los números más grandes de A (aunque podrían ser importantes). Lo que queremos son los términos más significativos de A. **Y esos términos son matrices uv^{T} de rango uno**. Un tema predominante en el álgebra lineal aplicada es :

Factorizar A en CR y observar las piezas $c_k r_k^*$ de $A = CR$.

Factorizar A en CR es lo inverso de multiplicar $CR = A$. Factorizar lleva más tiempo, especialmente si los términos involucran *autovalores o valores singulares*. Estos números contienen información interna sobre la matriz A. Pero esa información no es visible hasta que se factoriza.

A continuación describimos cinco factorizaciones importantes con las letras usuales para la matriz producto original (por lo general A) y luego para sus factores. En este libro desarrollaremos las cinco.

$$\boxed{A = LU \qquad A = QR \qquad S = Q\Lambda Q^{\mathrm{T}} \qquad A = X\Lambda X^{-1} \qquad A = U\Sigma V^{\mathrm{T}}}$$

Por ahora mencionaremos palabras clave y propiedades de cada una de estas factorizaciones.

1 $A = LU$ se obtiene por **eliminación**. Combinaciones de filas llevan de A a U y de U de vuelta a A. La matriz L es triangular inferior y U es triangular superior como en la ecuación (4).

2 $A = QR$ se obtiene de **ortogonalizar** las columnas $\boldsymbol{a}_1 \ldots \boldsymbol{a}_n$ como en *Gram-Schmidt*. Q tiene columnas ortogonales ($Q^{\mathrm{T}}Q = I$) y R es triangular superior.

3 $S = Q\Lambda Q^{\mathrm{T}}$ se obtiene de los autovalores $\lambda_1, \cdots, \lambda_n$ de una matriz simétrica $S = S^{\mathrm{T}}$. Los autovalores en la diagonal de Λ. Los **autovectores ortonormales** en las columnas de Q.

4 $A = X\Lambda X^{-1}$ es la **diagonalización** cuando A es de n por n con n autovectores independientes. Los *autovalores* de A en la diagonal de Λ. Los *autovectores* de A en las columnas de X.

5 $A = U\Sigma V^{\mathrm{T}}$ es la **Descomposición en Valores Singulares** de cualquier matriz A (cuadrada o no). Los **valores singulares** $\sigma_1, \ldots, \sigma_r$ en Σ. **Vectores singulares** ortonormales en U y en V.

Voy a elegir una favorita (la número 3) para ilustrar la idea. La factorización especial $Q\Lambda Q^{\mathrm{T}}$ comienza con una matriz simétrica S. Esa matriz tiene autovectores unitarios ortogonales $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n$. Estos autovectores perpendiculares (producto punto $= 0$) van en las columnas de Q.

S y Q son los reyes y reinas del álgebra lineal :

Matriz simétrica S	$S^{\mathrm{T}} = S$	$s_{ij} = s_{ji}$
Matriz ortogonal Q	$Q^{\mathrm{T}} = Q^{-1}$	$q_i \cdot q_j = \begin{cases} 0 & \text{si } i \neq j \\ 1 & \text{si } i = j \end{cases}$

La matriz diagonal Λ contiene los autovalores reales λ_1 a λ_n. Toda matriz simétrica real S tiene n autovectores ortonormales $\boldsymbol{q}_1$ a $\boldsymbol{q}_n$. Al multiplicarlos por S, los autovectores mantienen la dirección. Solamente son reescalados por el número λ :

$$\boxed{\textbf{Autovector } \boldsymbol{q} \textbf{ y autovalor } \boldsymbol{\lambda} \qquad S\boldsymbol{q} = \lambda \boldsymbol{q}} \tag{5}$$

No es fácil encontrar λ y $\boldsymbol{q}$ en una matriz grande. Pero n pares siempre existen cuando S es simétrica. Nuestro propósito es ver cómo $SQ = Q\Lambda$ proviene, columna por columna, de $Sq = \lambda q$:

$$\boldsymbol{SQ} = S\begin{bmatrix} \boldsymbol{q}_1 & \cdots & \boldsymbol{q}_n \end{bmatrix} = \begin{bmatrix} \lambda_1\boldsymbol{q}_1 & \cdots & \lambda_n\boldsymbol{q}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_1 & \cdots & \boldsymbol{q}_n \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} = \boldsymbol{Q\Lambda} \tag{6}$$

Multiplicamos $SQ = Q\Lambda$ por $Q^{-1} = Q^{\mathrm{T}}$ para obtener $S = Q\Lambda Q^{\mathrm{T}} =$ una matriz simétrica. Cada autovalor λ_k y cada autovector $\boldsymbol{q}_k$ contribuye un término $\lambda_k \boldsymbol{q}_k \boldsymbol{q}_k^{\mathrm{T}}$ de rango uno a S.

$$\textbf{Los términos de rango uno} \qquad S = (Q\Lambda)Q^{\mathrm{T}} = (\lambda_1 \boldsymbol{q}_1)\boldsymbol{q}_1^{\mathrm{T}} + (\lambda_2 \boldsymbol{q}_2)\boldsymbol{q}_2^{\mathrm{T}} + \cdots + (\lambda_n \boldsymbol{q}_n)\boldsymbol{q}_n^{\mathrm{T}} \tag{7}$$

$$\textbf{Todas simétricas} \qquad \text{La traspuesta de } \ \boldsymbol{q}_i\boldsymbol{q}_i^{\mathrm{T}} \ \text{ es } \ \boldsymbol{q}_i\boldsymbol{q}_i^{\mathrm{T}} \tag{8}$$

Notar que las columnas de $Q\Lambda$ son $\lambda_1\boldsymbol{q}_1, \ldots, \lambda_n\boldsymbol{q}_n$. Cuando se multiplica a la derecha una matriz por la matriz diagonal Λ, se multiplican sus *columnas* por los λ's.

Terminamos esta sección con un comentario sobre la prueba del teorema espectral $S = Q\Lambda Q^{\mathrm{T}}$: toda matriz simétrica S tiene n autovalores reales y n autovectores ortonormales. En la Sección I.6 hallaremos los autovalores como las raíces del polinomio de orden n, $P_n(\lambda) =$ determinante de $S - \lambda I$. Cuando $S = S^{\mathrm{T}}$ estas raíces son todas reales. El aspecto delicado de la prueba aparece cuando un autovalor λ_i está *repetido*, es decir, cuando es una raíz doble o una raíz de orden M correspondiente a un factor $(\lambda - \lambda_j)^M$. En este caso será necesario hallar M autovectores independientes. El rango de $S - \lambda_j I$ debe ser $n - M$. Esto es cierto cuando $S = S^{\mathrm{T}}$. Pero requiere de una prueba.

De la misma manera, la descomposición en valores singulares $A = U\Sigma V^{\mathrm{T}}$ requiere de mucha paciencia cuando un valor singular σ se repite M veces en la matriz diagonal Σ. Nuevamente, existen M pares de vectores singulares v y u con $Av = \sigma u$. Esta afirmación requiere de una prueba.

Notación por filas Hemos elegido los símbolos $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*$ para denotar las filas de la segunda matriz en AB. Usted podría pensar que deberíamos usar $\boldsymbol{b}_1^{\mathrm{T}}, \ldots, \boldsymbol{b}_n^{\mathrm{T}}$, que era nuestra primera opción, pero esta notación no es clara, pues parecería indicar las columnas de B traspuestas. Como ese factor a la derecha puede ser U, R, Q^{T}, X^{-1} o V^{T}, es mejor dejar en claro que : *queremos las filas de esa matriz.*

G. Strang, *Multiplying and factoring matrices*, Amer. Math. Monthly **125** (2018) 223-230.

G. Strang, *Introduction to Linear Algebra*, 5th ed., Wellesley-Cambridge Press (2016).

Problemas de la Sección I.2

1 Supogamos que $A\boldsymbol{x} = \mathbf{0}$ y $A\boldsymbol{y} = \mathbf{0}$ (donde $\boldsymbol{x}$, $\boldsymbol{y}$ y $\mathbf{0}$ son vectores). Escriba estas dos afirmaciones juntas como una ecuación matricial $AB = C$. ¿ Quiénes son esas matrices B y C? Si la matriz A es de m por n, ¿ cuáles son los tamaños de B y C ?

2 Sean $\boldsymbol{a}$ y $\boldsymbol{b}$ vectores columna con componentes $a_1, \ldots, a_m$ y $b_1, \ldots, b_p$. ¿ Es posible multiplicar $\boldsymbol{a}$ por $\boldsymbol{b}^{\mathrm{T}}$? ¿ Cuál es el tamaño de la respuesta $\boldsymbol{a}\boldsymbol{b}^{\mathrm{T}}$? ¿ Qué número se encuentra en la fila i, columna j de $\boldsymbol{a}\boldsymbol{b}^{\mathrm{T}}$? ¿ Qué puede decir sobre $\boldsymbol{a}\boldsymbol{a}^{\mathrm{T}}$?

3 Continuación del Problema 2 : práctica con subíndices. En lugar de un vector $\boldsymbol{a}$, suponga que hay n vectores $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ en las columnas de A. Suponga que hay n vectores $\boldsymbol{b}_1^{\mathrm{T}}, \ldots, , \boldsymbol{b}_n^{\mathrm{T}}$ en las filas de B.

(a) Halle una fórmula de tipo *suma de matrices de rango uno* para el producto matriz-matriz AB.

(b) Halle la fórmula para el elemento i, j de ese producto matriz-matriz AB. Use notación de sumatoria (Σ) para sumar los elementos i, j de cada matriz $\boldsymbol{a}_k\boldsymbol{b}_k^{\mathrm{T}}$ del Problema 2.

4 Sea B una matriz con una sola columna ($p = 1$), es decir, cada fila tiene un solo número. A tiene columnas $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$. Escriba la fórmula tipo columnas por filas de AB. En palabras, el vector columna AB de m por 1 es una combinación de _____ .

5 Consideremos una matriz B. Si queremos tomar combinaciones de sus filas, hay que premultiplicar por A para obtener AB. Si se quiere tomar combinaciones de sus columnas, hay que postmultiplicar por C para obtener BC. Para este ejercicio tenemos que hacer ambos :

Primero por filas y luego por columnas	primero AB y luego $(AB)C$
Primero por columnas y luego por filas	primero BC y luego $A(BC)$

La **ley asociativa** establece que obtendremos el mismo resultado final con ambos métodos.

Verifique que $\boldsymbol{(AB)C = A(BC)}$ para $A = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}$ $B = \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix}$ $C = \begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix}$.

6 Si A tiene columnas $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3$ y $B = I$ es la matriz identidad, ¿ cuáles son las matrices de rango uno $\boldsymbol{a}_1\boldsymbol{b}_1^*$, $\boldsymbol{a}_2\boldsymbol{b}_2^*$ y $\boldsymbol{a}_3\boldsymbol{b}_3^*$? La suma debería dar $AI = A$.

7 Afirmación : las columnas de AB son combinaciones de las columnas de A. Por lo tanto, el *espacio* columna de AB está *contenido* en el espacio columna de A. Dé ejemplos de matrices A y B donde el espacio columna de AB sea menor que el espacio columna de A.

8 Para calcular $C = AB = (m \text{ por } n)(n \text{ por } p)$, ¿ qué orden de las mismas tres instrucciones "Para" de abajo conduce a los productos externos columnas por filas ?

Filas por columnas	**Columnas por filas**
Para $i = 1$ a m	Para...
Para $j = 1$ a p	Para...
Para $k = 1$ a n	Para...
$C(i,j) = C(i,j) + A(i,k) * B(k,j)$	$C =$

I.3. Los cuatro subespacios fundamentales

En esta sección explicaremos el marco general del álgebra lineal. Veremos cómo cada matriz A de m por n da lugar a cuatro subespacios: dos subespacios de $\mathbf{R}^m$ y dos de $\mathbf{R}^n$. El primer ejemplo es una matriz $\boldsymbol{uv}^{\mathrm{T}}$ de 2 por 2, de rango uno, donde el espacio columna es la recta en la dirección de $\boldsymbol{u}$ y el espacio fila es la recta en la dirección de $\boldsymbol{v}$. El segundo ejemplo es una matriz de 2 por 3.

El tercer ejemplo (una matriz A de 5 por 4) será *la matriz de incidencia de un grafo*. Los grafos se han convertido en los modelos más importantes de la matemática discreta, así que vale la pena entender este ejemplo. Los cuatro subespacios tienen un significado especial en el contexto de grafos.

Ejemplo 1 $\boldsymbol{A} = \begin{bmatrix} \mathbf{1} & \mathbf{2} \\ \mathbf{3} & \mathbf{6} \end{bmatrix} = \boldsymbol{uv}^{\mathrm{T}}$ tiene $m = 2$ y $n = 2$. Tenemos subespacios de $\mathbf{R}^2$.

1 El espacio columna $\mathbf{C}(A)$ es la recta de dirección $\boldsymbol{u} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$. La columna 2 está sobre esa recta.

2 El espacio fila $\mathbf{C}(A^{\mathrm{T}})$ es la recta de dirección $\boldsymbol{v} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$. La fila 2 de A está sobre esa recta.

3 El espacio nulo $\mathbf{N}(A)$ es la recta de dirección $\boldsymbol{x} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$. Entonces $A\boldsymbol{x} = \mathbf{0}$.

4 El espacio nulo a izquierda $\mathbf{N}(A^{\mathrm{T}})$ es la recta de dirección $\boldsymbol{y} = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$. Entonces $A^{\mathrm{T}}\boldsymbol{y} = \mathbf{0}$.

Los cuatro subespacios en la Figura I.1 se construyeron a partir de sus definiciones :

El **espacio columna** $\mathbf{C}(A)$ contiene todas las combinaciones de las columnas de A
El **espacio fila** $\mathbf{C}(A^{\mathrm{T}})$ contiene todas las combinaciones de las columnas de A^{T}
El **espacio nulo** $\mathbf{N}(A)$ contiene todas las soluciones $\boldsymbol{x}$ de $A\boldsymbol{x} = \mathbf{0}$
El **espacio nulo a izquierda** $\mathbf{N}(A^{\mathrm{T}})$ contiene todas las soluciones $\boldsymbol{y}$ de $A^{\mathrm{T}}\boldsymbol{y} = \mathbf{0}$

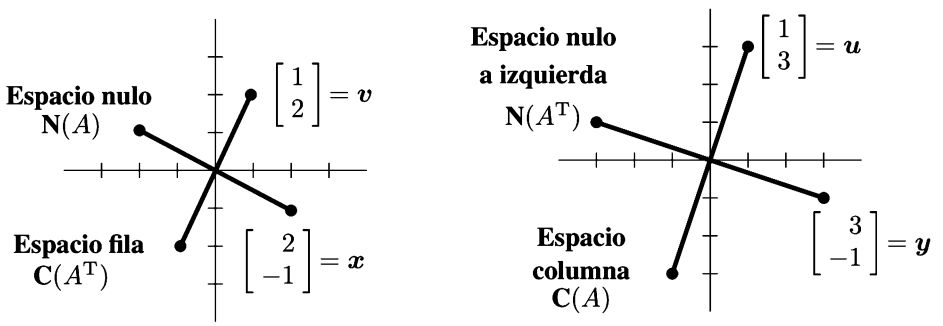


Figura I.1: los cuatro subespacios fundamentales (**4 rectas infinitas**) de $A = \begin{bmatrix} 1 & 2 \\ 3 & 6 \end{bmatrix}$.

El ejemplo tiene un solo $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{x}$ e $\boldsymbol{y}$. Los cuatro subespacios son de dimensión 1 (rectas). Los vectores $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{x}$ e $\boldsymbol{y}$ siempre serán independientes: constituyen una "**base**" de cada subespacio. Una matriz más grande requerirá de más de un vector por base en cada subespacio. La elección de vectores para construir las bases es crucial en cálculo científico.

Ejemplo 2 Sea $B = \begin{bmatrix} 1 & -2 & -2 \\ 3 & -6 & -6 \end{bmatrix}$ con $m = 2$ y $n = 3$. Los subespacios estarán contenidos en $\mathbf{R^3}$ y $\mathbf{R^2}$. Comparando los cambios de A a B, dos de los subespacios se transforman y los otros dos se mantienen iguales. El espacio columna de B sigue estando contenido en $\mathbf{R}^2$, y tiene la misma base. Pero ahora hay $n = 3$ números en las filas de B y la mitad izquierda de la Figura I.2 deberíamos hacerla en $\mathbf{R}^3$. Todavía hay un solo $\boldsymbol{v}$ en el espacio fila. El rango todavía es $r = 1$ porque ambas filas de B tienen la misma dirección (son vectores paralelos).

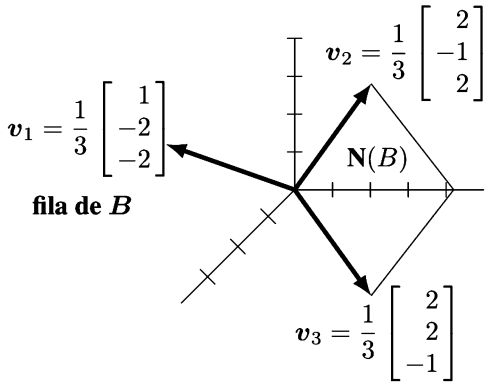
Con $n = 3$ incógnitas y solo $r = 1$ ecuación independiente, $B\boldsymbol{x} = \mathbf{0}$ tendrá $3 - 1 = 2$ soluciones independientes $\boldsymbol{x_1}$ y $\boldsymbol{x_2}$. Todas esas soluciones pertenecen al espacio nulo.

$$B\boldsymbol{x} = \begin{bmatrix} 1 & -2 & -2 \\ 3 & -6 & -6 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ tiene soluciones } \boldsymbol{x}_1 = \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} \text{ y } \boldsymbol{x}_2 = \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}.$$

En el libro *Introduction to Linear Algebra*, a los vectores $\boldsymbol{x}_1$ y $\boldsymbol{x}_2$ se les llama "soluciones especiales". Provienen del proceso de eliminación, y es fácil ver que $B\boldsymbol{x}_1 = \mathbf{0}$ y $B\boldsymbol{x}_2 = \mathbf{0}$. Pero estas no son las elecciones perfectas del espacio nulo de B porque los vectores $\boldsymbol{x}_1$ y $\boldsymbol{x}_2$ **no son perpendiculares**.

Este libro tiene una fuerte preferencia por bases con vectores perpendiculares. La sección II.2 explica cómo producir vectores perpendiculares a partir de vectores independientes usando el *proceso de ortogonalización de "Gram-Schmidt"*.

Nuestro espacio nulo $\mathbf{N}(B)$ es un plano en $\mathbf{R}^3$. En la figura se puede observar una **base ortonormal** $\boldsymbol{v}_2$ y $\boldsymbol{v}_3$. Los ejes $\boldsymbol{v}_2$ y $\boldsymbol{v}_3$ forman un ángulo de 90 ° entre sí y también con $\boldsymbol{v}_1$.



Espacio fila = recta infinita con dirección $\boldsymbol{v}_1$
Espacio nulo = plano infinito por $\boldsymbol{v}_2$ y $\boldsymbol{v}_3$
$n = 3$ columnas de B
$r = 1$ columna independiente

$$\begin{bmatrix} & & \\ \boldsymbol{v}_1 & \boldsymbol{v}_2 & \boldsymbol{v}_3 \\ & & \end{bmatrix} = \begin{matrix} \text{base} \\ \text{ortonormal} \\ \text{de } \mathbf{R}^3 \end{matrix}$$

Figura I.2: espacio fila y espacio nulo de $B = \begin{bmatrix} \mathbf{1} & \mathbf{-2} & \mathbf{-2} \\ \mathbf{3} & \mathbf{-6} & \mathbf{-6} \end{bmatrix}$: ¡ Recta perpendicular al plano !

Ley de conteo: r ecuaciones independientes $Ax = 0$ tienen $n - r$ soluciones independientes

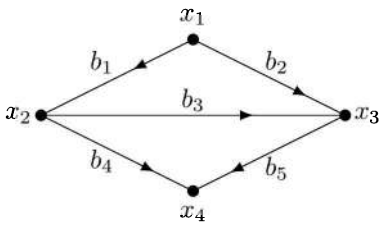
Ejemplo 3 A partir de un grafo El siguiente ejemplo tiene cinco ecuaciones (una para cada arista del grafo). Las ecuaciones tienen cuatro incógnitas (una para cada nodo del grafo). La matriz en $Ax = b$ es **la matriz de incidencia de 5 por 4** del grafo.

A tiene 1 y -1 en cada fila, para señalar el nodo final y el nodo inicial de cada arista.

Diferencias $Ax = b$ a través de las aristas 1, 2, 3, 4, 5 entre los nodos 1, 2, 3, 4

$$\begin{array}{cccccc} -x_1 & +x_2 & & & = b_1 \\ -x_1 & & +x_3 & & = b_2 \\ & -x_2 & +x_3 & & = b_3 \\ & -x_2 & & +x_4 & = b_4 \\ & & -x_3 & +x_4 & = b_5 \end{array}$$

Al comprender los cuatro subespacios fundamentales de esta matriz de incidencia (*los espacios columna y los espacios nulos de A y A^{T}*), usted habrá comprendido una de las ideas centrales del álgebra lineal.



$$\boldsymbol{A} = \begin{array}{c} \text{nodos} \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} \mathbf{-1} & \mathbf{1} & 0 & 0 \\ \mathbf{-1} & 0 & \mathbf{1} & 0 \\ 0 & \mathbf{-1} & \mathbf{1} & 0 \\ 0 & \mathbf{-1} & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{-1} & \mathbf{1} \end{bmatrix} \end{array} \begin{array}{c} \text{aristas} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}$$

Este "grafo" tiene 5 aristas y 4 nodos. A es su matriz de incidencia de 5 por 4.

El espacio nulo $\mathbf{N}(A)$ Para encontrar el espacio nulo tomamos $\boldsymbol{b} = \mathbf{0}$ en las 5 ecuaciones anteriores. Entonces la primera ecuación establece que $x_1 = x_2$. La segunda ecuación es $x_3 = x_1$. La ecuación 4 es $x_2 = x_4$. *Las cuatro incógnitas x_1, x_2, x_3, x_4 tienen el mismo valor c.* **El vector $\boldsymbol{x} = (1, 1, 1, 1)$ y todos los vectores $\boldsymbol{x} = (c, c, c, c)$ son las soluciones de $A\boldsymbol{x} = \mathbf{0}$.**

El espacio nulo es una recta en $\mathbf{R}^4$. La solución especial $\boldsymbol{x} = (1, 1, 1, 1)$ es una base para $\mathbf{N}(A)$. La dimensión de $\mathbf{N}(A)$ es 1 (es decir, un vector en la base, una recta tiene dimensión 1). *El rango de A debe ser* 3, *pues* $n - r = 4 - 3 = 1$. Del rango $r = 3$, se pueden conocer las dimensiones de los cuatro subespacios.

dimensión del espacio fila $= r = 3$ **dimensión del espacio columna $= r = 3$**
dimensión del espacio nulo $= n - r = 1$ **dimensión del espacio nulo de $A^{\mathrm{T}} = m - r = 2$**

El espacio columna $\mathbf{C}(A)$ Existen $r = 4 - 1 = 3$ columnas independientes. La manera más rápida de ver esto es examinando las primeras tres columnas. Ellas forman una base del espacio columna de A:

Las columnas 1, 2, 3 de A son independientes

$$\begin{matrix} -1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{matrix}$$

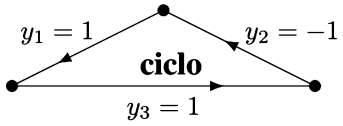
La columna 4 es una combinación de estas tres columnas básicas

"Independientes" significa que la única solución de $A\boldsymbol{x} = \mathbf{0}$ es $(x_1, x_2, x_3) = (0, 0, 0)$. De la ecuación quinta $0x_1 + 0x_2 - x_3 = 0$ sabemos que $x_3 = 0$. Sabemos que $x_2 = 0$ de la cuarta ecuación $0x_1 - x_2 + 0x_3 = 0$. Entonces sabemos que $x_1 = 0$ utilizando la primera ecuación.

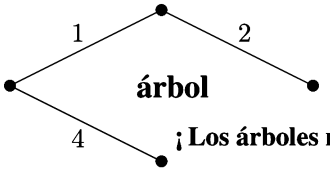
La columna 4 de la matriz de incidencia de A es la suma de esas tres columnas, multiplicada por -1.

El espacio fila $\mathbf{C}(A^{\mathbf{T}})$ La dimensión debe ser $r = 3$, la misma del espacio columna. Pero las primeras 3 filas de A no son independientes: fila 3 = fila 2 − fila 1. *Las tres primeras filas independientes son las filas* 1, 2, 4. Estas filas forman una base (una posible) del espacio fila.

Las aristas $1, 2, 3$ forman un **ciclo** en el grafo : filas dependientes $1, 2, 3$.
Las aristas $1, 2, 4$ forman un **árbol** en el grafo : filas independientes $1, 2, 4$.



Este y resuelve $A^{\mathbf{T}}y = \mathbf{0}$



¡Los árboles no tienen ciclos!

El espacio nulo a izquierda $\mathbf{N}(\boldsymbol{A}^{\mathbf{T}})$ Ahora resolveremos $A^{\mathrm{T}}\boldsymbol{y} = \mathbf{0}$. Las combinaciones de las filas que dan cero. Ya pudimos observar que fila 3 = fila 2 − fila 1, entonces una solución es $\boldsymbol{y} = (1, -1, 1, 0, 0)$. Diremos: esta $\boldsymbol{y}$ se obtiene al seguir el ciclo superior del grafo: hacia adelante (siguiendo la flecha) a lo largo de las aristas 1 y 3 y hacia atrás (en sentido contrario a la flecha) a lo largo de la arista 2.

Otra $\boldsymbol{y}$ se obtiene al recorrer *el ciclo inferior* del grafo: hacia adelante a lo largo de la arista 4, hacia atrás a lo largo las 5 y 3. Esta $\boldsymbol{y} = (0, 0, -1, 1, -1)$ es una solución independiente de $A^{\mathrm{T}}\boldsymbol{y} = \mathbf{0}$. La dimension del espacio nulo a izquierda $\mathbf{N}(A^{\mathrm{T}})$ es $\boldsymbol{m - r} = 5 - 3 = \mathbf{2}$. Entonces esas dos $\boldsymbol{y}$ son una base para el espacio nulo a izquierda.

Tal vez se pregunte cómo se vieron involucrados los "ciclos" y los "árboles" en este problema ya que no era necesario. Se podría haber usado eliminación para resolver $A^{\mathrm{T}}\boldsymbol{y} = \mathbf{0}$. La matriz A^{T} de 4 por 5 tendría tres pivotes. El espacio nulo de A^{T} tiene dimensión dos: $m - r = 5 - 3 = 2$. Pero *los ciclos* y *los árboles* identifican *filas dependientes* y *filas independientes* de una manera muy bella.

Las ecuaciones $A^{\mathrm{T}}\boldsymbol{y} = \mathbf{0}$ dan las "corrientes" y_1, y_2, y_3, y_4, y_5 en las cinco aristas del grafo. Los flujos en los ciclos obedecen **a la ley de corrientes de Kirchhoff**: **lo que ingresa = lo que sale**. Estas palabras se aplican a una red eléctrica. Pero las ideas detrás de esas palabras se encuentran en la ingeniería, la ciencia, la economía y los negocios: equilibrio de fuerzas, flujos y presupuesto.

Los grafos son *el modelo más importante en la matemática discreta aplicada*. Los grafos están por todas partes: las calles, los conductos, el flujo sanguíneo, el cerebro, internet, la economía de un país o del mundo. Podemos entender sus matrices de incidencia A y A^{T}. En la Sección IV.6, la matriz $A^{\mathrm{T}}A$ será la "*matriz laplaciana*". Y la ley de Ohm nos llevará a $A^{\mathrm{T}}CA$.

Cuatro subespacios para un grafo conexo con m aristas y n nodos: matriz de incidencia A

$\mathbf{N}(A)$	Los vectores constantes $(c, c, \ldots, c)$ componen el espacio nulo unidimensional de A.
$\mathbf{C}(A^{\mathrm{T}})$	Las r aristas de un árbol dan r filas independientes de A : rango $= r = n - 1$.
$\mathbf{C}(A)$	***Ley de voltaje***: Las componentes de $A\boldsymbol{x}$ suman zero a lo largo de los ciclos.
$\mathbf{N}(A^{\mathrm{T}})$	***Ley de corriente***: $A^{\mathrm{T}}\boldsymbol{y} =$ (**flujo de entrada**) $-$ (**flujo de salida**) $= \mathbf{0}$ se resuelve con ciclos de corriente.
	Hay $m - r = m - n + 1$ ***ciclos pequeños independientes en el grafo.***

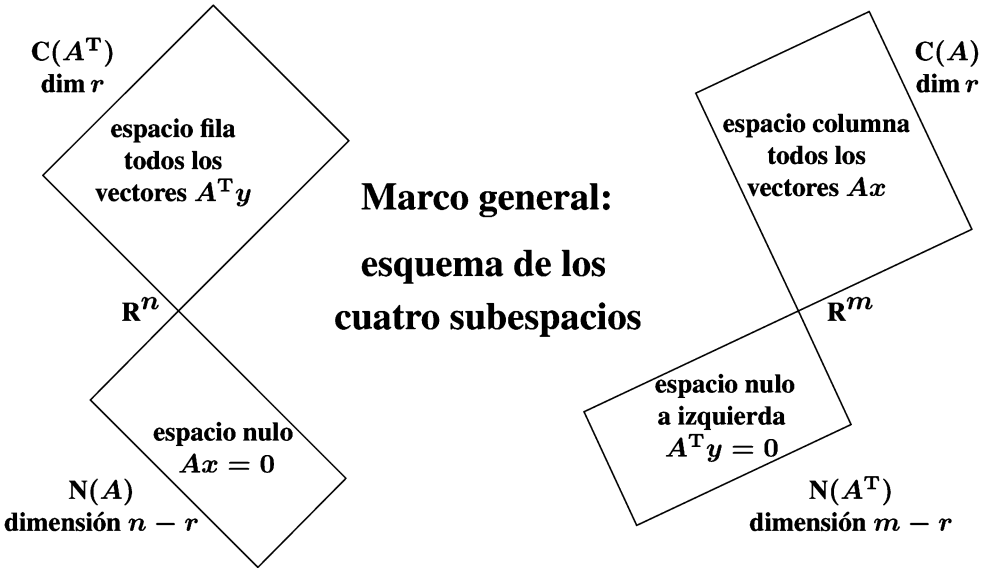


Figura I.3: los cuatro subespacios fundamentales : sus dimensiones suman n y m.

Los rangos de AB y $A+B$

Esta página reúne información esencial sobre los rangos : **cuando multiplicamos matrices, el rango no puede aumentar**. Descubrirá esto inspeccionando los espacios columna y los espacios fila. Hay una situación particular en que el rango no puede disminuir. Entonces conocerá el rango de AB. La afirmación 4 será relevante cuando se factorice una matriz como UV o CR en ciencia de datos.

He aquí cuatro aspectos clave en un mismo lugar: desigualdades e igualdades para el rango.

1 **Rango de $AB \leq$ rango de A Rango de $AB \leq$ rango de B**

2 **Rango de $A+B \leq$ (rango de A) + (rango de B)**

3 **Rango de $A^{\mathrm{T}}A =$ rango de $AA^{\mathrm{T}} =$ rango de $A =$ rango de A^{T}**

4 **Si A es de m por r y B es de r por n**—ambas de rango r— **entonces AB también tiene rango r**

Afirmación 1 se refiere al espacio columna y al espacio fila de AB:

$\mathbf{C}(AB)$ está contenido en $\mathbf{C}(A)$ $\mathbf{C}((AB)^{\mathrm{T}})$ está contenido en $\mathbf{C}(B^{\mathrm{T}})$

Toda columna de AB es una combinación de las columnas de A (*multiplicación matricial*).
Toda fila de AB es una combinación de las filas de B (*multiplicación matricial*).

Como se mencionó en la Sección I.1, **rango por filas = rango por columnas**; es posible usar filas o columnas. *El rango no puede aumentar cuando se multiplica* AB. La afirmación **1** en el recuadro se usa frecuentemente.

Afirmación 2 Cada columna de $A+B$ es la suma de (columna de A) + (columna de B).

rango $(A+B) \leq$ rango (A) + rango (B) se cumple siempre. Se combinan las bases de $\mathbf{C}(A)$, $\mathbf{C}(B)$.

rango$(A+B)$ = rango (A) + rango (B) no siempre es cierto. Es claramente falso si $A = B = I$.

Afirmación 3 Tanto A como $A^{\mathrm{T}}A$ tienen n columnas. **También tienen el mismo espacio nulo.** (Este es el Problema 6). Entonces $n-r$ es el mismo para ambos, y *el rango r es el mismo para ambos*. Entonces rango$(A^{\mathrm{T}}) \leq$ rango$(A^{\mathrm{T}}A) =$ rango(A). Intercambiando A y A^{T} concluimos que sus rangos son iguales.

Afirmación 4 Sean A y B de rango r. Según la afirmación **3**, $A^{\mathrm{T}}A$ y BB^{T} tienen rango r. Como son matrices de r por r resultan invertibles y también lo es su producto $A^{\mathrm{T}}ABB^{\mathrm{T}}$. Entonces

$$r = \textbf{rango de } (\boldsymbol{A^{\mathrm{T}}ABB^{\mathrm{T}}}) \leq \textbf{ rango de } (\boldsymbol{AB})$$

pues según la afirmación **1**: multiplicar por A^{T}, B^{T} no puede aumentar el rango. También rango de $(AB) \leq$ rango de $A = r$, entonces hemos demostrado que **AB tiene exactamente rango r.**

Observación Esto no significa que todos los productos de matrices de rango r tendrán rango r. La afirmación **4** supone que A tiene exactamente r columnas y B tiene r filas, y ambas tienen rango r. La conclusión puede fallar con BA. Por ejemplo:

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix} \qquad \boldsymbol{B} = \begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{-3} \end{bmatrix} \qquad \boldsymbol{AB} \textbf{ tiene rango } \mathbf{1} \qquad \textbf{¡pero } \boldsymbol{BA} \textbf{ es cero!}$$

Problemas de la Sección I.3

1 Demuestre que el espacio nulo de AB contiene al espacio nulo de B. **Si $Bx = 0$ entonces...**

2 Encuentre una matriz cuadrada con rango $(A^2) <$ rango (A). Confirme que rango $(A^{\mathrm{T}}A) =$ rango (A).

3 ¿Cómo se relaciona el espacio nulo de C con los espacios nulos de A y B si $C = \begin{bmatrix} A \\ B \end{bmatrix}$?

4 Si espacio fila de $A =$ espacio columna de A, y $\mathbf{N}(A) = \mathbf{N}(A^{\mathrm{T}})$, ¿es A simétrica?

5 Cuatro posibilidades para el rango r y tamaño m, n coinciden con cuatro posibilidades para $Ax = b$.

Encuentre cuatro matrices A_1 a A_4 que muestren esas posibilidades:

$r = m = n$	$A_1x = b$ tiene **1 solución para cada b**
$r = m < n$	$A_2x = b$ tiene **∞ soluciones**
$r = n < m$	$A_3x = b$ tiene **0 o 1 solución**
$r < m,\ r < n$	$A_4x = b$ tiene **0 o ∞ soluciones**

6 (*Importante*) Demuestre que **$A^{\mathrm{T}}A$ tiene el mismo espacio nulo que A**. Esta es una manera: Primero, si Ax es cero entonces $A^{\mathrm{T}}Ax$ es igual a _____. Esto prueba que $\mathbf{N}(A) \subset \mathbf{N}(A^{\mathrm{T}}A)$.

Segundo, si $A^{\mathrm{T}}Ax = 0$, entonces $x^{\mathrm{T}}A^{\mathrm{T}}Ax = ||Ax||^2 = 0$. Deduzca $\mathbf{N}(A^{\mathrm{T}}A) = \mathbf{N}(A)$.

7 ¿Tienen A^2 y A siempre el mismo espacio nulo? A es una matriz cuadrada.

8 Encuentre el espacio columna $\mathbf{C}(A)$ y el espacio nulo $\mathbf{N}(A)$ de $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$. Recuerde que son espacios vectoriales, no vectores aislados. Este es un ejemplo inusual de $\mathbf{C}(A) = \mathbf{N}(A)$. No podría ocurrir que $\mathbf{C}(A) = \mathbf{N}(A^{\mathrm{T}})$ porque estos dos subespacios son ortogonales.

9 Dibuje un cuadrado y conecte sus esquinas con el punto central: 5 nodos y 8 aristas. Determine la matriz de incidencia A de 8 por 5 de este grafo (rango $r = 5 - 1 = 4$). Encuentre un vector x perteneciente a $\mathbf{N}(A)$ y $8 - 4$ vectores independientes y pertenecientes a $\mathbf{N}(A^{\mathrm{T}})$.

10 Si $\mathbf{N}(A)$ es el subespacio que contiene solamente al vector cero, ¿qué vectores ipertenecen al espacio nulo de $B = [A \ \ A \ \ A]$?

11 Si $\mathbf{S}$ y $\mathbf{T}$ son subespacios de $\mathbf{R}^{10}$ con dimensiones 2 y 7, ¿cuáles son las posibles dimensiones de los siguientes subespacios?

(i) $\mathbf{S} \cap \mathbf{T} =$ {todos los vectores que están en ambos subespacios}

(ii) $\mathbf{S} + \mathbf{T} =$ {todas las sumas de $s + t$ con s en $\mathbf{S}$ y t en $\mathbf{T}$}

(iii) $\mathbf{S}^{\perp} =$ {todos los vectores de $\mathbf{R}^{10}$ que son perpendiculares a todos los vectores de $\mathbf{S}$}.

I.4. Eliminación y $A = LU$

El primer problema y el más fundamental del álgebra lineal es resolver $A\boldsymbol{x} = \boldsymbol{b}$. Nos dan la matriz A de n por n y el vector columna $\boldsymbol{b}$ de n por 1. Buscamos el vector solución $\boldsymbol{x}$. Sus componentes $x_1, x_2, \ldots, x_n$ son las n incógnitas y tenemos n ecuaciones. Generalmente para una matriz cuadrada A hay solamente una solución de $A\boldsymbol{x} = \boldsymbol{b}$ (pero no siempre es así). Podemos encontrar $\boldsymbol{x}$ usando geometría o álgebra.

Esta sección comienza con el gráfico por filas y el gráfico por columnas de $A\boldsymbol{x} = \boldsymbol{b}$. Luego se resolverán las ecuaciones simplificándolas : eliminando x_1 de $n-1$ ecuaciones para obtener un sistema $A_2\boldsymbol{x}_2 = \boldsymbol{b}_2$ de tamaño $n-1$. Eventualmente llegaremos al sistema de 1 por 1 $A_n x_n = b_n$ y sabemos que $x_n = b_n/A_n$. Trabajando hacia atrás obtenemos x_{n-1} y eventualmente conoceremos x_2 y x_1.

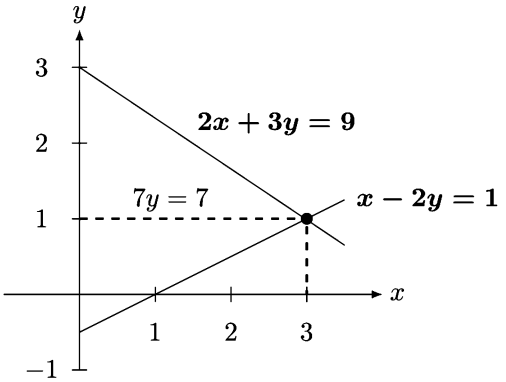
El objetivo de esta sección es poder ver los pasos del proceso de eliminación en términos de matrices de rango 1. **En cada paso (a partir de $\boldsymbol{A}$ pasando por $\boldsymbol{A_2}$ hasta $\boldsymbol{A_n}$) se elimina una matriz $\boldsymbol{\ell u^*}$.** Así la matriz A original es la suma de esas matrices de rango uno. Como veremos la suma es exactamente la grandiosa factorización $A = LU$ en matrices triangulares inferior y superior L y U.

$\boldsymbol{A = L}$ por $\boldsymbol{U}$ es la descripción matricial del proceso de eliminación sin intercambios de filas. Eso será el punto de vista algebraico. Comencemos con geometría en este ejemplo de 2 por 2.

2 ecuaciones y 2 incógnitas
matriz de 2 por 2 en $\boldsymbol{Ax = b}$

$$\begin{bmatrix} 1 & -2 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 9 \end{bmatrix} \qquad \begin{matrix} x - 2y = 1 \\ 2x + 3y = 9 \end{matrix} \tag{1}$$

Importante : $A\boldsymbol{x}$ se multiplicó usando los productos internos (productos punto). Cada fila de la matriz A se multiplicó por el vector $\boldsymbol{x}$. Lo que resultó en las dos ecuaciones para x e y, y las dos rectas de la Figura I.4, que se cortan en la solución $x = 3$, $y = 1$. He aquí **el gráfico por filas**.



$$\begin{bmatrix} 1 & -2 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 9 \end{bmatrix}$$

$$A \quad \boldsymbol{x} \quad = \quad \boldsymbol{b}$$

En la solución

$$\begin{bmatrix} 1 & -2 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} \mathbf{3} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} 1 \\ 9 \end{bmatrix}$$

Figura I.4: **el gráfico por filas** $A\boldsymbol{x} = \boldsymbol{b}$: dos rectas se cortan en la solución $x = 3, y = 1$.

En la Figura I.4 también se muestra la recta horizontal $7y = 7$. Esta ecuación se obtiene al restar $2\times$(ecuación 1) de (ecuación 2). La incógnita x ha sido eliminada de $7y = 7$. Esto es álgebra :

$$\begin{bmatrix} \mathbf{1} & \mathbf{-2} \\ \mathbf{2} & \mathbf{3} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ \mathbf{9} \end{bmatrix} \quad \text{se convierte en} \quad \begin{bmatrix} \mathbf{1} & \mathbf{-2} \\ \mathbf{0} & \mathbf{7} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ \mathbf{7} \end{bmatrix} \qquad \begin{matrix} \boldsymbol{x = 3} \\ \boldsymbol{y = 1} \end{matrix}$$

Gráfico por columnas **Una ecuación vectorial** en lugar de dos ecuaciones escalares. Buscamos hallar una combinación de las columnas de A que dé como resultado $\boldsymbol{b}$. La figura I.5 muestra que la combinación correcta (la solución $\boldsymbol{x}$) tiene los mismos $x = 3$ e $y = 1$ del gráfico por filas.

$A\boldsymbol{x}$ es una combinación de columnas
Las columnas se combinan para obtener $\boldsymbol{b}$

$$\begin{bmatrix} 1 & -2 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = x \begin{bmatrix} 1 \\ 2 \end{bmatrix} + y \begin{bmatrix} -2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ 9 \end{bmatrix} \quad (2)$$

Sumarle $\mathbf{3}\times$**(columna 1)** a $\mathbf{1}\times$**(columna 2)** nos da $\boldsymbol{b}$ como una combinación de las columnas.

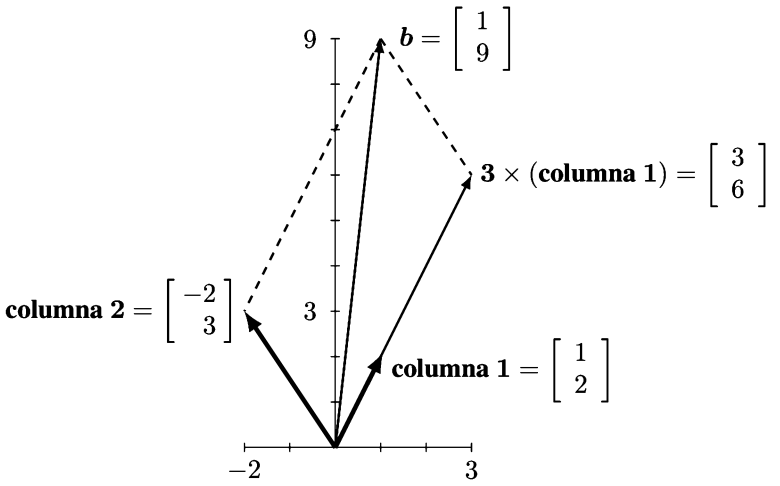


Figura I.5: **gráfico por columnas** : $3\times$(columna 1) $+$ $1\times$(columna 2) nos da $\boldsymbol{b}$.

Para $n = 2$, el gráfico por filas parece sencillo. Pero para $n \geq 3$, gana el gráfico por columnas. ¡Es mejor dibujar tres vectores columna que tres planos! Tres ecuaciones para $\boldsymbol{x} = (x, y, z)$.

Gráfico por filas en 3D	Los tres planos se cortan en un punto. Un plano para cada ecuación.
Gráfico por columnas 3D	Tres vectores columna se combinan para obtener el vector $\boldsymbol{b}$.

Resolviendo $A\boldsymbol{x} = \boldsymbol{b}$ por eliminación

No es fácil visualizar tres planos que se corten en $\mathbf{R}^3$. Y n "hiperplanos" cortándose en un punto en $\mathbf{R}^n$ es sin dudas desafiante. Es más simple combinar los vectores columna : la matriz A debe tener 3 (o n) columnas **independientes**. *No todas las columnas deben estar en el mismo plano de* $\mathbf{R}^3$ (o en el mismo hiperplano de $\mathbf{R}^n$). Esto se traduce algebraicamente de la siguiente manera :

Columnas independientes **La única solución de $A\boldsymbol{x} = \mathbf{0}$ es el vector nulo $\boldsymbol{x} = \mathbf{0}$.**

En palabras, **la independencia** significa que la única combinación que da como resultado el vector nulo es la que tiene cero multiplicado por cada columna. Entonces la única solución para $A\boldsymbol{x} = \mathbf{0}$ es $\boldsymbol{x} = \mathbf{0}$. Cuando esto se cumple, el proceso de eliminación resolverá $A\boldsymbol{x} = \boldsymbol{b}$ y encontrará la única combinación de columnas que produce $\boldsymbol{b}$.

He aquí el proceso completo, columna por columna, cuando la eliminación se puede llevar a cabo en su orden natural:

Columna 1. *Use la ecuación* **1** *para crear ceros debajo del primer pivote. ¡ Los pivotes no pueden ser cero !*

Columna 2. *Use la nueva ecuación* **2** *para crear ceros debajo del segundo pivote.*

Columnas 3 a n. *Continúe hasta encontrar la U triangular superior: n pivotes en su diagonal.*

$$\textbf{Paso 1}\begin{bmatrix} \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x} \\ \boldsymbol{0} & x & x & x \\ \boldsymbol{0} & x & x & x \\ \boldsymbol{0} & x & x & x \end{bmatrix}. \qquad \textbf{Paso 2}\begin{bmatrix} \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x} \\ \boldsymbol{0} & \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x} \\ \boldsymbol{0} & \boldsymbol{0} & x & x \\ \boldsymbol{0} & \boldsymbol{0} & x & x \end{bmatrix}. \qquad \textbf{Final } U = \begin{bmatrix} \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x} \\ & \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x} \\ & & \boldsymbol{x} & \boldsymbol{x} \\ & & & \boldsymbol{x} \end{bmatrix}.$$

La fila 1 es la primera fila pivote—no cambia. Multiplicamos esa fila por los números $\ell_{21}, \ell_{31}, \ell_{41}$ y la restamos de las filas $2, 3, 4$ de A. Los números para obtener los ceros en la primera columna fueron

$$\textbf{Multiplicadores} \qquad \boldsymbol{\ell_{21} = \frac{a_{21}}{a_{11}}} \qquad \boldsymbol{\ell_{31} = \frac{a_{31}}{a_{11}}} \qquad \boldsymbol{\ell_{41} = \frac{a_{41}}{a_{11}}}$$

Si la primera entrada es $a_{11} = 3 =$ primer pivote, y si el a_{21} debajo es 12, entonces $\ell_{21} = 12/3 = 4$.

El paso 2 usa la nueva fila 2 (la segunda fila pivote). Multiplicamos esa fila por ℓ_{32} y ℓ_{42} y la restamos a las filas 3 y 4 para obtener ceros en la segunda columna. Continuamos hasta llegar a U.

Hasta ahora hemos trabajado con la matriz A (no con $\boldsymbol{b}$). El proceso de eliminación en A requiere de $\frac{1}{3}n^3$ diferentes multiplicaciones y sumas, mucho más que los n^2 pasos para cada $\boldsymbol{b}$ del lado derecho. Es necesario un registro de ese trabajo y la forma perfecta es mediante un producto $\boldsymbol{A = LU}$ de matrices triangulares: **triangular inferior $\boldsymbol{L}$ por triangular superior $\boldsymbol{U}$.**

La factorización $A = LU$

¿ Cómo está relacionada la A original con la matriz final U ? Los multiplicadores ℓ_{ij} nos llevaron de una a la otra en tres pasos. El primer paso redujo el problema de 4 por 4 a un problema de 3 por 3, al restar múltiplos de la fila 1:

$$\textbf{Idea clave: Paso 1 quitamos } \boldsymbol{\ell_1 u_1^*} \qquad A = \begin{bmatrix} 1 \text{ por fila } 1 \\ \ell_{21} \text{ por fila } 1 \\ \ell_{31} \text{ por fila } 1 \\ \ell_{41} \text{ por fila } 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & & & \\ 0 & & \boldsymbol{A_2} & \\ 0 & & & \end{bmatrix}. \tag{3}$$

¿ Qué hemos hecho ? Se quitó de A la primera matriz de la derecha, que es un vector columna $1, \ell_{21}, \ell_{31}, \ell_{41}$ por la fila 1. **¡ Es la matriz $\boldsymbol{\ell_1 u_1^*}$ de rango 1!**

Ejemplo de 3 por 3
Quitar una matriz de rango 1
Anular columna / fila

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 7 \\ 2 & 7 & 8 \end{bmatrix} - \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 2 & 4 & 6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \boldsymbol{1} & \boldsymbol{1} \\ 0 & \boldsymbol{3} & \boldsymbol{2} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & & \\ 0 & & \boldsymbol{A_2} \end{bmatrix}$$

El siguiente paso trabaja con la columna 2 de la matriz remanente A_2. La nueva fila 2 es $\boldsymbol{u_2^*}$ = la segunda fila pivote. Hay que multiplicarla por $\ell_{12} = 0$, $\ell_{22} = 1$, ℓ_{32} y ℓ_{42}. **Luego hay que restar $\boldsymbol{\ell_2 u_2^*}$ de las cuatro filas**. Ahora la fila 2 también es cero y A_2 se reduce a A_3.

$$\textbf{Paso 2} \qquad A = \boldsymbol{\ell_1 u_1^*} + \begin{bmatrix} \mathbf{0} \text{ por la fila pivote 2} \\ \mathbf{1} \text{ por la fila pivote 2} \\ \boldsymbol{\ell_{32}} \text{ por la fila pivote 2} \\ \boldsymbol{\ell_{42}} \text{ por la fila pivote 2} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \multicolumn{2}{c}{\multirow{2}{*}{$\boxed{\boldsymbol{A_3}}$}} \\ 0 & 0 & & \end{bmatrix}. \tag{4}$$

Este último paso consistió en restar la matriz de rango uno $\boldsymbol{\ell_2 u_2^*}$ con $\boldsymbol{\ell_2} = (0, 1, \ell_{32}, \ell_{42})$ y $\boldsymbol{u_2^*}$ = fila pivote 2. El paso 3 reducirá la matriz de 2 por 2 A_3 a un solo número A_4 (matriz de 1 por 1). En este momento la fila pivote $\boldsymbol{u_3^*}$ = fila 1 de A_3 tiene solo dos elementos no nulos. Y la columna $\boldsymbol{\ell_3}$ es $(0, 0, 1, \ell_{43})$.

Esta manera de ver el proceso de la eliminación, de a una columna a la vez, produce $\boldsymbol{A = L\,U}$. La multiplicación matricial LU es siempre suma de las columnas de L por filas de U :

$$\boldsymbol{A = \ell_1 u_1^* + \ell_2 u_2^* + \ell_3 u_3^* + \ell_4 u_4^*} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \ell_{21} & 1 & 0 & 0 \\ \ell_{31} & \ell_{32} & 1 & 0 \\ \ell_{41} & \ell_{42} & \ell_{43} & 1 \end{bmatrix} \begin{bmatrix} \text{fila pivote 1} \\ \text{fila pivote 2} \\ \text{fila pivote 3} \\ \text{fila pivote 4} \end{bmatrix} = \boldsymbol{L\,U}. \tag{5}$$

El proceso de eliminación factorizó $\boldsymbol{A = LU}$ en una $\boldsymbol{L}$ triangular inferior multiplicada por una $\boldsymbol{U}$ triangular superior

Observaciones sobre la factorización $\boldsymbol{LU}$ Hemos presentado $A = LU$ a partir de la idea principal de eliminación : reducir el tamaño del problema de n a $n-1$ eliminando x_1 de las últimas $n-1$ ecuaciones. Se restaron múltiplos de la fila 1 (la fila pivote). Entonces la matriz restada era de *rango uno*. Luego de n pasos, la matriz A completa es suma de n matrices de rango uno. Esa suma (por la regla de las columnas por las filas para la multiplicación matricial) es exactamente L por U.

Esta prueba no se encuentra en mi libro *Introduction to Linear Algebra*. La idea era mirar las filas de U en lugar de trabajar con las columnas de A. La fila 3 se obtuvo a partir de restar múltiplos de las filas pivote 1 y 2 de la fila 3 de A :

$$\text{Fila 3 de } U = (\text{fila 3 de } A) - \ell_{31}\,(\text{fila 1 de } U) - \ell_{32}\,(\text{fila 2 de } U). \tag{6}$$

Reescribimos esta ecuación para ver que la fila $[\ell_{31} \;\; \ell_{32} \;\; 1]$ de L multiplica la matriz U :

$$\text{Fila 3 de } A = \ell_{31}\,(\text{fila 1 de } U) + \ell_{32}\,(\text{fila 2 de } U) + 1\,(\text{fila 3 de } U). \tag{7}$$

Esta es la fila 3 de $\boldsymbol{A = LU}$. La clave está en que las filas restadas eran filas pivote y ya se encontraban en U. **Sin intercambios de fila, se puede ver que $\boldsymbol{A = LU}$.**

La solución de $Ax = b$

Al resolver un sistema hay que aplicar las mismas operaciones al lado derecho de una ecuación y al lado izquierdo. La manera directa es incluir $\boldsymbol{b}$ como una columna adicional. Se trabaja con la matriz $[A \;\; \boldsymbol{b}]$. Los pasos para eliminación en A (multiplicaron A por L^{-1} para obtener U) también se aplican a $\boldsymbol{b}$:

A partir de $\left[\begin{array}{cc} A & \boldsymbol{b} \end{array}\right] = \left[\begin{array}{cc} LU & \boldsymbol{b} \end{array}\right]$ **El proceso de eliminación produce** $\left[\begin{array}{cc} U & L^{-1}\boldsymbol{b} \end{array}\right] = \left[\begin{array}{cc} U & \boldsymbol{c} \end{array}\right]$.

Los pasos de A a U (triangular superior) modificarán el lado derecho $\boldsymbol{b}$ a $\boldsymbol{c}$. **El proceso de eliminación en $\boldsymbol{Ax = b}$ produce las ecuaciones $\boldsymbol{Ux = c}$ que permiten fácilmente realizar sustitución hacia atrás**:

$$\begin{array}{l} 2x + 3y = \;\; 8 \\ 4x + 7y = 18 \end{array} \rightarrow \left[\begin{array}{ccc} 2 & 3 & 8 \\ 4 & 7 & 18 \end{array}\right] \rightarrow \left[\begin{array}{ccc} \mathbf{2} & \mathbf{3} & \mathbf{8} \\ 0 & \mathbf{1} & \mathbf{2} \end{array}\right] = \left[\begin{array}{cc} U & \boldsymbol{c} \end{array}\right]. \tag{8}$$

L restó 2 veces la fila 1 de la fila 2. Luego el sistema triangular $U\boldsymbol{x} = \boldsymbol{c}$ se resuelve con **sustitución hacia atrás**—de abajo hacia arriba:

$$\begin{array}{r} 2x + 3y = 8 \\ 1y = 2 \end{array} \quad \text{nos da} \quad \boldsymbol{y = 2} \quad \text{y luego} \quad \boldsymbol{x = 1}. \quad U\boldsymbol{x} = \boldsymbol{c} \quad \text{nos da} \quad \boldsymbol{x} = U^{-1}\boldsymbol{c}.$$

Mirando más detenidamente, el sistema cuadrado $A\boldsymbol{x} = \boldsymbol{b}$ se convirtió en dos sistemas triangulares:

$\boldsymbol{Ax = b}$ **se separó en** $\boldsymbol{Lc = b}$ **y** $\boldsymbol{Ux = c}$**. La eliminación dio** $\boldsymbol{c}$ **y la sustitución hacia atrás dio** $\boldsymbol{x}$.

El resultado final es $\boldsymbol{x} = U^{-1}\boldsymbol{c} = U^{-1}L^{-1}\boldsymbol{b} = A^{-1}\boldsymbol{b}$. Se ha encontrado la solución correcta.

Por favor observe Esos pasos requirieron pivotes no nulos. Se dividió por esos números. El primer pivote era a_{11}. El segundo pivote estaba en la esquina de A_2, el n-ésimo pivote estaba en la matriz A_n de 1 por 1. Estos números se encuentran sobre la diagonal principal de U.

¿Qué hacemos si $\boldsymbol{a_{11} = 0}$? El primer pivote no puede ser cero. Si hay un número distinto de cero más abajo en la columna 1, la fila a la cual pertenece puede ser la fila pivote. **Un buen código elegirá al mayor número como pivote.** De esta manera se reducen los errores, incluso cuando a_{11} es distinto de cero.

A continuación observaremos el efecto de los intercambios de filas en $A = LU$. Se utilizará una matriz P.

Intercambios de filas (Permutaciones)

En este ejemplo, el número mayor de la columna 1 se encuentra en la fila 3: $a_{31} = 2$. **La fila 3 será la primera fila pivote** $\boldsymbol{u_1^*}$. Esa fila se multiplica por $\ell_{21} = \frac{1}{2}$ y se resta de la fila 2.

$$\begin{array}{l} \boldsymbol{u_1^*} = \textbf{ fila 3 de } \boldsymbol{A} \\ \textbf{= primera fila pivote} \end{array} \qquad A = \left[\begin{array}{ccc} 0 & 1 & 1 \\ 1 & 3 & 7 \\ \mathbf{2} & \mathbf{4} & \mathbf{8} \end{array}\right] \rightarrow \left[\begin{array}{ccc} 0 & 1 & 1 \\ 0 & 1 & 3 \\ \mathbf{2} & \mathbf{4} & \mathbf{8} \end{array}\right] \tag{9}$$

Nuevamente, ese paso del proceso de eliminación resta una matriz de rango uno $\boldsymbol{\ell_1 u_1^*}$. Pero A_2 está en una nueva ubicación.

$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 3 & 7 \\ 2 & 4 & 8 \end{bmatrix} = \begin{bmatrix} 0 \\ 1/2 \\ 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & 8 \end{bmatrix} + \begin{bmatrix} 0 & \boxed{1} & \boxed{1} \\ 0 & \boxed{1} & \boxed{3} \\ 0 & 0 & 0 \end{bmatrix} \quad \boldsymbol{\leftarrow A_2} \tag{10}$$

La eliminación en A_2 produce dos términos más de rango uno. Entonces $A = LU$ tiene tres sumandos :

$$\boldsymbol{\ell_1 u_1^*} + \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1/2 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & 4 & 8 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix}. \tag{11}$$

¡ La última matriz U es triangular pero la matriz L no lo es ! El orden de los pivotes para esta A es **3**,**1**,**2**. Si queremos que los pivotes sean $\mathbf{1}, \mathbf{2}, \mathbf{3}$ es necesario mover la fila 3 de A hacia arriba :

Intercambio de filas por medio de una permutación P

$$\boldsymbol{PA} = \begin{bmatrix} 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & 0 \\ 0 & \mathbf{1} & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 3 & 7 \\ \mathbf{2} & \mathbf{4} & \mathbf{8} \end{bmatrix} = \begin{bmatrix} \mathbf{2} & \mathbf{4} & \mathbf{8} \\ 0 & 1 & 1 \\ 1 & 3 & 7 \end{bmatrix}$$

Cuando ambos lados de $A\boldsymbol{x} = \boldsymbol{b}$ se multiplican por P, **se restaura el orden y $\boldsymbol{PA = LU}$** :

$$\boldsymbol{PA} = \begin{bmatrix} 2 & 4 & 8 \\ 0 & 1 & 1 \\ 1 & 3 & 7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1/2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & 8 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix} = \boldsymbol{L\,U}. \tag{12}$$

Para cada matriz A de n por n invertible se tiene que $PA = LU$: P = permutación.

Existen seis permutaciones de 3 por 3 : seis maneras de ordenar las filas de la matriz identidad.

1 intercambio (**P impar**)

$$\boldsymbol{P_{213}} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{P_{321}} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \boldsymbol{P_{132}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

0 o 2 intercambios (**P par**)

$$\boldsymbol{P_{123}} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \quad \boldsymbol{P_{312}} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \boldsymbol{P_{231}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

La inversa de cualquier matriz de permutación P es su traspuesta $P^{\mathbf{T}}$. Los intercambios de fila también se aplican al lado derecho $\boldsymbol{b}$ si estamos resolviendo $A\boldsymbol{x} = \boldsymbol{b}$. La computadora recuerda los intercambios sin realmente mover las filas.

Hay $\boldsymbol{n!}$ (n factorial) matrices de permutación de tamaño n : $3! = (3)(2)(1) = 6$. Cuando A tiene filas dependientes (no tiene inversa), la eliminación lleva a una fila nula y termina antes de tiempo.

Problemas de la Sección I.4

1 Halle la factorización $A = LU$ de las siguientes matrices:

$$A = \begin{bmatrix} 2 & 1 \\ 6 & 7 \end{bmatrix} \qquad A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

2 Si $a_{11}, \ldots, a_{1n}$ es la primera fila de una matriz A de rango 1 y $a_{11}, \ldots, a_{m1}$ es la primera columna, *encuentre una fórmula para* a_{ij}. Verifique la fórmula en el caso $a_{11} = 2, a_{12} = 3, a_{21} = 4$. ¿Cuándo no funcionará la fórmula?

3 ¿Qué matriz triangular inferior E transforma A en una triangular superior $EA = U$? Multiplique por $E^{-1} = L$ para factorizar A como LU:

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 4 & 2 \\ 6 & 3 & 5 \end{bmatrix}$$

4 **Este problema muestra cómo se multiplican las inversas en cada paso para obtener L.** Esto es claro cuando $A = L$ ya es una matriz triangular inferior con unos en la diagonal. **Entonces $U = I$:**

$$\text{Multiplique } A = \begin{bmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ b & c & 1 \end{bmatrix} \text{ por } E_1 = \begin{bmatrix} 1 & & \\ -a & 1 & \\ -b & 0 & 1 \end{bmatrix} \text{ y entonces } E_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -c & 1 \end{bmatrix}.$$

(a) Multiplique $E_2 E_1$ para encontrar la matriz E que produce $EA = I$.

(b) Multiplique $E_1^{-1} E_2^{-1}$ para encontrar la matriz $A = L$.

Los multiplicadores a, b, c están mezclados en $E = L^{-1}$ pero están perfectos en L.

5 *La factorización $A = LU$ no es posible cuando aparece un cero en una posición pivote* (es necesario que los pivotes sean distintos de cero en U). Demuestre por qué en estos ejemplos las descomposiciones LU son imposibles:

$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \ell & 1 \end{bmatrix} \begin{bmatrix} d & e \\ 0 & f \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & \\ \ell & 1 & \\ m & n & 1 \end{bmatrix} \begin{bmatrix} d & e & g \\ & f & h \\ & & i \end{bmatrix}.$$

Estas matrices necesitan un intercambio de filas utilizando una *matriz de permutación* P.

6 ¿Qué número c hace que sea cero la segunda posición pivote? Se necesita un intercambio de filas y $A = LU$ no será posible. ¿Qué c produce un cero en la tercera posición pivote? Entonces un intercambio de filas no sirve y la eliminación falla:

$$A = \begin{bmatrix} 1 & c & 0 \\ 2 & 4 & 1 \\ 3 & 5 & 1 \end{bmatrix}.$$

7 (*Recomendado*) Calcule L y U para la matriz simétrica A:

$$A = \begin{bmatrix} a & a & a & a \\ a & b & b & b \\ a & b & c & c \\ a & b & c & d \end{bmatrix}.$$

Encuentre condiciones sobre a, b, c, d para obtener $A = LU$ con cuatro pivotes no nulos.

8 Las *matrices tridiagonales* tienen ceros en todas las posiciones excepto en la diagonal principal y las dos diagonales adyacentes. Factorice estas matrices como $A = LU$. Por simetría resultará $A = LDL^{\mathrm{T}}$:

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad \text{y} \quad A = \begin{bmatrix} a & a & 0 \\ a & a+b & b \\ 0 & b & b+c \end{bmatrix}.$$

9 *Fácil pero importante.* Si A tiene pivotes $5, 9, 3$ sin hacer intercambio de filas, ¿Cuales son los pivotes para la submatriz superior izquierda de 2 por 2 A_2 (es decir, la matriz que resulta de quitarle a A la fila 3 y la columna 3)?

10 ¿Qué matrices invertibles permiten la descomposición $A = LU$ (eliminación sin intercambio de filas)? ¡*Buena pregunta*! Observe cada una de las submatrices cuadradas superiores izquierdas $A_1, A_2, \ldots, A_n$.

Todas las submatrices superiores izquierdas A_k deben ser invertibles: la de tamaño 1 por 1, la de 2 por 2, ..., la de n por n.

Explique la respuesta: A_k se factoriza como _____ ya que $LU = \begin{bmatrix} L_k & 0 \\ * & * \end{bmatrix} \begin{bmatrix} U_k & * \\ 0 & * \end{bmatrix}$.

11 En algunas aplicaciones de ciencia de datos, el primer pivote es el *mayor número* $|a_{ij}|$ *de* A. La fila i se convierte en la primera fila pivote $\boldsymbol{u}_1^*$. La columna j es la primera columna pivote. Se divide esta columna por a_{ij} de manera que $\boldsymbol{\ell}_1$ tiene un 1 en la fila i. Luego se resta $\boldsymbol{\ell}_1 \boldsymbol{u}_1^*$ de A.

En este ejemplo $a_{22} = 4$ es el primer pivote ($i = j = 2$). Dividiendo por 4 se obtiene $\boldsymbol{\ell}_1$:

$$\begin{bmatrix} 1 & 2 \\ 3 & \mathbf{4} \end{bmatrix} = \begin{bmatrix} 1/2 \\ 1 \end{bmatrix} \begin{bmatrix} 3 & \mathbf{4} \end{bmatrix} + \begin{bmatrix} -1/2 & 0 \\ 0 & 0 \end{bmatrix} = \boldsymbol{\ell}_1 \boldsymbol{u}_1^* + \boldsymbol{\ell}_2 \boldsymbol{u}_2^* = \begin{bmatrix} 1/2 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 3 & 4 \\ -1/2 & 0 \end{bmatrix}$$

Para esta A, L y U contienen permutaciones. P_1 intercambia las filas para obtener L. P_2 intercambia las columnas para obtener la matriz triangular superior U. Entonces $\boldsymbol{P_1 A P_2 = LU}$.

Permutadas de antemano $\quad \boldsymbol{P_1 A P_2} = \begin{bmatrix} 1 & 0 \\ 1/2 & 1 \end{bmatrix} \begin{bmatrix} 4 & 3 \\ 0 & -1/2 \end{bmatrix} = \begin{bmatrix} \mathbf{4} & \mathbf{3} \\ \mathbf{2} & \mathbf{1} \end{bmatrix}$

Para $A = \begin{bmatrix} \mathbf{1} & \mathbf{3} \\ \mathbf{2} & \mathbf{4} \end{bmatrix}$, aplique pivoteo completo para producir $P_1 A P_2 = LU$.

12 Si la matriz baja y ancha A tiene $m < n$, ¿Cómo se ve a partir del proceso de eliminación que hay soluciones no nulas de $A\boldsymbol{x} = \boldsymbol{0}$? ¿Qué sabe acerca de la dimensión de ese *espacio nulo de* A que contiene todas las soluciones $\boldsymbol{x}$? La dimensión del espacio nulo es al menos _____.

Sugerencia: primero construya una matriz A de 2 por 3 y hágase esas preguntas sobre A.

I.5. Matrices ortogonales y subespacios

La palabra **ortogonal** aparece por todos lados en álgebra lineal. Significa *perpendicular*. Su uso se extiende más allá del concepto de ángulo entre dos vectores. Algunas extensiones de esta idea clave son :

1. **Vectores ortogonales $\boldsymbol{x}$, $\boldsymbol{y}$.** La condición es $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} = x_1y_1 + \cdots + x_ny_n = 0$.
   Si $\boldsymbol{x}$ e $\boldsymbol{y}$ tienen componentes complejas, cambie a $\overline{\boldsymbol{x}}^{\mathrm{T}}\boldsymbol{y} = \overline{x}_1y_1 + \cdots + \overline{x}_ny_n = 0$.

2. **Base ortogonal para un subespacio :** cada par de vectores de la base cumple $\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{v}_j = 0$.
   Base ortonormal : base ortogonal de **vectores unitarios** : cada $\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{v}_i = 1$ (magnitud o longitud 1).
   De ortogonal a ortonormal, solo divida cada vector de las base $\boldsymbol{v}_i$ por su longitud $||\boldsymbol{v}_i||$.

3. **Subespacios ortogonales R y N.** Todo vector en el espacio fila **R** es ortogonal a todo vector de **N.** ¡ Preste atención ! El espacio fila y el espacio nulo son ortogonales :

$$\begin{array}{c} A\boldsymbol{x} = \mathbf{0} \text{ significa} \\ \text{que } \textbf{cada fila} \cdot \boldsymbol{x} = \mathbf{0} \end{array} \qquad \begin{bmatrix} \textbf{fila 1 de } \boldsymbol{A} \\ \vdots \\ \textbf{fila } \boldsymbol{m} \textbf{ de } \boldsymbol{A} \end{bmatrix} \begin{bmatrix} \\ \boldsymbol{x} \\ \\ \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}. \tag{1}$$

   Toda fila (y toda combinación de filas) es ortogonal a todo $\boldsymbol{x}$ del espacio nulo.

4. **Matrices $\boldsymbol{Q}$ delgadas y altas con columnas ortonormales** : $\boldsymbol{Q^{\mathrm{T}}Q = I}$.

$$\boldsymbol{Q^{\mathrm{T}}Q} = \begin{bmatrix} \text{—} & \boldsymbol{q_1}^{\mathrm{T}} & \text{—} \\ & \vdots & \\ \text{—} & \boldsymbol{q_n^{\mathrm{T}}} & \text{—} \end{bmatrix} \begin{bmatrix} \\ \boldsymbol{q}_1 \cdots \boldsymbol{q}_n \\ \\ \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \boldsymbol{I} \tag{2}$$

   Si multiplicamos por esta Q a cualquier vector $\boldsymbol{x}$, la longitud del vector no cambia :

$$||Q\boldsymbol{x}|| = ||\boldsymbol{x}|| \text{ porque } (Q\boldsymbol{x})^{\mathrm{T}}(Q\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}Q^{\mathrm{T}}Q\,\boldsymbol{x} = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} \tag{3}$$

   Si $m > n$ las m filas no pueden ser ortogonales en $\mathbf{R}^n$. Para las matrices delgadas y altas se tiene que $Q\,Q^{\mathrm{T}} \neq I$.

5. **Las "matrices ortogonales" son cuadradas con columnas ortonormales : $\boldsymbol{Q^{\mathrm{T}} = Q^{-1}}$.**
   Para matrices cuadradas, $\boldsymbol{Q^{\mathrm{T}}Q = I}$ nos lleva a $\boldsymbol{QQ^{\mathrm{T}} = I}$
   Para matrices cuadradas Q, la inversa a izquierda Q^{T} es también una inversa a derecha de Q.
   Las columnas de esta matriz ortogonal de n por n son una base ortonormal de $\mathbf{R}^n$.
   Las filas de Q son una base ortonormal (probablemente distinta) de $\mathbf{R}^n$.
   El nombre "matriz ortogonal" debería ser en realidad "matriz ortonormal".

En la siguiente página daremos ejemplos de **vectores**, **bases**, **subespacios** y **matrices** ortogonales.

1. Vectores ortogonales. La igualdad $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} = 0$ se conecta con los triángulos rectángulos a través de $c^2 = a^2 + b^2$:

Teorema de Pitágoras para triángulos rectángulos $$||\boldsymbol{x} - \boldsymbol{y}||^2 = ||\boldsymbol{x}||^2 + ||\boldsymbol{y}||^2. \tag{4}$$

El lado izquierdo es $(\boldsymbol{x} - \boldsymbol{y})^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{y})$. Esto se expande a $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}$. Cuando los últimos dos términos son cero, tenemos la ecuación (4) : $\boldsymbol{x} = (1, 2, 2)$ e $\boldsymbol{y} = (2, 1, -2)$ cumplen $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} = 0$. La hipotenusa es $\boldsymbol{x} - \boldsymbol{y} = (-1, 1, 4)$. Entonces Pitágoras dice que $\mathbf{18} = \mathbf{9} + \mathbf{9}$.

Los productos punto $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}$ e $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}$ son siempre iguales a $||\boldsymbol{x}||\,||\boldsymbol{y}||\cos\theta$, donde θ es el ángulo entre $\boldsymbol{x}$ e $\boldsymbol{y}$. Entonces en todos los casos tenemos la Ley de los Cosenos $c^2 = a^2 + b^2 - 2ab\cos\theta$:

Ley de los cosenos $$||\boldsymbol{x} - \boldsymbol{y}||^2 = ||\boldsymbol{x}||^2 + ||\boldsymbol{y}||^2 - 2\,||\boldsymbol{x}||\,||\boldsymbol{y}||\cos\theta. \tag{5}$$

Los vectores ortogonales cumplen $\cos\theta = 0$ y ese último término desaparece.

2. Base ortogonal. La "base estándar" es ortogonal (e incluso ortonormal) en $\mathbf{R}^n$:

$$\text{Base estándar } \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k} \text{ en } \mathbf{R}^3 \qquad \boldsymbol{i} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \boldsymbol{j} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad \boldsymbol{k} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Aquí tenemos tres matrices de Hadamard H_2, H_4, H_8 que contienen bases ortogonales de $\mathbf{R}^2$, $\mathbf{R}^4$, $\mathbf{R}^8$.

Matrices de Hadamard
Columnas ortogonales
tamaño 2, 4 y 8

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \qquad \begin{bmatrix} H_4 & H_4 \\ H_4 & -H_4 \end{bmatrix}$$

¿Son matrices ortogonales ? *No*. Las magnitudes de las columnas son $\sqrt{2}, \sqrt{4}, \sqrt{8}$. Si dividimos por esas magnitudes, nos encontraremos al principio de una lista infinita : bases ortonormales en $2, 4, 8, 16, 32, \ldots$ dimensiones.

La **conjetura de Hadamard** dice que hay una matriz de ± 1 con columnas ortogonales cuando n es múltiplo de 4. Wikipedia dice que $n = 668$ es el menor de esos tamaños sin una matriz de Hadamard conocida. La construcción para $n = 16, 32, \ldots$ sigue el patrón mostrado arriba.

He aquí un dato clave : **todo subespacio de $\mathbf{R}^n$ tiene una base ortogonal**. Piense en un plano en el espacio tridimensional $\mathbf{R}^3$. El plano contiene dos vectores independientes, $\boldsymbol{a}$ y $\boldsymbol{b}$. Para obtener una base ortogonal, restamos de $\boldsymbol{b}$ su componente en la dirección de $\boldsymbol{a}$:

Base ortogonal $\boldsymbol{a}$ y $\boldsymbol{c}$ $$\boldsymbol{c} = \boldsymbol{b} - \frac{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{b}}{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{a}}\,\boldsymbol{a}. \tag{6}$$

El producto interno $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{c}$ es $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{b} - \boldsymbol{a}^{\mathrm{T}}\boldsymbol{b} = 0$. El concepto de "ortogonalización" se puede aplicar sin importar la cantidad de vectores de la base : *la base se convierte en una base ortogonal*. Esa es la idea de Gram-Schmidt en la Sección II.2.

3. Subespacios ortogonales. La ecuación (1) miraba a $A\boldsymbol{x} = \mathbf{0}$, donde cada fila de A multiplica a ese vector $\boldsymbol{x}$ del espacio nulo. Así cada fila (y todas las combinaciones de filas) serán ortogonales a $\boldsymbol{x}$ en $\mathbf{N}(A)$. **El espacio fila de $\boldsymbol{A}$ es ortogonal al espacio nulo de $\boldsymbol{A}$.**

$$A\boldsymbol{x} = \begin{bmatrix} \text{fila } 1 \\ \vdots \\ \text{fila } m \end{bmatrix} \begin{bmatrix} \\ \boldsymbol{x} \\ \\ \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \qquad A^{\mathrm{T}}\boldsymbol{y} = \begin{bmatrix} (\text{columna } 1)^{\mathrm{T}} \\ \vdots \\ (\text{columna } n)^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \\ \boldsymbol{y} \\ \\ \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{7}$$

De $A^{\mathrm{T}}\boldsymbol{y} = \mathbf{0}$, se tiene que las columnas de A son todas ortogonales a $\boldsymbol{y}$. Sus combinaciones (el espacio columna entero) también será ortogonal a $\boldsymbol{y}$. **El espacio columna de $\boldsymbol{A}$ es ortogonal al espacio nulo de $\boldsymbol{A^{\mathrm{T}}}$.** Esto produce el "Marco general del Álgebra Lineal" en la Figura I.6.

Observemos que las dimensiones r y $n-r$ suman n. Cada vector $\boldsymbol{v}$ de $\mathbf{R}^n$ tiene una componente $\boldsymbol{v}_r$ del espacio fila y una componente $\boldsymbol{v}_n$ del espacio nulo con $\boldsymbol{v} = \boldsymbol{v}_r + \boldsymbol{v}_n$. Una base del espacio fila (r vectores) junto con una base de espacio nulo ($n - r$ vectores) dan lugar a una base para todo $\mathbf{R}^n$ (n vectores).

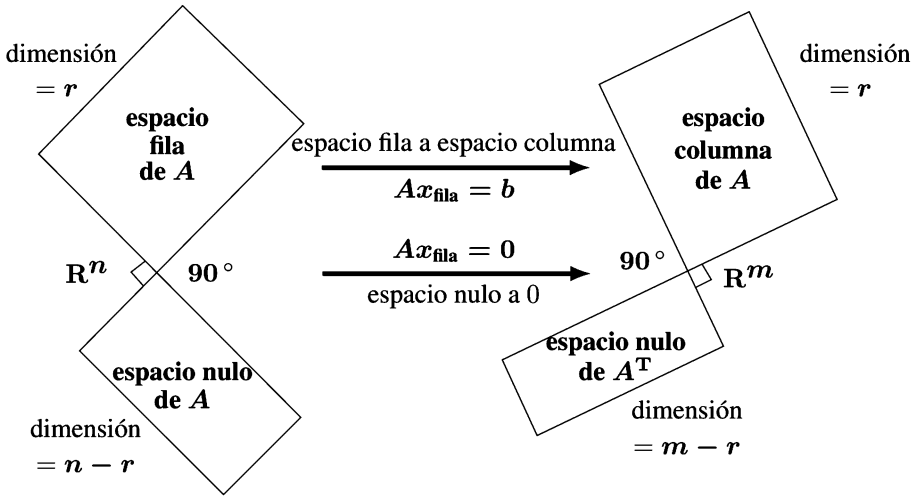


Figura I.6: dos pares de subespacios ortogonales. Las dimensiones suman n y m. **Este es el marco general**: dos subespacios en $\mathbf{R}^n$ y dos subespacios en $\mathbf{R}^m$.

Quiero destacar una gran mejora que se obtiene a partir de la Descomposición en Valores Singulares (SVD). La SVD es el teorema más importante en ciencia de datos. Encuentra bases ortonormales $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r$ para el espacio fila de A y $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r$ para el espacio columna de A. Gram-Schmidt puede hacerlo. Las bases especiales de la SVD tienen la propiedad adicional de que cada par ($\boldsymbol{v}$ y $\boldsymbol{u}$) está conectado por A:

$$\boxed{\textbf{Vectores singulares} \qquad A\boldsymbol{v}_1 = \sigma_1\boldsymbol{u}_1 \qquad A\boldsymbol{v}_2 = \sigma_2\boldsymbol{u}_2 \qquad \cdots \qquad A\boldsymbol{v}_r = \sigma_r\boldsymbol{u}_r.} \tag{8}$$

En la Figura I.6, imaginemos las $\boldsymbol{v}$ a la izquierda y las $\boldsymbol{u}$ a la derecha. Para las bases de la SVD, multiplicar por A lleva una base ortogonal de vectores $\boldsymbol{v}$ a una base ortogonal de vectores $\boldsymbol{u}$.

4. Matriz Q alta y delgada con columnas ortonormales: $Q^{\mathrm{T}}Q = I$.
Aquí tenemos tres Q posibles, una de (3 por 1), una (3 por 2) y una matriz ortogonal Q_3 de (3 por 3).

$$Q_1 = \frac{1}{3}\begin{bmatrix} 2 \\ 2 \\ -1 \end{bmatrix} \qquad Q_2 = \frac{1}{3}\begin{bmatrix} 2 & 2 \\ 2 & -1 \\ -1 & 2 \end{bmatrix} \qquad Q_3 = \frac{1}{3}\begin{bmatrix} 2 & 2 & -1 \\ 2 & -1 & 2 \\ -1 & 2 & 2 \end{bmatrix}. \tag{9}$$

Cada una de esas matrices cumple $Q^{\mathrm{T}}Q = I$. Así Q^{T} es una inversa a izquierda de Q. Solo la última matriz cumple $Q_3 Q_3^{\mathrm{T}} = I$. Entonces Q_3^{T} también es una inversa a derecha. Q_3 es simétrica y ortogonal. Es una reina y también un rey, sin dudas una matriz de la realeza.

Observemos que todas las matrices $P = QQ^{\mathrm{T}}$ cumplen $P^2 = P$:

$$\boldsymbol{P^2} = (QQ^{\mathrm{T}})\,(QQ^{\mathrm{T}}) = Q\,(Q^{\mathrm{T}}Q)\,Q^{\mathrm{T}} = QQ^{\mathrm{T}} = \boldsymbol{P}. \tag{10}$$

En el medio quitamos $Q^{\mathrm{T}}Q = I$. La ecuación $P^2 = P$ indica que P es **una matriz de proyección**.

Si $P^2 = P = P^{\mathrm{T}}$ entonces Pb es la proyección ortogonal de b en el espacio columna de P.

Ejemplo 1 Para proyectar $\boldsymbol{b} = (3,3,3)$ en la recta Q_1, multiplique por $P_1 = Q_1 Q_1^{\mathrm{T}}$.

$$P_1\boldsymbol{b} = \frac{1}{9}\begin{bmatrix} 2 \\ 2 \\ -1 \end{bmatrix}\begin{bmatrix} 2 & 2 & -1 \end{bmatrix}\begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix} = \frac{1}{9}\begin{bmatrix} 2 \\ 2 \\ -1 \end{bmatrix} 9 = \begin{bmatrix} \mathbf{2} \\ \mathbf{2} \\ \mathbf{-1} \end{bmatrix} = \begin{matrix} \text{proyección} \\ \text{sobre una recta} \end{matrix}$$

Esa matriz divide $\boldsymbol{b}$ en dos partes perpendiculares: la proyección $P_1\boldsymbol{b}$ y el error $\boldsymbol{e} = (I - P_1)\,\boldsymbol{b}$.

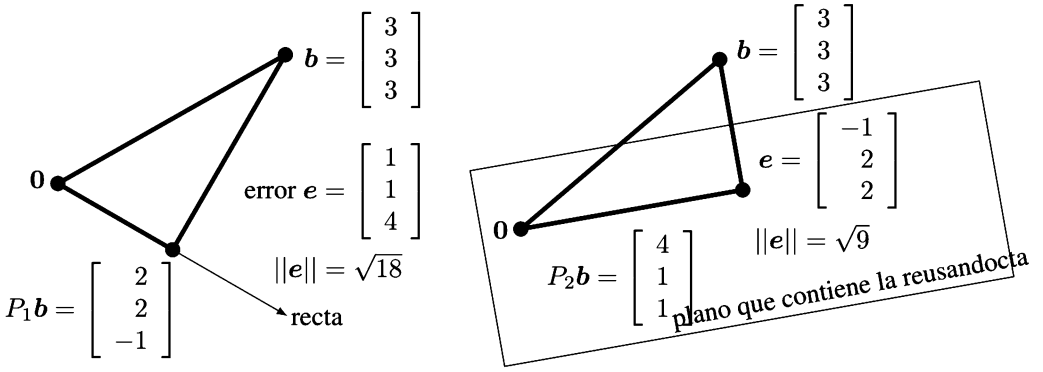


Figura I.7: proyección de $\boldsymbol{b}$ sobre una recta usando $P_1 = Q_1 Q_1^{\mathrm{T}}$ y sobre un plano con $P_2 = Q_2 Q_2^{\mathrm{T}}$.

Ahora proyecte el mismo $\boldsymbol{b} = (3,3,3)$ en el espacio columna Q_2 (un plano). El vector de error $\boldsymbol{b} - P_2\,\boldsymbol{b}$ tiene menor longitud que $\boldsymbol{b} - P_1\boldsymbol{b}$ porque el plano contiene la recta.

$$P_2\,\boldsymbol{b} = \frac{1}{9}\begin{bmatrix} 2 & 2 \\ 2 & -1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} 2 & 2 & -1 \\ 2 & -1 & 2 \end{bmatrix}\begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix} = \frac{1}{9}\begin{bmatrix} 2 & 2 \\ 2 & -1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} 9 \\ 9 \end{bmatrix} = \begin{bmatrix} \mathbf{4} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}$$

Pregunta: ¿Qué vector es $P_3\,\boldsymbol{b} = Q_3 Q_3^{\mathrm{T}}\,\boldsymbol{b}$? Ahora se está proyectando $\boldsymbol{b}$ en todo el espacio $\mathbf{R}^3$. *Respuesta*: $P_3\,\boldsymbol{b} = \boldsymbol{b}$. De hecho, ¡$P_3 = Q_3 Q_3^{\mathrm{T}} =$ matriz identidad ! **El error e ahora es cero**. Las proyecciones constituyen el corazón de los "cuadrados mínimos" en la Sección II.2.

5. Matrices ortogonales: Ahora Q es *cuadrada*: $Q^{\mathrm{T}}Q = I$ y $QQ^{\mathrm{T}} = I$. Así que $\boldsymbol{Q^{-1} = Q^{\mathrm{T}}}$. Estas Q son muy importantes. Para 2 por 2, son **rotaciones** del plano o **reflexiones**.

Cuando todo el plano rota alrededor de $(0,0)$, las longitudes no cambian. Los ángulos entre vectores no cambian. Las columnas de Q son vectores unitarios ortogonales, con $\cos^2\theta + \operatorname{sen}^2\theta = 1$:

$$\boldsymbol{Q_{\text{rotar}}} = \begin{bmatrix} \boldsymbol{\cos\theta} & \boldsymbol{-\operatorname{sen}\theta} \\ \boldsymbol{\operatorname{sen}\theta} & \boldsymbol{\cos\theta} \end{bmatrix} = \textbf{rotación de un ángulo } \theta. \tag{11}$$

Si multiplicamos una columna por -1, las dos columnas siguen siendo ortogonales de norma 1.

$$\boldsymbol{Q_{\text{reflejar}}} = \begin{bmatrix} \boldsymbol{\cos\theta} & \boldsymbol{\operatorname{sen}\theta} \\ \boldsymbol{\operatorname{sen}\theta} & \boldsymbol{-\cos\theta} \end{bmatrix} = \textbf{reflexión a través de la recta } \frac{\theta}{2}. \tag{12}$$

Ahora Q refleja cada vector de manera espejada. Es una reflexión con determinante -1 en lugar de una rotación con determinante $+1$. El plano x-y o bien rota o bien se refleja.

Es importante saber que la multiplicación de matrices ortogonales produce una matriz ortogonal.

$Q_1 Q_2$ es ortogonal $\qquad (Q_1\,Q_2)^{\mathrm{T}}(Q_1\,Q_2) = Q_2^{\mathrm{T}}\,Q_1^{\mathrm{T}}\,Q_1 Q_2 = Q_2^{\mathrm{T}} Q_2 = I.$

Rotación por rotación = rotación. Reflexión por reflexión = rotación. Rotación por reflexión = *reflexión*. Todo esto también se cumple en $\mathbf{R}^n$.

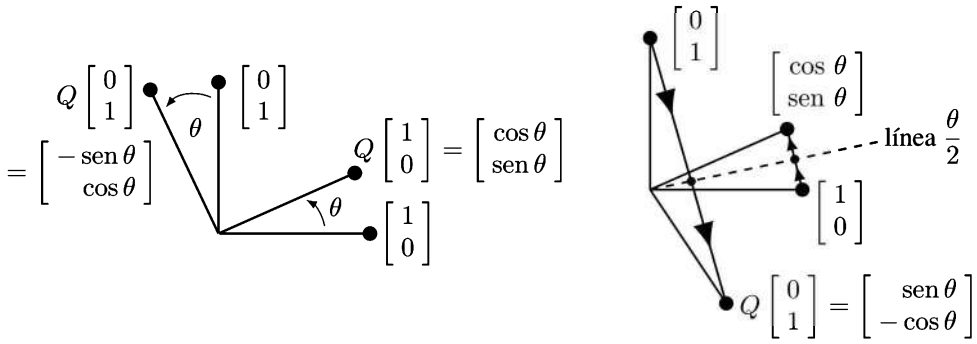


Figura I.8: rota el plano en un ángulo θ. Refleja todo vector "a través" de la recta en ángulo $\theta/2$.

La figura muestra cómo las columnas de Q se obtienen a partir de $Q \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ y $Q \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

Base ortogonal = Ejes ortogonales en $\mathbf{R}^n$

Supongamos que la matriz ortogonal Q de n por n tiene columnas $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n$. Esos vectores unitarios son una base del espacio n-dimensional $\mathbf{R}^n$. Todo vector $\boldsymbol{v}$ se puede escribir como una combinación de los vectores de la base (los $\boldsymbol{q}$) :

$$\boldsymbol{v} = c_1\boldsymbol{q}_1 + \cdots + c_n\boldsymbol{q}_n \tag{13}$$

Esos $c_1\boldsymbol{q}_1$, $c_2\boldsymbol{q}_2$ y $c_n\boldsymbol{q}_n$ son las componentes de $\boldsymbol{v}$ en las direcciones de los ejes. ¡Son las proyecciones de $\boldsymbol{v}$ sobre los ejes ! Hay una fórmula sencilla para cada uno de los números c_1 a c_n :

Coeficientes en una base ortonormal

$$\boxed{c_1 = \boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{v} \qquad c_2 = \boldsymbol{q}_2^{\mathrm{T}}\boldsymbol{v} \quad \cdots \quad c_n = \boldsymbol{q}_n^{\mathrm{T}}\boldsymbol{v}} \tag{14}$$

Haremos una prueba vectorial y una prueba matricial. Hagamos un producto punto con $\boldsymbol{q}_1$ en la ecuación (13) :

$$\boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{v} = c_1\boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{q}_1 + \cdots + c_n\boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{q}_n = c_1 \tag{15}$$

Todos los términos son cero excepto $c_1\boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{q}_1 = c_1$. Así que $\boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{v} = c_1$ y luego $\boldsymbol{q}_k^{\mathrm{T}}\boldsymbol{v} = c_k$.

Si escribimos (13) como una ecuación matricial $\boldsymbol{v} = Q\boldsymbol{c}$, **multiplicamos por $\boldsymbol{Q^{\mathrm{T}}}$ para concluir** (14) :

$$Q^{\mathrm{T}}\boldsymbol{v} = Q^{\mathrm{T}}Q\boldsymbol{c} = \boldsymbol{c} \quad \text{da los coeficientes } c_k = \boldsymbol{q}_k^{\mathrm{T}}\boldsymbol{v} \text{ todos de una vez.}$$

Esta es la aplicación clave de las bases ortogonales (por ejemplo la base para las series de Fourier). Cuando la bases es ortonormal, ¡ cada coeficiente c_1 a c_n se puede encontrar por separado !

Reflexiones de Householder

He aquí unos lindos ejemplos de **matrices de reflexiones** $Q = \boldsymbol{H_n}$. Comenzamos con la matriz identidad. Elegimos un vector unitario $\boldsymbol{u}$. Le restamos la matriz $2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$ simétrica de rango uno. Entonces $I - 2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$ será una "Matriz de Householder". Por ejemplo, elegimos $\boldsymbol{u} = (1, 1, \ldots, 1)/\sqrt{n}$.

Ejemplo de Householder

$$\boxed{\boldsymbol{H_n} = \boldsymbol{I} - 2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} = \boldsymbol{I} - \frac{\mathbf{2}}{\boldsymbol{n}}\,\mathbf{unos}(\boldsymbol{n}, \boldsymbol{n}).} \tag{16}$$

Con $\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$, H_n es sin dudas simétrica. Dos reflexiones dan $H^2 = I$ ya que $\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u} = 1$:

$$\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H} = \boldsymbol{H}^{\boldsymbol{2}} = (I - 2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}})\,(I - 2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}) = I - 4\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} + 4\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} = \boldsymbol{I}. \tag{17}$$

Los ejemplos de 3 por 3 y de 4 por 4 son fáciles de recordar, y H_4 es como una matriz de Hadamard :

$$\boldsymbol{H_3} = I - \frac{2}{3}\,\mathbf{unos} = \frac{1}{3}\begin{bmatrix} 1 & -2 & -2 \\ -2 & 1 & -2 \\ -2 & -2 & 1 \end{bmatrix} \qquad \boldsymbol{H_4} = I - \frac{2}{4}\,\mathbf{unos} = \frac{1}{2}\begin{bmatrix} \mathbf{1} & -1 & -1 & -1 \\ -1 & \mathbf{1} & -1 & -1 \\ -1 & -1 & \mathbf{1} & -1 \\ -1 & -1 & -1 & \mathbf{1} \end{bmatrix}$$

La matriz de reflexión de Householder de n por n cumple $\boldsymbol{H_n}\boldsymbol{u} = (I - 2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}})\boldsymbol{u} = \boldsymbol{u} - 2\boldsymbol{u} = \boldsymbol{-u}$. Además $\boldsymbol{H_n}\boldsymbol{w} = \boldsymbol{+w}$ cuando $\boldsymbol{w}$ es perpendicular a $\boldsymbol{u}$. Los "autovalores" de H son -1 (una vez) y $+1$ ($n - 1$ veces). Todas las matrices de reflexiones tienen autovalores -1 y 1.

Problemas de la Sección I.5

1 Si $\boldsymbol{u}$ y $\boldsymbol{v}$ son vectores unitarios ortogonales, demuestre que $\boldsymbol{u}+\boldsymbol{v}$ es ortogonal a $\boldsymbol{u}-\boldsymbol{v}$. ¿Cuáles son las normas o longitudes de esos vectores ?

2 Dibuje dos vectores unitarios $\boldsymbol{u}$ y $\boldsymbol{v}$ que *no sean* ortogonales. Muestre que $\boldsymbol{w}=\boldsymbol{v}-\boldsymbol{u}(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{v})$ es ortogonal a $\boldsymbol{u}$ (y agregue $\boldsymbol{w}$ al gráfico).

3 Dibuje dos vectores $\boldsymbol{u}$ y $\boldsymbol{v}$ cualesquiera a partir del origen $(0,0)$. Complete dos lados más para hacer un paralelogramo con diagonales $\boldsymbol{w}=\boldsymbol{u}+\boldsymbol{v}$ y $\boldsymbol{z}=\boldsymbol{u}-\boldsymbol{v}$. Muestre que $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}+\boldsymbol{z}^{\mathrm{T}}\boldsymbol{z}$ es igual a $2\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}+2\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}$.

4 Propiedad clave de toda matriz ortogonal: $||Q\boldsymbol{x}||^2 = ||\boldsymbol{x}||^2$ para todo vector $\boldsymbol{x}$. Más aun, muestre que $(Q\boldsymbol{x})^{\mathrm{T}}(Q\boldsymbol{y})=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}$ para todo vector $\boldsymbol{x}$ y $\boldsymbol{y}$. Entonces Q *no modifica las longitudes ni los ángulos*. **¡Los cálculos con Q nunca se dan error de overflow !**

5 Si Q es ortogonal, ¿cómo sabe que Q es invertible y que Q^{-1} también es ortogonal ?
Si $Q_1^{\mathrm{T}}=Q_1^{-1}$ y $Q_2^{\mathrm{T}}=Q_2^{-1}$, demuestre que Q_1Q_2 también es una matriz ortogonal.

6 Una **matriz de permutación** tiene las mismas columnas que la matriz identidad (en algún orden). *Explique por qué esta y todas las matrices de permutación son ortogonales*:

$P=\begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\1&0&0&0\end{bmatrix}$ tiene columnas ortonormales entonces $P^{\mathrm{T}}P=$ _____ y $P^{-1}=$ _____ .

Cuando una matriz es simétrica u ortogonal, **tendrá autovectores ortogonales**. Esta es la fuente más importante de vectores ortogonales en matemática aplicada.

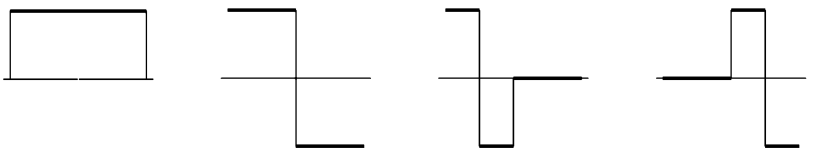
7 Cuatro autovectores de esa matriz P son $\boldsymbol{x}_1=(1,1,1,1)$, $\boldsymbol{x}_2=(1,i,i^2,i^3)$, $\boldsymbol{x}_3=(1,i^2,i^4,i^6)$, y $\boldsymbol{x}_4=(1,i^3,i^6,i^9)$. Multiplique P por cada uno de esos vectores para encontrar $\lambda_1,\lambda_2,\lambda_3,\lambda_4$. Los autovectores son las columnas de la **matriz de Fourier $\boldsymbol{F}$** de 4 por 4 .

Muestre que $Q=\dfrac{F}{2}=\dfrac{1}{2}\begin{bmatrix}1&1&1&1\\1&i&-1&i^3\\1&i^2&1&i^6\\1&i^3&-1&i^9\end{bmatrix}$ tiene columnas ortonormales : $\boldsymbol{\overline{Q}^{\mathrm{T}}Q=I}$

8 **Las ondículas de Haar son vectores ortogonales (columnas de W) que usan solo $1,-1,0$.**

$n=4$ $\quad W=\begin{bmatrix}1&1&1&0\\1&1&-1&0\\1&-1&0&1\\1&-1&0&-1\end{bmatrix}$

Encuentre $\boldsymbol{W^{\mathrm{T}}W}$ y $\boldsymbol{W^{-1}}$ y las ocho ondículas de Haar para $\boldsymbol{n=8}$.



I.6. Autovalores y autovectores

Los autovectores de A no cambian de dirección cuando los multiplicamos por A. El resultado $A\boldsymbol{x}$ pertenece a la misma recta que el vector de entrada $\boldsymbol{x}$.

$$\boxed{\begin{array}{ll}\boldsymbol{x} = \textbf{autovector de } \boldsymbol{A} & \\ \boldsymbol{\lambda} = \textbf{autovalor de } \boldsymbol{A} & \boldsymbol{A\boldsymbol{x} = \lambda\boldsymbol{x}}\end{array}} \tag{1}$$

El autovector $\boldsymbol{x}$ se multiplicó por su autovalor λ. Lo multiplicamos de nuevo por A, para ver que $\boldsymbol{x}$ **también es un autovector de** $\boldsymbol{A^2}$: $A^2\boldsymbol{x} = \lambda^2\boldsymbol{x}$.

$$\begin{array}{l}\boldsymbol{x} \ = \text{mismo autovector} \\ \boldsymbol{\lambda^2} = \text{cuadrado del autovalor}\end{array} \qquad A(A\boldsymbol{x}) = A(\lambda\boldsymbol{x}) = \lambda(A\boldsymbol{x}) = \lambda^2\boldsymbol{x} \ . \tag{2}$$

Sin dudas $A^k\boldsymbol{x} = \lambda^k\boldsymbol{x}$ para todo $k = 1, 2, 3, \ldots$ y $A^{-1}\boldsymbol{x} = \frac{1}{\lambda}\boldsymbol{x}$ siempre que $\lambda \neq 0$.

Estos autovectores son vectores especiales que dependen de A. La mayoría de las matrices de n por n tienen n autovectores independientes $\boldsymbol{x}_1$ a $\boldsymbol{x}_n$ con n autovalores diferentes λ_1 a λ_n. En ese caso todo vector $\boldsymbol{v}$ n-dimensional será una combinación de los autovectores:

$$\boxed{\begin{array}{ll}\textbf{Cualquier } \boldsymbol{v} & \boldsymbol{v} = c_1\boldsymbol{x}_1 + \cdots + c_n\boldsymbol{x}_n \\ \textbf{Multiplicado por } \boldsymbol{A} & A\boldsymbol{v} = c_1\lambda_1\boldsymbol{x}_1 + \cdots + c_n\lambda_n\boldsymbol{x}_n \\ \textbf{Multiplicado por } \boldsymbol{A^k} & A^k\boldsymbol{v} = c_1\lambda_1^k\boldsymbol{x}_1 + \cdots + c_n\lambda_n^k\boldsymbol{x}_n\end{array}} \tag{3}$$

Aquí podemos ver por qué los autovalores y autovectores son útiles. Nos permiten ver el corazón de una matriz. Si $|\lambda_1| > 1$ entonces la componente $c_1\lambda_1^n\boldsymbol{x}_1$ crecerá cuando n crece. Si $|\lambda_2| < 1$ entonces esa componente $c_2\lambda_2^n\boldsymbol{x}_2$ gradualmente desaparecerá. **¡Seguimos cada autovector por separado!**

Ejemplo 1 $S = \begin{bmatrix}\mathbf{2} & \mathbf{1} \\ \mathbf{1} & \mathbf{2}\end{bmatrix}$ tiene autovectores $S\begin{bmatrix}1 \\ 1\end{bmatrix} = \mathbf{3}\begin{bmatrix}1 \\ 1\end{bmatrix}$ y $S\begin{bmatrix}1 \\ -1\end{bmatrix} = \begin{bmatrix}1 \\ -1\end{bmatrix}$.

Entonces $\boldsymbol{\lambda_1 = 3}$ y $\boldsymbol{\lambda_2 = 1}$. **Las potencias** $\boldsymbol{S^k}$ **crecen como** $\boldsymbol{3^k}$**.** Esos autovalores y autovectores poseen cuatro propiedades destacables:

(**Traza de** $\boldsymbol{S}$) La suma $\lambda_1 + \lambda_2 = 3 + 1$ es igual a la suma de la diagonal $2 + 2 = 4$

(**Determinante**) El producto $\lambda_1\lambda_2 = (3)(1) = 3$ es igual al determinante $4 - 1$

(**Autovalores reales**) Las matrices simétricas $S = S^{\mathrm{T}}$ siempre tienen autovalores reales

(**Autovectores ortogonales**) Si $\lambda_1 \neq \lambda_2$ entonces $\boldsymbol{x}_1 \cdot \boldsymbol{x}_2 = 0$. Aquí $(1, 1) \cdot (1, -1) = 0$.

Las matrices simétricas S son en cierto sentido como números reales (cada λ es real). Las matrices ortogonales Q son como números complejos $e^{i\theta} = \cos\theta + i\,\mathrm{sen}\,\theta$ de magnitud 1 (todo $|\lambda| = 1$). Las potencias de Q no crecen ni decaen porque $Q^2, Q^3, \ldots$ también son matrices ortogonales.

Ejemplo 2 La rotación $Q = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ tiene autovalores imaginarios $\boldsymbol{i}$ y $\boldsymbol{-i}$:

$$Q\begin{bmatrix} 1 \\ -i \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ -i \end{bmatrix} = (\boldsymbol{i})\begin{bmatrix} 1 \\ -i \end{bmatrix} \quad \text{y} \quad Q\begin{bmatrix} 1 \\ i \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ i \end{bmatrix} = (\boldsymbol{-i})\begin{bmatrix} 1 \\ i \end{bmatrix}.$$

Claramente $\lambda_1 + \lambda_2 = i - i$ coincide con la traza $0 + 0$ de la diagonal principal de Q, y $(\lambda_1)(\lambda_2) = (i)(-i)$ coincide con el determinante de $Q = 1$. Los autovectores de Q siguen siendo ortogonales cuando consideramos (como es debido) el producto punto de los vectores complejos. Cambiamos todo i en $\boldsymbol{x}_1$ a $-i$. Esto produce su conjugado $\overline{\boldsymbol{x}}_1$.

$$\overline{\boldsymbol{x}}_1^{\mathrm{T}}\boldsymbol{x}_2 = \begin{bmatrix} 1 & i \end{bmatrix}\begin{bmatrix} 1 \\ i \end{bmatrix} = 1 + i^2 = \mathbf{0} \text{ : } \textbf{autovectores ortogonales}.$$

Advertencias sobre autovalores y autovectores

Los autovalores de $A + B$ no suelen ser $\lambda(A)$ más $\lambda(B)$.

Los autovalores de AB no suelen ser $\lambda(A)$ por $\lambda(B)$.

Un autovalor doble $\lambda_1 = \lambda_2$ *puede o no* tener dos autovectores independientes.

Los autovectores de una matriz A real son ortogonales si y solo si $A^{\mathrm{T}}A = AA^{\mathrm{T}}$.

La matriz A también controla al sistema de ecuaciones diferenciales lineales $d\boldsymbol{u}/dt = A\boldsymbol{u}$. El sistema comienza en el vector inicial $\boldsymbol{u}(0)$ cuando $t = 0$. Todo autovector crece o decae u oscila dependiendo de su autovalor λ. Las potencias λ^n se reemplazan por exponenciales $e^{\lambda t}$:

$$\boxed{\begin{array}{ll} \textbf{Vector inicial} & \boldsymbol{u(0) = c_1 x_1 + \cdots + c_n x_n} \\ \textbf{Vector solución} & \boldsymbol{u(t) = c_1 e^{\lambda_1 t} x_1 + \cdots + c_n e^{\lambda_n t} x_n} \end{array}} \tag{4}$$

La diferencia entre crecimiento y decrecimiento ahora se decide mirando si $\operatorname{Re}\lambda > 0$ o $\operatorname{Re}\lambda < 0$, en lugar de $|\lambda| > 1$ o $|\lambda| < 1$. La parte real de $\lambda = a + ib$ es $\operatorname{Re}\lambda = a$. El valor absoluto de $e^{\lambda t}$ es e^{at}. El otro factor $e^{ibt} = \cos bt + i \operatorname{sen} bt$ tiene $\cos^2 bt + \operatorname{sen}^2 bt = 1$. Esta parte oscila mientras e^{at} crece o decrece.

Cálculo de autovalores (a mano)

Observemos que $A\boldsymbol{x} = \lambda\boldsymbol{x}$ es lo mismo que $(A - \lambda I)\boldsymbol{x} = 0$. Entonces $A - \lambda I$ no es invertible: esta matriz es *singular*. **El determinante de $\boldsymbol{A - \lambda I}$ debe ser cero**. Esto nos da una ecuación de grado n para λ, y esta ecuación $\det(A - \lambda I) = 0$ tiene n raíces. Aquí $n = 2$ y $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ tiene dos autovalores:

$$\boxed{\textbf{Determinante de } \boldsymbol{A - \lambda I} = \begin{vmatrix} \boldsymbol{a - \lambda} & \boldsymbol{b} \\ \boldsymbol{c} & \boldsymbol{d - \lambda} \end{vmatrix} = \boldsymbol{\lambda^2 - (a + d)\lambda + (ad - bc) = 0}}$$

Esta ecuación cuadrática puede factorizarse fácilmente en $(\lambda - \lambda_1)(\lambda - \lambda_2)$. La "fórmula cuadrática" siempre dará las dos raíces λ_1 y λ_2 de nuestra ecuación, usando el signo $+$ o el signo $-$.

$$\boldsymbol{\lambda} = \frac{1}{2}\left[a + d \pm \sqrt{(a+d)^2 - 4\,(ad - bc)}\right] = \frac{1}{2}\left[\boldsymbol{a + d \pm \sqrt{(a-d)^2 + 4bc}}\right].$$

Podemos ver que $\lambda_1 + \lambda_2$ es igual a $a + d$ (la traza de la matriz). Los términos $\pm$ raíz cuadrada se cancelan.

Observemos también que los autovalores son reales cuando A es simétrica ($b = c$), pues en ese caso estamos tomando la raíz cuadrada de un número no negativo para encontrar λ. Cuando bc es muy negativo, ¡ los autovalores y autovectores se vuelven complejos !

Ejemplo 3 Encuentre los autovalores y autovectores de $A = \begin{bmatrix} \mathbf{8} & \mathbf{3} \\ \mathbf{2} & \mathbf{7} \end{bmatrix}$: **no simétrica.**

El determinante de $A - \lambda I$ es $\begin{vmatrix} 8-\lambda & 3 \\ 2 & 7-\lambda \end{vmatrix} = \lambda^2 - 15\lambda + 50 = \boldsymbol{(\lambda - 10)\,(\lambda - 5)}$.

$\boldsymbol{\lambda_1 = 10}$ tiene $\begin{bmatrix} 8-10 & 3 \\ 2 & 7-10 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ **Autovector** $\boldsymbol{x}_1 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$

$\boldsymbol{\lambda_2 = 5}$ tiene $\begin{bmatrix} 8-5 & 3 \\ 2 & 7-5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ **Autovector** $\boldsymbol{x}_2 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

$\mathbf{10 + 5 = 8 + 7}$. Estos autovectores no son ortogonales. Aumente 3 a 30 para tener λ complejos.

Pregunta : Si A se desplaza a $A + sI$, ¿qué ocurre con los $\boldsymbol{x}$ y los λ ?
Respuesta : Los autovectores $\boldsymbol{x}$ se mantienen iguales. Todo autovalor λ se desplaza en s :

$$\boxed{\begin{array}{c}\textbf{Desplazamiento en } \boldsymbol{A} \Rightarrow \textbf{Desplazamiento en cada } \boldsymbol{\lambda} \\ (\boldsymbol{A + sI})\,\boldsymbol{x} = \lambda\boldsymbol{x} + s\boldsymbol{x} = (\boldsymbol{\lambda + s})\,\boldsymbol{x}\end{array}} \tag{5}$$

Matrices semejantes

Para toda matriz invertible B, los autovalores de BAB^{-1} son los mismos que los autovalores de A. Los autovectores $\boldsymbol{x}$ de A se multiplican por B para obtener los autovectores $B\boldsymbol{x}$ de BAB^{-1} :

$$\boxed{\text{Si } A\boldsymbol{x} = \lambda\boldsymbol{x} \text{ entonces } (BAB^{-1})\,(B\boldsymbol{x}) = BA\boldsymbol{x} = B\lambda\boldsymbol{x} = \lambda(B\boldsymbol{x}).} \tag{6}$$

Las matrices BAB^{-1} (para cada B invertible) son **"semejantes" a $\boldsymbol{A}$: mismos autovalores.**

Usamos esta idea para calcular los autovalores de matrices grandes (cuando el determinante de $A - \lambda I$ es imposible de formular). La idea es que BAB^{-1} gradualmente se vuelva una matriz triangular. Los autovalores no cambian y gradualmente aparecen en la diagonal principal de BAB^{-1} :

$$\textbf{Los autovalores de cualquier matriz triangular } \begin{bmatrix} \boldsymbol{a} & \boldsymbol{b} \\ \mathbf{0} & \boldsymbol{d} \end{bmatrix} \textbf{ son } \boldsymbol{\lambda_1 = a} \textbf{ y } \boldsymbol{\lambda_2 = d.} \tag{7}$$

Podrá notar que $A - aI$ y $A - dI$ tienen determinante cero. Entonces a y d son los autovalores de esta matriz triangular.

Diagonalizar una Matriz

Supongamos que A tiene un conjunto completo de n autovectores independientes (la mayoría de las matrices lo tienen, pero no todas). Colocamos esos autovectores $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ en una matriz X invertible. Luego multiplicamos AX, columna por columna, para obtener las columnas $\lambda_1 \boldsymbol{x}_1$ a $\lambda_n \boldsymbol{x}_n$. ¡ Importante ! Esa matriz se factoriza en X por Λ.

$$A \begin{bmatrix} & & \\ \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_n \\ & & \end{bmatrix} = \begin{bmatrix} & & \\ A\boldsymbol{x}_1 & \cdots & A\boldsymbol{x}_n \\ & & \end{bmatrix} = \begin{bmatrix} & & \\ \lambda_1 \boldsymbol{x}_1 & \cdots & \lambda_n \boldsymbol{x}_n \\ & & \end{bmatrix} = \begin{bmatrix} & & \\ \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_n \\ & & \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}. \quad (8)$$

La matriz de autovalores Λ va a la derecha de X, porque los λ en Λ multiplican a las columnas de X. Esa ecuación $AX = X\Lambda$ nos dice que $\boldsymbol{A = X\Lambda X^{-1}}$. Si conocemos los autovalores y autovectores, conocemos la matriz A. Además, podemos calcular con facilidad las potencias de A:

$\boldsymbol{\Lambda}$ = matriz diagonal de autovalores	$A = X\Lambda X^{-1}$
$\boldsymbol{X}$ = matriz invertible de autovectores	$A^2 = (X\Lambda X^{-1})(X\Lambda X^{-1}) = X\Lambda^2 X^{-1}$

$A = X\Lambda X^{-1}$ en el Ejemplo 3 $\begin{bmatrix} 8 & 3 \\ 2 & 7 \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 10 & \\ & 5 \end{bmatrix} \frac{1}{5} \begin{bmatrix} 1 & 1 \\ 2 & -3 \end{bmatrix} =$ (**autovectores**) por ($\boldsymbol{\lambda}$) por (**autovectores a izquierda**)

La ecuación $A^k = X\Lambda^k X^{-1}$ nos dice lo que ya sabíamos. Los autovalores de A^k son $\lambda_1^k, \ldots, \lambda_n^k$. Los autovectores de A^k son los mismos que los autovectores de A. Tres pasos para calcular $A^k \boldsymbol{v}$.

Paso 1 : $\boldsymbol{X^{-1}v}$	Este nos da los c en $\boldsymbol{v} = c_1\boldsymbol{x}_1 + \cdots + c_n\boldsymbol{x}_n$
Paso 2 : $\boldsymbol{\Lambda^k X^{-1}v}$	Este nos da los λ en $c_1\lambda_1^k\boldsymbol{x}_1 + \cdots + c_n\lambda_n^k\boldsymbol{x}_n$
Paso 3 : $\boldsymbol{X\Lambda^k X^{-1}v}$	Este suma los términos en $A^k\boldsymbol{v} = c_1\lambda_1^k\boldsymbol{x}_1 + \cdots + c_n\lambda_n^k\boldsymbol{x}_n$

Ejemplo 4 Si dividimos el Ejemplo 3 por 10, todos los autovalores se dividen por 10. Entonces $\lambda_1 = 1$ and $\lambda_2 = \frac{1}{2}$. En este caso A es una **Matriz de Markov, con columnas no negativas que suman 1.**

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{0.8} & \mathbf{0.3} \\ \mathbf{0.2} & \mathbf{0.7} \end{bmatrix} \quad \begin{array}{l} A^k\boldsymbol{v} = c_1(1)^k\boldsymbol{x}_1 + c_2\left(\frac{1}{2}\right)^k\boldsymbol{x}_2 \\ \textbf{A medida que } \boldsymbol{k} \textbf{ aumenta, } \boldsymbol{A^k v} \textbf{ se acerca a } \boldsymbol{c_1 x_1} \textbf{ = estado estacionario} \end{array}$$

Podemos seguir cada autovector por separado. Su crecimiento o decrecimiento depende del autovalor λ. La acción de toda la matriz A se divide en acciones simples (solo multiplicar por λ) sobre cada autovector. **Para resolver una ecuación diferencial $\boldsymbol{du/dt = Au}$ multiplicamos cada autovector por $\boldsymbol{e^{\lambda t}}$.**

Matrices no diagonalizables (Opcional)

Supongamos que λ es un autovalor de A. Podemos descubrir esto de dos maneras:

1. **Autovectores** (geométrico) Hay soluciones no nulas de $A\boldsymbol{x} = \lambda\boldsymbol{x}$.
2. **Autovalores** (algebraico) El determinante de $A - \lambda I$ es cero.

El número λ puede ser un autovalor simple o un autovalor múltiple, y queremos conocer su ***multiplicidad***. La mayoría de los autovalores tienen multiplicidad $M = 1$ (autovalores simples). En este caso hay una recta de autovectores, y $\det(A - \lambda I)$ no tiene un factor doble.

Para matrices excepcionales, un autovalor puede estar ***repetido***. En ese caso tenemos dos maneras diferentes de contar su multiplicidad. Siempre GM $\leq$ AM para cada λ:

1. **(Multiplicidad geométrica = GM)** Contar los **autovectores independientes** correspondientes a λ. Observar la dimensión del espacio nulo de $A - \lambda I$.
2. **(Multiplicidad algebraica = AM)** Contar las **repeticiones de $\boldsymbol{\lambda}$** entre los autovalores. Observar las raíces de $\det(A - \lambda I) = 0$.

Si A tiene $\lambda = 4, 4, 4$, entonces ese autovalor tiene AM = 3 y GM = 1 o 2 o 3.

La siguiente matriz A es el ejemplo típico de situación problemática. Su autovalor $\lambda = 0$ se repite. Es un autovalor doble (AM = 2) con un solo autovector (GM = 1).

$$\begin{matrix}\textbf{AM} = \textbf{2} \\ \textbf{GM} = \textbf{1}\end{matrix} \quad A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \text{ tiene } \det(A - \lambda I) = \begin{vmatrix} -\lambda & 1 \\ 0 & -\lambda \end{vmatrix} = \lambda^2. \qquad \begin{matrix}\boldsymbol{\lambda = 0, 0} \textbf{ pero} \\ \textbf{1 autovector}\end{matrix}$$

"Debería" haber dos autovectores, porque $\lambda^2 = 0$ tiene una raíz doble. El factor doble λ^2 indica que AM = 2. Pero hay solo un autovector $\boldsymbol{x} = (1, 0)$. Así que GM = 1. ***La escasez de autovectores cuando* GM < AM *significa que A no es diagonalizable***. No hay una matriz invertible de autovectores. La fórmula $A = X\Lambda X^{-1}$ falla.

Estas tres matrices tienen la misma escasez de autovectores. Su autovalor repetido es $\lambda = 5$. Todas las trazas son 10 y los determinantes son 25:

$$A = \begin{bmatrix} 5 & 1 \\ 0 & 5 \end{bmatrix} \quad \text{y} \quad A = \begin{bmatrix} 6 & -1 \\ 1 & 4 \end{bmatrix} \quad \text{y} \quad A = \begin{bmatrix} 7 & 2 \\ -2 & 3 \end{bmatrix}.$$

Todas tienen $\det(A - \lambda I) = (\lambda - 5)^2$. La multiplicidad algebraica es AM = 2. Pero cada $A - 5I$ es de rango $r = 1$. La multiplicidad geométrica es GM = 1. Solo hay una recta de autovectores para $\lambda = 5$, y estas matrices no son diagonalizables.

Problemas de la Sección I.6

1 La rotación $Q = \begin{bmatrix} \cos\theta & -\operatorname{sen}\theta \\ \operatorname{sen}\theta & \cos\theta \end{bmatrix}$ tiene autovalores complejos $\lambda = \cos\theta \pm i \operatorname{sen}\theta$:

$$Q\begin{bmatrix} 1 \\ -i \end{bmatrix} = (\cos\theta + i\operatorname{sen}\theta)\begin{bmatrix} 1 \\ -i \end{bmatrix} \quad \text{y} \quad Q\begin{bmatrix} 1 \\ i \end{bmatrix} = (\cos\theta - i\operatorname{sen}\theta)\begin{bmatrix} 1 \\ i \end{bmatrix}.$$

Verifique que $\lambda_1 + \lambda_2$ es igual a la traza de Q (que es la suma $q_{11} + q_{22}$ de los elementos de la diagonal). Verifique que $(\lambda_1)(\lambda_2)$ es igual al determinante. Verifique que esos autovectores complejos son ortogonales, usando el producto punto complejo $\overline{\boldsymbol{x}}_1 \cdot \boldsymbol{x}_2$ (¡ y no $\boldsymbol{x}_1 \cdot \boldsymbol{x}_2$!).

¿Quién es Q^{-1} y cuáles son sus autovalores ?

2 Calcule los autovalores y autovectores de A y A^{-1}. ¡ Verifique la traza !

$$A = \begin{bmatrix} 0 & 2 \\ 1 & 1 \end{bmatrix} \quad \text{y} \quad A^{-1} = \begin{bmatrix} -1/2 & 1 \\ 1/2 & 0 \end{bmatrix}.$$

A^{-1} tiene los _____ autovectores que A. Cuando A tiene autovalores λ_1 y λ_2, su inversa tiene autovalores _____.

3 Encuentre los autovalores de A y B (fácil en matrices triangulares) y $A + B$:

$$A = \begin{bmatrix} 3 & 0 \\ 1 & 1 \end{bmatrix} \quad \text{y} \quad B = \begin{bmatrix} 1 & 1 \\ 0 & 3 \end{bmatrix} \quad \text{y} \quad A + B = \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}.$$

Los autovalores de $A + B$ (*son iguales a*)(*no son iguales a*) los autovalores de A más los autovalores de B.

4 Encuentre los autovalores de A, B, AB y BA:

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad \text{y} \quad B = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \quad \text{y} \quad AB = \begin{bmatrix} 1 & 2 \\ 1 & 3 \end{bmatrix} \quad \text{y} \quad BA = \begin{bmatrix} 3 & 2 \\ 1 & 1 \end{bmatrix}.$$

(a) ¿Son los autovalores de AB iguales a los autovalores de A por los autovalores de B?

(b) ¿Son los autovalores de AB iguales a los autovalores de BA?

5 (a) Si sabe que $\boldsymbol{x}$ es un autovector, la manera de encontrar λ es _____.

(b) Si sabe que λ es un autovalor, la manera de encontrar $\boldsymbol{x}$ es _____.

6 Encuentre los autovalores y autovectores para las dos matrices de Markov A y A^∞. Explique a partir de las respuestas por qué A^{100} es cercana a A^∞:

$$A = \begin{bmatrix} .6 & .2 \\ .4 & .8 \end{bmatrix} \quad \text{y} \quad A^\infty = \begin{bmatrix} 1/3 & 1/3 \\ 2/3 & 2/3 \end{bmatrix}.$$

7 **El determinante de A es igual al producto $\lambda_1\lambda_2\cdots\lambda_n$.** Comience escribiendo el polinomio $\det(A-\lambda I)$ separado en sus n factores (siempre posible). Luego tome $\lambda = 0$:

$$\det(A-\lambda I) = (\lambda_1-\lambda)(\lambda_2-\lambda)\cdots(\lambda_n-\lambda) \quad \text{entonces} \quad \det A = _____.$$

Verifique esta regla en el Ejemplo 4 donde la matriz de Markov tiene $\lambda = 1$ y $\frac{1}{2}$.

8 La suma de los elementos de la diagonal (la *traza*) es igual a la suma de sus autovalores:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \text{tiene} \quad \det(A-\lambda I) = \lambda^2 - (a+d)\lambda + ad - bc = 0.$$

La fórmula cuadrática da los autovalores $\lambda = (a+d+\sqrt{\quad})/2$ y $\lambda =$ _____. Su suma es _____. Si A tiene $\lambda_1 = 3$ y $\lambda_2 = 4$ entonces $\det(A-\lambda I) =$ _____.

9 Si A tiene $\lambda_1 = 4$ y $\lambda_2 = 5$ entonces $\det(A-\lambda I) = (\lambda-4)(\lambda-5) = \lambda^2 - 9\lambda + 20$. Encuentre tres matrices que tengan traza $a+d=9$, determinante 20 y $\lambda = 4, 5$.

10 Elija las últimas filas de A y C para que tengan autovalores $4, 7$ y $1, 2, 3$:

Matrices compañeras

$$A = \begin{bmatrix} 0 & 1 \\ * & * \end{bmatrix} \quad C = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ * & * & * \end{bmatrix}.$$

11 ***Los autovalores de A son iguales a los autovalores de A^{T}.*** Esto se debe a que $\det(A-\lambda I)$ es igual a $\det(A^{\mathrm{T}}-\lambda I)$. Esto se cumple porque _____. Muestre con un ejemplo que los autovectores de A y A^{T} *no* son los mismos.

12 Esta matriz es singular de rango uno. Encuentre tres λ y tres autovectores:

$$A = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 2 \\ 4 & 2 & 4 \\ 2 & 1 & 2 \end{bmatrix}.$$

13 Suponga que A y B tienen los mismos autovalores $\lambda_1, \ldots, \lambda_n$ con los mismos autovectores independientes $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. Entonces $A = B$. *Razone*: cualquier vector $\boldsymbol{x}$ es una combinación $c_1\boldsymbol{x}_1 + \cdots + c_n\boldsymbol{x}_n$. ¿Qué es $A\boldsymbol{x}$? ¿Qué es $B\boldsymbol{x}$?

14 Suponga que A tiene autovalores $0, 3, 5$ con autovectores independientes $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}$.

(a) Dé una base para el espacio nulo y una base para el espacio columna.

(b) Encuentre una solución particular para $A\boldsymbol{x} = \boldsymbol{v} + \boldsymbol{w}$. Encuentre todas las soluciones.

(c) $A\boldsymbol{x} = \boldsymbol{u}$ no tiene solución. Si la tuviera entonces _____ estaría en el espacio columna.

15 (a) Factorice estas dos matrices en $A = X\Lambda X^{-1}$:

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix} \quad \text{y} \quad A = \begin{bmatrix} 1 & 1 \\ 3 & 3 \end{bmatrix}.$$

(b) Si $A = X\Lambda X^{-1}$ entonces $A^3 = (\quad)(\quad)(\quad)$ y $A^{-1} = (\quad)(\quad)(\quad)$.

16 Suponga que $A = X\Lambda X^{-1}$. ¿Cuál es la matriz de autovalores para $A + 2I$? ¿Cuál es la matriz de autovectores? Verifique que $A + 2I = (\quad)(\quad)(\quad)^{-1}$.

17 Verdadero o falso: Si las columnas de X (autovectores de A) son linealmente independientes, entonces

(a) A es invertible (b) A es diagonalizable

(c) X es invertible (d) X es diagonalizable.

18 Escriba la matriz más general que tiene autovectores $\begin{bmatrix}1\\1\end{bmatrix}$ y $\begin{bmatrix}1\\-1\end{bmatrix}$.

19 Verdadero o falso: si los autovalores de A son $2, 2, 5$ entonces la matriz es sin dudas

(a) invertible (b) diagonalizable (c) no diagonalizable.

20 Verdadero o falso: si los únicos autovectores de A son múltiplos de $(1, 4)$ entonces A

(a) no tiene inversa (b) tiene un autovalor repetido (c) no existe diagonalización $X\Lambda X^{-1}$.

21 $A^k = X\Lambda^k X^{-1}$ se acerca a la matriz cero cuando $k \to \infty$ si y solo si todo λ tiene valor absoluto menor a _____. ¿Cuál de estas matrices cumple $A^k \to 0$?

$$A_1 = \begin{bmatrix}.6 & .9\\ .4 & .1\end{bmatrix} \qquad \text{y} \qquad A_2 = \begin{bmatrix}.6 & .9\\ .1 & .6\end{bmatrix}.$$

22 Diagonalice A y calcule $X\Lambda^k X^{-1}$ para demostrar esta fórmula para A^k:

$$A = \begin{bmatrix}2 & -1\\ -1 & 2\end{bmatrix} \qquad \text{tiene} \qquad A^k = \frac{1}{2}\begin{bmatrix}1+3^k & 1-3^k\\ 1-3^k & 1+3^k\end{bmatrix}.$$

23 Los autovalores de A son 1 y 9, y los autovalores de B son -1 y 9:

$$A = \begin{bmatrix}5 & 4\\ 4 & 5\end{bmatrix} \qquad \text{y} \qquad B = \begin{bmatrix}4 & 5\\ 5 & 4\end{bmatrix}.$$

Encuentre una matriz raíz cuadrada de A a partir de $R = X\sqrt{\Lambda}\,X^{-1}$. ¿Por qué no hay una matriz real que sea raíz cuadrada de B?

24 Suponga que la misma X diagonaliza a ambas matrices A y B. Tienen *los mismos autovectores* en $A = X\Lambda_1 X^{-1}$ y $B = X\Lambda_2 X^{-1}$. Demuestre que $AB = BA$.

25 La traspuesta de $A = X\Lambda X^{-1}$ es $A^{\mathrm{T}} = (X^{-1})^{\mathrm{T}}\Lambda X^{\mathrm{T}}$. Los autovectores en $A^{\mathrm{T}}\boldsymbol{y} = \lambda\boldsymbol{y}$ son las columnas de la matriz $(X^{-1})^{\mathrm{T}}$. Se los conoce como ***autovectores a izquierda de A***. ¿Qué matrices multiplicaría y cómo para encontrar la siguiente fórmula para A?

Suma de matrices de rango 1 $\quad A = X\Lambda X^{-1} = \lambda_1 \boldsymbol{x}_1\boldsymbol{y}_1^{\mathrm{T}} + \cdots + \lambda_n \boldsymbol{x}_n\boldsymbol{y}_n^{\mathrm{T}}.$

26 **¿Cuándo es una matriz A semejante a su matriz de autovalores Λ?**

A y Λ siempre tienen los mismos autovalores. Pero para que sean semejantes se requiere de una matriz B con $A = B\Lambda B^{-1}$. Entonces B es la matriz _____ y A debe tener n _____ independientes.

I.7. Matrices simétricas y definidas positivas

Las matrices simétricas $S = S^{\mathrm{T}}$ se merecen toda la atención que reciben. Mirando sus autovalores y autovectores, podrá ver por qué son tan especiales :

1 **Los n autovalores λ de una matriz simétrica S son números reales.**

2 **Los n autovectores q pueden ser elegidos ortogonales (perpendiculares entre sí).**

La matriz identidad $S = I$ es un caso extremo. Todos sus autovalores son $\lambda = 1$. Todo vector no nulo $\boldsymbol{x}$ es un autovector : $I\boldsymbol{x} = 1\boldsymbol{x}$. Esto explica por qué escribimos "pueden ser elegidos" en el punto 2 arriba. Con autovalores repetidos como $\lambda_1 = \lambda_2 = 1$, tenemos muchas posibilidades de elecciones para los autovectores. Podemos elegir que sean ortogonales y podemos reescalarlos para que sean vectores unitarios (norma 1). Entonces esos autovectores $\boldsymbol{q}_1, \ldots \boldsymbol{q}_n$ no son solo ortogonales sino que son **ortonormales. La matriz de autovectores para S satisface $\boldsymbol{Q^{\mathrm{T}}Q = I}$** : columnas ortonormales en Q.

$$\boldsymbol{q_i}^{\mathrm{T}}\boldsymbol{q_j} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \quad \text{nos lleva a} \quad \begin{bmatrix} \boldsymbol{q_1^{\mathrm{T}}} \\ \vdots \\ \boldsymbol{q_n^{\mathrm{T}}} \end{bmatrix} \begin{bmatrix} \boldsymbol{q_1} & \cdots & \boldsymbol{q_n} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & 0 & \cdot & 0 \\ 0 & \mathbf{1} & 0 & \cdot \\ \cdot & 0 & \mathbf{1} & 0 \\ 0 & \cdot & 0 & \mathbf{1} \end{bmatrix}.$$

Escribimos Q en lugar de X para la matriz de autovectores de S, para enfatizar que estos autovectores son ortonormales : $Q^{\mathrm{T}}Q = I$ y $Q^{\mathrm{T}} = Q^{-1}$. Esta matriz de autovectores es una **matriz ortogonal**. La $A = X\Lambda X^{-1}$ que conocemos se convierte en $S = Q\Lambda Q^{\mathrm{T}}$:

Teorema espectral	**Toda matriz simétrica real tiene la forma $\boldsymbol{S = Q\Lambda Q^{\mathrm{T}}}$.**

Toda matriz con esa forma es simétrica : Trasponga $Q\Lambda Q^{\mathrm{T}}$ para obtener $Q^{\mathrm{TT}}\Lambda^{\mathrm{T}}Q^{\mathrm{T}} = Q\Lambda Q^{\mathrm{T}}$.

Demostraciones rápidas : Autovectores ortogonales y autovalores reales

Suponga primero que $S\boldsymbol{x} = \lambda\boldsymbol{x}$ y $S\boldsymbol{y} = 0\boldsymbol{y}$. La matriz simétrica S tiene un autovalor distinto de cero λ y un autovalor cero. Entonces $\boldsymbol{y}$ está en el espacio nulo de S y $\boldsymbol{x}$ está en el espacio columna de S ($\boldsymbol{x} = S\boldsymbol{x}/\lambda$ es una combinación de las columnas de S). *Pero S es simétrica : ¡ espacio columna = espacio fila* ! Como el espacio fila y el espacio nulo son siempre ortogonales, hemos podido demostrar que **$\boldsymbol{x}$ es ortogonal a $\boldsymbol{y}$**.

Cuando el segundo autovalor es distinto de cero, tenemos $S\boldsymbol{y} = \alpha\boldsymbol{y}$. En este caso miramos la matriz $S - \alpha I$. Entonces $(S - \alpha I)\boldsymbol{y} = 0\boldsymbol{y}$ y $(S - \alpha I)\boldsymbol{x} = (\lambda - \alpha)\boldsymbol{x}$ con $\lambda - \alpha \neq 0$. Ahora $\boldsymbol{y}$ está en el espacio nulo y $\boldsymbol{x}$ está en el espacio columna (¡ = espacio fila !) de $S - \alpha I$. Por lo tanto $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x} = 0$: *Los autovectores son ortogonales cuando los autovalores $\lambda \neq \alpha$ son distintos.*

En esos párrafos se supuso que los autovalores y los autovectores eran reales. Para demostrar esto, multiplicamos $S\boldsymbol{x} = \lambda\boldsymbol{x}$ por el vector conjugado complejo $\overline{\boldsymbol{x}}^{\mathrm{T}}$ (cada i cambia a $-i$). Entonces $\overline{\boldsymbol{x}}^{\mathrm{T}}S\boldsymbol{x} = \lambda\overline{\boldsymbol{x}}^{\mathrm{T}}\boldsymbol{x}$. Cuando mostremos que $\overline{\boldsymbol{x}}^{\mathrm{T}}\boldsymbol{x}$ y $\overline{\boldsymbol{x}}^{\mathrm{T}}S\boldsymbol{x}$ son reales, sabremos que **$\boldsymbol{\lambda}$ es real.**

$$\overline{\boldsymbol{x}}^{\mathrm{T}}\boldsymbol{x} = \overline{x}_1 x_1 + \cdots + \overline{x}_n x_n \quad \text{y cada} \quad \overline{x}_k\, x_k \quad \text{es} \quad (a - ib)(a + ib) = a^2 + b^2 \ \textbf{(real)}$$

También es real : $\overline{\boldsymbol{x}}^{\mathrm{T}} S\boldsymbol{x} = S_{11}\overline{x}_1 x_1 + S_{12}(\overline{x}_1 x_2 + x_1 \overline{x}_2) + \cdots$ y de nuevo $\overline{x}_1 x_1$ es real

$$\overline{x}_1 x_2 + x_1 \overline{x}_2 = (a - ib)(c + id) + (a + ib)(c - id) = 2ac + 2bd = \textbf{ real}$$

Luego **el cociente** $\boldsymbol{\lambda = \overline{x}^{\mathrm{T}} S x / \overline{x}^{\mathrm{T}} x}$ **es real.** Así $(S - \lambda I)\boldsymbol{x} = \boldsymbol{0}$ nos da un autovector real.

Comentario sobre complejos : trasponer $S\boldsymbol{x} = \lambda \boldsymbol{x}$ y tomar conjugados complejos nos da $\overline{\boldsymbol{x}}^{\mathrm{T}}\, \overline{S}^{\mathrm{T}} = \overline{\lambda}\, \overline{\boldsymbol{x}}^{\mathrm{T}}$. Para nuestras matrices simétricas reales, $\overline{S}^{\mathrm{T}}$ es exactamente S. Dependemos de este paso doble, *trasponer y conjugar*, para obtener de vuelta S. Como la prueba solo necesita $\overline{S}^{\mathrm{T}} = S$, también permite matrices complejas : **Cuando $\boldsymbol{\overline{S}^{\mathrm{T}} = S}$ todos los autovalores de $\boldsymbol{S}$ son reales.**

Ejemplo complejo $\boldsymbol{S = \begin{bmatrix} 2 & \mathbf{3 - 3i} \\ \mathbf{3 + 3i} & 5 \end{bmatrix} = \overline{S}^{\mathbf{T}}}$ tiene autovalores reales 8 y -1.

La clave es $\overline{3 + 3i} = 3 - 3i$. El determinante es $(2)(5) - (3 + 3i)(3 - 3i) = 10 - 18 = -8$. Los autovectores de esta matriz son $\boldsymbol{x}_1 = (1, 1 + i)$ y $\boldsymbol{x}_2 = (1 - i, -1)$. Esos vectores son ortogonales cuando consideramos el producto interno complejo $\overline{\boldsymbol{x}}_1^{\mathrm{T}} \boldsymbol{x}_2$. Este es el producto interno correcto para vectores complejos, y produce $\overline{\boldsymbol{x}}_1^{\mathrm{T}} \boldsymbol{x}_2 = 0$:

$$\text{Cambiamos } \boldsymbol{x}_1^{\mathrm{T}} \boldsymbol{x}_2 = \begin{bmatrix} 1 & 1 + i \end{bmatrix} \begin{bmatrix} 1 - i \\ -1 \end{bmatrix} = \boldsymbol{-2i} \quad \text{a} \quad \overline{\boldsymbol{x}}_1^{\mathrm{T}} \boldsymbol{x}_2 = \begin{bmatrix} \mathbf{1} & \mathbf{1 - i} \end{bmatrix} \begin{bmatrix} \mathbf{1 - i} \\ \mathbf{-1} \end{bmatrix} = \mathbf{0}.$$

Sistemas como MATLAB y **Julia** entienden : el vector $\boldsymbol{x}'$ y la matriz A' son automáticamente conjugados cuando se trasponen. *Todo i cambia a $-i$.* Entonces $\boldsymbol{x}'$ es $\overline{\boldsymbol{x}}^{\mathrm{T}}$ y A' es $\overline{A}^{\mathrm{T}}$. Otro símbolo conocido para $\overline{\boldsymbol{x}}^{\mathrm{T}}$ y $\overline{A}^{\mathrm{T}}$ es un asterisco : $\boldsymbol{x}^*$ y A^*.

Matrices definidas positivas

Estamos trabajando con matrices simétricas reales $S = S^{\mathrm{T}}$. Todos sus autovalores son reales. Algunas de esas matrices simétricas (*no todas*) tienen una propiedad potente adicional que las ubica en una posición central en la matemática aplicada. He aquí dicha propiedad :

Condición 1 | **Una matriz definida positiva tiene todos sus autovalores positivos.**

Nos gustaría verificar que los autovalores son positivos sin tener que calcular los λ. Luego de algunos ejemplos, veremos cuatro condiciones más para determinar cuándo una matriz es definida positiva.

1 $S = \begin{bmatrix} 2 & 0 \\ 0 & 6 \end{bmatrix}$ es definida positiva. Sus autovalores 2 y 6 son ambos positivos

2 $S = Q \begin{bmatrix} 2 & 0 \\ 0 & 6 \end{bmatrix} Q^{\mathrm{T}}$ es definida positiva si $Q^{\mathrm{T}} = Q^{-1}$: mismos $\lambda = 2$ y 6

3 $S = C \begin{bmatrix} 2 & 0 \\ 0 & 6 \end{bmatrix} C^{\mathrm{T}}$ es definida positiva si C es invertible (esto no es obvio)

4 $S = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$ es definida positiva exactamente cuando $a > 0$ y $ac > b^2$

5 $S = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$ es solo **semidefinida positiva** : tiene $\boldsymbol{\lambda \geq 0}$ pero no $\lambda > 0$

Definición basada en energía

¿ Puedo hablarles de la idea más importante relacionada con las matrices definidas positivas ? Este nuevo enfoque no requiere de autovalores pero resulta ser un test perfecto de $\lambda > 0$. Esta es una buena definición de matrices definidas positivas. **Condición 2 : la condición de la energía**.

$$\boxed{\boldsymbol{S} \textbf{ es definida positiva si la energía } \boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x} \textbf{ es positiva para todos los vectores } \boldsymbol{x} \neq \boldsymbol{0}} \quad (1)$$

Por supuesto que $S = I$ es definida positiva : todo $\lambda_i = 1$. La energía es $\boldsymbol{x}^{\mathrm{T}} I \boldsymbol{x} = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$, positiva si $\boldsymbol{x} \neq 0$. Les mostraré la energía en una matriz de 2 por 2. Depende de $\boldsymbol{x} = (x_1, x_2)$.

$$\textbf{Energía} \quad \boxed{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 2 & 4 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \boldsymbol{2\,x_1^2 + 8\,x_1 x_2 + 9\,x_2^2}}$$

¿ Es positiva para toda x_1 y x_2 excepto $(x_1, x_2) = (0, 0)$? *Sí, es una suma de cuadrados* :

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x} = 2x_1^2 + 8x_1x_2 + 9x_2^2 = 2\,(x_1 + 2x_2)^2 + x_2^2 = \text{ energía positiva.}$$

Debemos conectar la energía positiva $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x} > 0$ a los autovalores positivos $\lambda > 0$:

$$\textbf{Si } \boldsymbol{S}\boldsymbol{x} = \boldsymbol{\lambda}\boldsymbol{x} \textbf{ entonces } \boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x} = \boldsymbol{\lambda}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}. \textbf{ Así } \boldsymbol{\lambda > 0} \textbf{ nos lleva a } \boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x} > \boldsymbol{0}.$$

Esta línea solo verifica la energía en cada autovector $\boldsymbol{x}$ por separado. Pero la teoría dice que si todo autovector tiene energía positiva, **entonces todos los vectores $\boldsymbol{x}$ distintos de cero tienen energía positiva** :

$$\textbf{Si } \boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x} > 0 \textbf{ para los autovectores de } \boldsymbol{S}, \textbf{ entonces } \boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x} > 0 \textbf{ para todo vector no nulo } \boldsymbol{x}.$$

He aquí la razón. Todo $\boldsymbol{x}$ es una combinación $c_1\boldsymbol{x}_1 + \cdots + c_n\boldsymbol{x}_n$ de los autovectores. Esos autovectores pueden elegirse ortogonales porque S es simétrica. Ahora mostraremos que : $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x}$ es una combinación positiva de las energías $\lambda_k \boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{x}_k > 0$ de cada autovector.

$$\begin{aligned} \boldsymbol{x}^{\mathrm{T}} S \boldsymbol{x} &= \left(c_1 \boldsymbol{x}_1^{\mathrm{T}} + \cdots + c_n \boldsymbol{x}_n^{\mathrm{T}}\right) S \left(c_1 \boldsymbol{x}_1 + \cdots + c_n \boldsymbol{x}_n\right) \\ &= \left(c_1 \boldsymbol{x}_1^{\mathrm{T}} + \cdots + c_n \boldsymbol{x}_n^{\mathrm{T}}\right) \left(c_1 \lambda_1 \boldsymbol{x}_1 + \cdots + c_n \lambda_n \boldsymbol{x}_n\right) \\ &= c_1^2 \lambda_1 \boldsymbol{x}_1^{\mathrm{T}} \boldsymbol{x}_1 + \cdots + c_n^2 \lambda_n \boldsymbol{x}_n^{\mathrm{T}} \boldsymbol{x}_n > \mathbf{0} \text{ si todo } \boldsymbol{\lambda_i} > \mathbf{0}. \end{aligned}$$

De la segunda a la tercera línea usamos la ortogonalidad de los autovectores de S: $\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j = 0$. He aquí un uso típico del test de energía sin conocer ningún autovalor o autovector.

Si $\boldsymbol{S_1}$ y $\boldsymbol{S_2}$ son simétricas definidas positivas, entonces también lo es $\boldsymbol{S_1 + S_2}$

Prueba sumando energías : $\boldsymbol{x}^{\mathrm{T}}(S_1 + S_2)\,\boldsymbol{x} = \boldsymbol{x}^{\mathrm{T}} S_1\, \boldsymbol{x} + \boldsymbol{x}^{\mathrm{T}} S_2\, \boldsymbol{x} > \mathbf{0} + \mathbf{0}$

Los autovalores y autovectores de $S_1 + S_2$ no son fáciles de encontrar. Las energías solo se suman.

Otras tres condiciones equivalentes

Hasta el momento tenemos las condiciones **1** y **2** : autovalores positivos y energía positiva. Ese test de energía rápidamente nos da otros tres tests útiles (y probablemente otros, pero nos detenemos en estos tres) :

Condición 3 $\boldsymbol{S = A^{\mathrm{T}} A}$ **para una matriz $\boldsymbol{A}$ con columnas independientes**

Condición 4 **Todos los determinantes principales $\boldsymbol{D_1, D_2, \ldots, D_n}$ de $\boldsymbol{S}$ son positivos**

Condición 5 **Todos los pivotes de $\boldsymbol{S}$ son positivos** (en eliminación)

La condición 3 se aplica a $S = A^{\mathrm{T}} A$. ¿ Por qué deben ser independientes las columnas de A en este test ? Observe estos paréntesis :

$$\boldsymbol{S = A^{\mathrm{T}} A} \qquad \textbf{Energía } = \boldsymbol{x}^{\mathrm{T}} S \boldsymbol{x} = \boldsymbol{x}^{\mathrm{T}} A^{\mathrm{T}} A \boldsymbol{x} = (\boldsymbol{Ax})^{\mathrm{T}} (\boldsymbol{Ax}) = ||A\boldsymbol{x}||^2. \tag{2}$$

Esos paréntesis son la clave. La energía es la **norma al cuadrado** del vector $A\boldsymbol{x}$. Esta energía es positiva siempre que $A\boldsymbol{x}$ no sea el vector nulo. Para garantizar que $A\boldsymbol{x} \neq \mathbf{0}$ cuando $\boldsymbol{x} \neq \mathbf{0}$, las columnas de A deben ser independientes. En este ejemplo de 2 por 3, A tiene *columnas dependientes* :

$$S = A^{\mathrm{T}} A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 4 \\ 3 & 5 & 7 \\ 4 & 7 & 10 \end{bmatrix} \quad \textbf{no} \text{ es definida positiva.}$$

Esta A tiene columna 1 + columna 3 = 2 (columna 2). Entonces $\boldsymbol{x} = (1, -2, 1)$ tiene energía cero. Es un autovector de $A^{\mathrm{T}} A$ con $\lambda = 0$. Entonces $S = A^{\mathrm{T}} A$ es solo semidefinida positiva.

La ecuación (2) establece que $A^{\mathrm{T}} A$ es al menos *semidefinida*, porque $\boldsymbol{x}^{\mathrm{T}} S \boldsymbol{x} = ||A\boldsymbol{x}||^2$ nunca es negativo. ***Las matrices semidefinidas permiten que la energía / los autovalores / los determinantes / los pivotes de $\boldsymbol{S}$ sean cero.***

Condición de determinantes y condición de pivote

La condición de determinantes es la forma más veloz para una matriz pequeña. Marcaré los cuatro "determinantes principales" D_1, D_2, D_3, D_4 en esta matriz de diferencias segundas de 4 por 4.

$$S = \begin{bmatrix} \boxed{\begin{matrix} 2 & -1 \\ -1 & 2 \end{matrix}} & & \\ & \boxed{\begin{matrix} 2 & -1 \\ -1 & 2 \end{matrix}} & \\ & & \boxed{\begin{matrix} 2 & -1 \\ -1 & 2 \end{matrix}} \end{bmatrix} \quad \text{tiene} \quad \begin{matrix} \text{1er determinante } D_1 = \mathbf{2} \\ \text{2do determinante } D_2 = \mathbf{3} \\ \text{3er determinante } D_3 = \mathbf{4} \\ \text{4to determinante } D_4 = \mathbf{5} \end{matrix}$$

¡ Aquí se cumple la condición de los determinantes ! La energía $\boldsymbol{x}^{\mathrm{T}} S \boldsymbol{x}$ también será positiva.

Los determinantes principales están vinculados a los pivotes (los números en las diagonales luego de la eliminación). Aquí el primer pivote es $\mathbf{2}$. El segundo pivote $\frac{\mathbf{3}}{\mathbf{2}}$ aparece cuando $\frac{1}{2}$(fila 1) se suma a la fila 2. El tercer pivote $\frac{\mathbf{4}}{\mathbf{3}}$ aparece cuando $\frac{2}{3}$(nueva fila 2) se suma a la fila 3. Esas fracciones $\frac{2}{1}, \frac{3}{2}, \frac{4}{3}$ son los cocientes de los determinantes. El último pivote es $\frac{\mathbf{5}}{\mathbf{4}}$.

El k-ésimo pivote es el cociente $\dfrac{\boldsymbol{D_k}}{\boldsymbol{D_{k-1}}}$ de los determinantes principales (tamaño k y $k-1$)

Así, los pivotes son todos positivos cuando todos los determinantes principales son positivos.

Puedo conectar rápidamente estas dos condiciones ($\mathbf{4}$ y $\mathbf{5}$) con la tercera condición $S = A^{\mathrm{T}}A$. De hecho, eliminación en S produce una importante elección particular de A. Recuerde que *eliminación = factorización triangular* ($\boldsymbol{S = LU}$). Hasta ahora L ha tenido unos en la diagonal y U contenía los pivotes. Pero con las matrices simétricas podemos balancear S como $\boldsymbol{LDL^{\mathrm{T}}}$:

$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & & \\ -\frac{1}{2} & 1 & \\ 0 & -\frac{2}{3} & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 & 0 \\ & \frac{3}{2} & -1 \\ & & \frac{4}{3} \end{bmatrix} \qquad \boldsymbol{S = LU} \qquad (3)$$

$$\begin{matrix} \text{quitar} \\ \text{los pivotes} \\ \text{en } D \end{matrix} = \begin{bmatrix} 1 & & \\ -\frac{1}{2} & 1 & \\ 0 & -\frac{2}{3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{2} & & \\ & \frac{\mathbf{3}}{\mathbf{2}} & \\ & & \frac{\mathbf{4}}{\mathbf{3}} \end{bmatrix} \begin{bmatrix} 1 & -\frac{1}{2} & 0 \\ & 1 & -\frac{2}{3} \\ & & 1 \end{bmatrix} = \boldsymbol{LDL^{\mathrm{T}}} \qquad (4)$$

$$\begin{matrix} \textbf{compartir esos} \\ \textbf{pivotes} \\ \textbf{entre } \boldsymbol{A^{\mathrm{T}}} \textbf{ y } \boldsymbol{A} \end{matrix} = \begin{bmatrix} \sqrt{2} & & \\ -\sqrt{\frac{1}{2}} & \sqrt{\frac{3}{2}} & \\ 0 & -\sqrt{\frac{2}{3}} & \sqrt{\frac{4}{3}} \end{bmatrix} \begin{bmatrix} \sqrt{2} & -\sqrt{\frac{1}{2}} & 0 \\ & \sqrt{\frac{3}{2}} & -\sqrt{\frac{2}{3}} \\ & & \sqrt{\frac{4}{3}} \end{bmatrix} = \boldsymbol{A^{\mathrm{T}}A} \qquad (5)$$

Perdón por esas raíces cuadradas, pero el patrón $S = A^{\mathrm{T}}A$ es hermoso : $\boldsymbol{A = \sqrt{D}L^{\mathrm{T}}}$.

La eliminación factoriza toda S definida positiva en $\boldsymbol{A^{\mathrm{T}}A}$ ($\boldsymbol{A}$ es triangular superior)

Esta es la factorización de Cholesky $S = A^{\mathrm{T}}A$ con $\sqrt{\text{pivotes}}$ en la diagonal principal de A.

La condición $S = A^{\mathrm{T}}A$: dos elecciones especiales de A

Para aplicar el test $S = A^{\mathrm{T}}A$ cuando S es definida positiva, debemos encontrar al menos una A posible. Hay muchas elecciones de A posibles, entre ellas (1) **simétrica** y (2) **triangular**.

1 Si $S = Q\Lambda Q^{\mathrm{T}}$, tomamos las raíces cuadradas de esos autovalores. Luego $A = Q\sqrt{\Lambda}Q^{\mathrm{T}} = A^{\mathrm{T}}$.

2 Si $S = LU = LDL^{\mathrm{T}}$ con pivotes positivos en D, entonces $S = (L\sqrt{D})\,(\sqrt{D}L^{\mathrm{T}})$.

En resumen Las cinco condiciones para verificar que S es definida positiva requieren de diferentes aspectos del álgebra lineal: pivotes de eliminación, determinantes, autovalores y $S = A^{\mathrm{T}}A$. Cada condición da una respuesta completa en sí misma: definida o semidefinida positiva o ninguna.

Energía positiva $\boldsymbol{x^{\mathrm{T}}Sx > 0}$ es la mejor definición: conecta todas las condiciones.

Matrices definidas positivas y problemas de mínimos

Sea S una matriz de 2 por 2 simétrica y definida positiva. Apliquemos cuatro de los tests:

$$S = \begin{bmatrix} a & b \\ b & c \end{bmatrix} \quad \begin{array}{ll} \textbf{determinantes} & a > 0, ac - b^2 > 0 \\ \textbf{autovalores} & \lambda_1 > 0,\ \lambda_2 > 0 \end{array} \quad \begin{array}{ll} \textbf{pivotes} & a > 0, (ac - b^2)/a > 0 \\ \textbf{energía} & ax^2 + 2bxy + cy^2 > 0 \end{array}$$

Usaré un ejemplo con $a = c = 5$ y $b = 4$. Esta matriz S tiene $\lambda = \mathbf{9}$ y $\lambda = \mathbf{1}$.

$$\textbf{Energía } \boldsymbol{E = x^{\mathrm{T}}Sx} \qquad \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{5x^2 + 8xy + 5y^2} > 0$$

El gráfico de esta función de energía $E(x, y)$ es un **bol con abertura hacia arriba.** El punto inferior del bol tiene energía $E = 0$ cuando $x = y = 0$. Esto conecta los problemas de mínimos del cálculo con las matrices definidas positivas del álgebra lineal.

La Parte VI de este libro describe minimización numérica. En los problemas más sencillos, la función es **estrictamente convexa**, una parábola que se abre hacia arriba. He aquí una condición perfecta: *la matriz de las derivadas segundas es definida positiva en todos los puntos*. Nos encontramos en dimensiones altas, pero el álgebra lineal identifica las propiedades cruciales de matriz de derivadas segundas.

Para una función $f(x)$ común de una variable x, la condición para un mínimo es famosa:

$$\text{Mínimo si } \textbf{la primera} \text{ derivada } \frac{df}{dx} = 0 \text{ y } \textbf{la segunda} \text{ derivada } \frac{d^2f}{dx^2} > 0 \text{ en } x = x_0$$

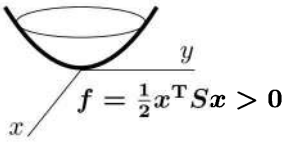
Para $f(x, y)$ con dos variables, las derivadas segundas van en una matriz: ¡ definida positiva !

Mínimo en $\boldsymbol{x_0, y_0}$	$\dfrac{\partial f}{\partial x} = 0$	y	$\dfrac{\partial f}{\partial y} = 0$	y	$\begin{bmatrix} \boldsymbol{\partial^2 f/\partial x^2} & \boldsymbol{\partial^2 f/\partial x \partial y} \\ \boldsymbol{\partial^2 f/\partial x \partial y} & \boldsymbol{\partial^2 f/\partial y^2} \end{bmatrix}$ **es definida positiva en $\boldsymbol{x_0, y_0}$**

El gráfico de $z = f(x, y)$ es plano en ese punto x_0, y_0 porque $\partial f/\partial x = \partial f/\partial y = 0$. El gráfico solo va hacia arriba cuando la matriz de derivadas segundas es definida positiva. Entonces tenemos un punto mínimo de la función $f(x, y)$.

Derivadas segundas $S = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$

$a > 0$ y $ac > b^2$



El gráfico de $2f = ax^2 + 2bxy + cy^2$ es un bol cuando S es definida positiva.

Si S tiene un autovalor negativo $\lambda < 0$, el gráfico va hacia abajo en alguna dirección. Hay un *máximo* si S es definida negativa (todo $\lambda < 0$, bol invertido). O un *punto de ensilladura* cuando S tiene autovalores tanto positivos como negativos. Una matriz correspondiente a un punto de ensilladura es "*indefinida*".

Optimización y aprendizaje automático (*machine learning*)

La Parte VI de este libro describirá el **descenso por gradiente**. Cada paso toma la dirección de mayor descenso hacia el punto $\boldsymbol{x^*}$ del fondo del bol. Pero esa dirección de mayor pendiente cambia a medida que descendemos. Aquí es donde el cálculo se encuentra con el álgebra lineal, en la búsqueda del punto mínimo $\boldsymbol{x^*}$.

Cálculo	Las derivadas parciales de f son todas cero en $\boldsymbol{x^*}$: $\dfrac{\partial f}{\partial x_i} = 0$
Álgebra lineal	La matriz S de las derivadas segundas $\dfrac{\partial^2 f}{\partial x_i \, \partial x_j}$ es definida positiva

Si S es definida (o semidefinida) positiva en todos los puntos $\boldsymbol{x} = (x_1, \ldots, x_n)$, entonces **la función $\boldsymbol{f(x)}$ es convexa**. Si los autovalores de S se mantienen mayores a un número positivo δ, entonces **la función $\boldsymbol{f(x)}$ es estrictamente convexa**. Estas son las mejores funciones para optimizar. Solo tienen un mínimo, y el descenso por gradiente lo encontrará.

El aprendizaje automático produce "funciones de pérdida" con cientos de miles de variables, que miden el error (que minimizamos). Pero calcular todas las derivadas segundas es imposible. Primero usamos las derivadas para determinar en qué dirección movernos: el error disminuye más rápido en la dirección más empinada. Luego damos otro paso descendente en una nueva dirección.

Este es el cálculo central en cuadrados mínimos, en redes neuronales y en aprendizaje profundo (*deep learning*).

La elipse $ax^2 + 2bxy + cy^2 = 1$

Tomemos una matriz S definida positiva. El gráfico de su energía $E = \boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}$ es un bol abierto hacia arriba. Cortémoslo a la altura $\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x} = 1$. La curva a lo largo del corte es una **elipse**.

$S = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}$ tiene $\lambda = \mathbf{9}$ y $\mathbf{1}$ Elipse de energía $\mathbf{5x^2 + 8xy + 5y^2 = 1}$ en la Figura I.9

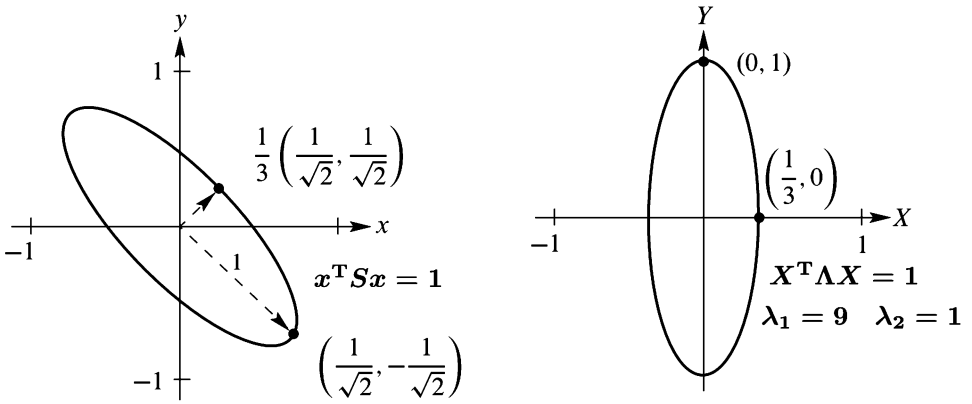


Figura I.9: la elipse inclinada $5x^2+8xy+5y^2=1$. Alineada es $9X^2+Y^2=1$.

Los autovectores son $\boldsymbol{q}_1=(1,1)$ y $\boldsymbol{q}_2=(1,-1)$. Los dividimos por $\sqrt{2}$ para obtener vectores unitarios. Luego $S=Q\Lambda Q^{\mathrm{T}}$. Ahora multiplicamos por $\boldsymbol{x}^{\mathrm{T}}=\begin{bmatrix}x & y\end{bmatrix}$ a la izquierda y por $\boldsymbol{x}$ a la derecha para obtener la energía $\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}=(\boldsymbol{x}^{\mathrm{T}}Q)\Lambda(Q^{\mathrm{T}}\boldsymbol{x})$. Los autovalores de S son 9 y 1.

$$\boxed{\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}=\textbf{ suma de cuadrados}\qquad 5x^2+8xy+5y^2=9\left(\frac{x+y}{\sqrt{2}}\right)^2+1\left(\frac{x-y}{\sqrt{2}}\right)^2.} \tag{6}$$

9 y 1 provienen de Λ. Dentro de los cuadrados podemos ver a $\boldsymbol{q}_1=(1,1)/\sqrt{2}$ y $\boldsymbol{q}_2=(1,-1)/\sqrt{2}$.

Los ejes de la elipse inclinada se alinean con los autovectores de $\boldsymbol{S}$. Esto explica por qué $S=Q\Lambda Q^{\mathrm{T}}$ es el "teorema del eje principal": muestra los ejes. No solo las direcciones (de los autovectores) sino también las longitudes de los ejes (por los λ) : **longitud $=1/\sqrt{\lambda}$**. Para ver todo, usemos letras mayúsculas para las nuevas coordenadas X,Y que alinean la elipse :

$$\textbf{Alineada}\qquad \frac{x+y}{\sqrt{2}}=X\qquad \text{y}\qquad \frac{x-y}{\sqrt{2}}=Y\qquad \text{y}\qquad \boldsymbol{9X^2+Y^2=1}.$$

El mayor valor de X^2 es $1/9$. El punto extremo del eje más corto tiene $X=1/3$ e $Y=0$. Observe : el autovalor *mayor* $\lambda_1=9$ corresponde al eje *más corto*, de la mitad de longitud $1/\sqrt{\lambda_1}=1/3$. El autovalor menor $\lambda_2=1$ corresponde a la mayor longitud $1/\sqrt{\lambda_2}=1$: eje Y en la Figura I.9.

En el sistema xy , los ejes tienen la dirección de los autovectores de S. En el sistema XY, los **ejes se encuentran en dirección a los autovectores de $\boldsymbol{\Lambda}$: los ejes coordenados**. Todo a partir de $S=Q\Lambda Q^{\mathrm{T}}$.

$$\boxed{\begin{array}{l}\boldsymbol{S=Q\Lambda Q^{\mathrm{T}}} \text{ es definida positiva cuando todo } \lambda_i>0. \text{ El gráfico de } \boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}=1 \text{ es una elipse, con sus ejes apuntando en la misma dirección que los autovectores de } S.\\[2ex] \textbf{Elipse}\qquad \begin{bmatrix}x & y\end{bmatrix}Q\Lambda Q^{\mathrm{T}}\begin{bmatrix}x\\y\end{bmatrix}=\begin{bmatrix}X & Y\end{bmatrix}\Lambda\begin{bmatrix}X\\Y\end{bmatrix}=\boldsymbol{\lambda_1X^2+\lambda_2Y^2=1}.\end{array}} \tag{7}$$

Problemas de la Sección I.7

1 Suponga que $S^{\mathrm{T}} = S$ y $S\boldsymbol{x} = \lambda\boldsymbol{x}$ y $S\boldsymbol{y} = \alpha\boldsymbol{y}$ son reales. Muestre que

$$\boldsymbol{y}^{\mathrm{T}}S\boldsymbol{x} = \lambda\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x} \quad \text{y} \quad \boldsymbol{x}^{\mathrm{T}}S\boldsymbol{y} = \alpha\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} \quad \text{y} \quad \boldsymbol{y}^{\mathrm{T}}S\boldsymbol{x} = \boldsymbol{x}^{\mathrm{T}}S\boldsymbol{y}.$$

Muestre que $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}$ debe ser cero si $\lambda \neq \alpha$: **autovectores ortogonales.**

2 ¿Cuál o cuáles de las S_1, S_2, S_3, S_4 tiene dos autovalores positivos? No calcule los λ, use alguna condición para ver que S_1 no es definida positiva. Encuentre también un $\boldsymbol{x}$ para el que $\boldsymbol{x}^{\mathrm{T}}S_1\boldsymbol{x} < 0$.

$$S_1 = \begin{bmatrix} 5 & 6 \\ 6 & 7 \end{bmatrix} \quad S_2 = \begin{bmatrix} -1 & -2 \\ -2 & -5 \end{bmatrix} \quad S_3 = \begin{bmatrix} 1 & 10 \\ 10 & 100 \end{bmatrix} \quad S_4 = \begin{bmatrix} 1 & 10 \\ 10 & 101 \end{bmatrix}.$$

3 ¿Para qué números b y c son estas matrices definidas positivas?

$$S = \begin{bmatrix} 1 & b \\ b & 9 \end{bmatrix} \qquad S = \begin{bmatrix} 2 & 4 \\ 4 & c \end{bmatrix} \qquad S = \begin{bmatrix} c & b \\ b & c \end{bmatrix}.$$

Con los pivotes en D y los multiplicadores en L, factorice cada A como LDL^{T}.

4 He aquí una "prueba" rápida de que los autovalores de toda matriz real A son reales:

$$\textbf{Prueba falsa} \quad A\boldsymbol{x} = \lambda\boldsymbol{x} \quad \text{da} \quad \boldsymbol{x}^{\mathrm{T}}A\boldsymbol{x} = \lambda\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} \quad \text{así} \quad \lambda = \frac{\boldsymbol{x}^{\mathrm{T}}A\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}} = \frac{\text{real}}{\text{real}}.$$

Encuentre la falla en este razonamiento: una suposición oculta injustificada. Puede seguir esos pasos con la matriz de rotación de $90°$ $[\,0 \;\; -1;\;\; 1 \;\; 0\,]$ con $\lambda = i$ y $\boldsymbol{x} = (i, 1)$.

5 Escriba S y B como $\lambda_1\boldsymbol{x}_1\boldsymbol{x}_1^{\mathrm{T}} + \lambda_2\boldsymbol{x}_2\boldsymbol{x}_2^{\mathrm{T}}$ del teorema espectral $Q\Lambda Q^{T}$:

$$S = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \qquad B = \begin{bmatrix} 9 & 12 \\ 12 & 16 \end{bmatrix} \quad \text{(mantenga } \|\boldsymbol{x}_1\| = \|\boldsymbol{x}_2\| = 1\text{)}.$$

6 (Recomendado) Esta matriz M es antisimétrica y también _____. Luego, todos sus autovalores λ son imaginarios puros con $|\lambda| = 1$. ($\|M\boldsymbol{x}\| = \|\boldsymbol{x}\|$ para todo $\boldsymbol{x}$ entonces $\|\lambda\boldsymbol{x}\| = \|\boldsymbol{x}\|$ para los autovectores.) Encuentre los cuatro autovalores a partir de la traza de M:

$$M = \frac{1}{\sqrt{3}}\begin{bmatrix} 0 & 1 & 1 & 1 \\ -1 & 0 & -1 & 1 \\ -1 & 1 & 0 & -1 \\ -1 & -1 & 1 & 0 \end{bmatrix} \quad \text{solo puede tener autovalores } i \text{ o } -i.$$

7 Muestre que esta A (**simétrica pero compleja**) tiene solo una recta de autovectores:

$$A = \begin{bmatrix} i & 1 \\ 1 & -i \end{bmatrix} \quad \text{no es ni siquiera diagonalizable: autovalores } \lambda = 0 \text{ y } 0.$$

$A^{\mathrm{T}} = A$ no es una propiedad especial para las matrices complejas. La buena propiedad es $\overline{\boldsymbol{A}}^{\mathbf{T}} = \boldsymbol{A}$. Entonces todos los autovalores son reales y A tiene n autovectores ortogonales.

8 Esta A es casi simétrica pero sus autovectores están lejos de ser ortogonales:

$$A = \begin{bmatrix} 1 & 10^{-15} \\ 0 & 1+10^{-15} \end{bmatrix} \quad \text{tiene autovectores} \quad \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{y} \quad \begin{bmatrix} ? \end{bmatrix}$$

¿ Cuál es el ángulo entre los autovectores ?

9 *¿ Qué matrices simétricas S también son ortogonales* ? Las que tienen $S^{\mathrm{T}} = S$ y $S^{\mathrm{T}} = S^{-1}$.

(a) Muestre cómo la simetría y la ortogonalidad llevan a $S^2 = I$.

(b) ¿ Cuáles son los autovalores posibles de S ? Describa todos los Λ posibles.

Entonces $S = Q\Lambda Q^{\mathrm{T}}$ para una de esas matrices de autovalores Λ y una Q ortogonal.

10 Si S es simétrica, muestre que $A^{\mathrm{T}}SA$ también es simétrica (tome la traspuesta de $A^{\mathrm{T}}SA$). Aquí A es m por n y S es m por m. ¿ Vale que autovalores de $S =$ autovalores de $A^{\mathrm{T}}SA$?

En caso de que A sea cuadrada e invertible, se dice que $A^{\mathrm{T}}SA$ es *congruente* con S.

Tienen la misma cantidad de autovalores positivos, negativos y cero: ***Ley de inercia***.

11 He aquí una manera de mostrar que a está *entre* los autovalores λ_1 y λ_2 de S :

$$S = \begin{bmatrix} a & b \\ b & c \end{bmatrix} \qquad \begin{array}{l} \det(S - \lambda I) = \lambda^2 - a\lambda - c\lambda + ac - b^2 \\ \text{es una parábola hacia arriba (debido a } \lambda^2) \end{array}$$

Muestre que $\det(S - \lambda I)$ es negativo en $\lambda = a$. Entonces la parábola cruza el eje a la izquierda y a la derecha de $\lambda = a$. Lo cruza en los dos autovalores de S, que entonces deben encerrar a a.

Los $n - 1$ autovalores de A siempre están entre los n autovalores de $S = \begin{bmatrix} A & \boldsymbol{b} \\ \boldsymbol{b}^{\mathrm{T}} & c \end{bmatrix}$.

En la Sección III.2 se explicará el entrelazado de autovalores.

12 La energía $\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x} = 2x_1x_2$ tiene ciertamente un punto de ensilladura y no un mínimo en $(0, 0)$. ¿ Qué matriz simétrica S produce esta energía ? ¿ Cuáles son sus autovalores ?

13 Verifique alguna condición para ver si $A^{\mathrm{T}}A$ es definida positiva en cada caso: A necesita columnas independientes.

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix} \quad \text{y} \quad A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 2 & 1 \end{bmatrix} \quad \text{y} \quad A = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

14 Encuentre la matriz S de 3 por 3 y sus pivotes, rango, autovalores, y determinante:

$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} \quad & \quad & \quad \\ & S & \\ \quad & \quad & \quad \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 4(x_1 - x_2 + 2x_3)^2.$$

15 Calcule los tres determinantes superiores izquierdos de S para establecer si es definida positiva. Verifique que sus cocientes den el segundo y tercer pivote.

$$\textbf{Pivotes} = \textbf{cocientes de determinantes} \qquad S = \begin{bmatrix} 2 & 2 & 0 \\ 2 & 5 & 3 \\ 0 & 3 & 8 \end{bmatrix}.$$

16 ¿Para qué números c y d son S y T definidas positivas? Verifique sus 3 determinantes:

$$S = \begin{bmatrix} c & 1 & 1 \\ 1 & c & 1 \\ 1 & 1 & c \end{bmatrix} \qquad \text{y} \qquad T = \begin{bmatrix} 1 & 2 & 3 \\ 2 & d & 4 \\ 3 & 4 & 5 \end{bmatrix}.$$

17 Encuentre una matriz con $a > 0$ y $c > 0$ y $a + c > 2b$ que tenga un autovalor negativo.

18 Una matriz definida positiva no puede tener un cero (o peor, un número negativo) en su diagonal principal. Muestre que esta matriz no verifica $\boldsymbol{x}^{\mathrm{T}} S \boldsymbol{x} > 0$:

$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} 4 & 1 & 1 \\ 1 & 0 & 2 \\ 1 & 2 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \text{ no es positivo cuando } (x_1, x_2, x_3) = (\quad , \quad , \quad).$$

19 Un elemento s_{jj} en la diagonal de una matriz simétrica no puede ser menor a todos los λ. Si lo fuera, entonces $S - s_{jj} I$ tendría autovalores _____ y sería definida positiva. Pero $S - s_{jj} I$ tiene un _____ en la diagonal principal, imposible según el Problema 18.

20 A partir de $S = Q \Lambda Q^{\mathrm{T}}$ calcule la raíz cuadrada simétrica y definida positiva $Q \sqrt{\Lambda} Q^{\mathrm{T}}$ de cada matriz. Verifique que la raíz cuadrada cumple $A^{\mathrm{T}} A = S$:

$$S = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix} \qquad \text{y} \qquad S = \begin{bmatrix} 10 & 6 \\ 6 & 10 \end{bmatrix}.$$

21 Dibuje la elipse inclinada $x^2 + xy + y^2 = 1$ y encuentre las longitudes de sus ejes a partir de los autovalores de su correspondiente matriz S.

22 En la factorización de Cholesky $S = A^{\mathrm{T}} A$, con $A = \sqrt{D} L^{\mathrm{T}}$, las raices cuadradas de los pivotes están en la diagonal de A. Encuentre A (triangular superior) para

$$S = \begin{bmatrix} 9 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 2 & 8 \end{bmatrix} \qquad \text{y} \qquad S = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 7 \end{bmatrix}.$$

23 Suponga que C es definida positiva ($\boldsymbol{y}^{\mathrm{T}} C \boldsymbol{y} > 0$ para todo $\boldsymbol{y} \neq \boldsymbol{0}$) y A tiene columnas independientes ($A\boldsymbol{x} \neq \boldsymbol{0}$ para todo $\boldsymbol{x} \neq \boldsymbol{0}$). Aplique la condición de la energía a $\boldsymbol{x}^{\mathrm{T}} A^{\mathrm{T}} C A \boldsymbol{x}$ para mostrar que $\boldsymbol{S = A^{\mathrm{T}} C A}$ ***es definida positiva: matriz crucial en ingeniería.***

El mínimo de una función $F(x, y, z)$

¿ Qué condición esperaría en un punto mínimo ? Primero que las pendientes sean nulas :

Las primeras derivadas son cero $\dfrac{\partial F}{\partial x} = \dfrac{\partial F}{\partial y} = \dfrac{\partial F}{\partial z} = \mathbf{0}$ en el punto mínimo.

Luego le sigue la versión del álgebra lineal de la típica condición de cálculo $d^2f/dx^2 > 0$:

La matriz de derivadas segundas H es definida positiva $H = \begin{bmatrix} \boldsymbol{F_{xx}} & \boldsymbol{F_{xy}} & \boldsymbol{F_{xz}} \\ \boldsymbol{F_{yx}} & \boldsymbol{F_{yy}} & \boldsymbol{F_{yz}} \\ \boldsymbol{F_{zx}} & \boldsymbol{F_{zy}} & \boldsymbol{F_{zz}} \end{bmatrix}$

Aquí $\boldsymbol{F_{xy}} = \dfrac{\partial}{\partial x}\left(\dfrac{\partial F}{\partial y}\right) = \dfrac{\partial}{\partial y}\left(\dfrac{\partial F}{\partial x}\right) = \boldsymbol{F_{yx}}$ es una derivada segunda "mixta".

24 Para $F_1(x, y) = \frac{1}{4}x^4 + x^2y + y^2$ y $F_2(x, y) = x^3 + xy - x$ encuentre las matrices de derivadas segundas H_1 y H_2 (las **matrices hessianas**) :

Condición para el mínimo $H = \begin{bmatrix} \partial^2F/\partial x^2 & \partial^2F/\partial x\partial y \\ \partial^2F/\partial y\partial x & \partial^2F/\partial y^2 \end{bmatrix}$ es definida positiva

H_1 es definida positiva así que F_1 es cóncava hacia arriba (= convexa). Encuentre el punto mínimo de F_1. *Encuentre el punto de ensilladura de* F_2 (busque solo donde las primeras derivadas son cero).

25 ¿ Qué valores de c dan un bol y cuáles c dan un punto de ensilladura en el gráfico de $z = 4x^2 + 12xy + cy^2$? Describa este gráfico para el valor de c donde se produce el cambio.

26 Sin multiplicar $S = \begin{bmatrix} \cos\theta & -\operatorname{sen}\theta \\ \operatorname{sen}\theta & \cos\theta \end{bmatrix}\begin{bmatrix} 2 & 0 \\ 0 & 5 \end{bmatrix}\begin{bmatrix} \cos\theta & \operatorname{sen}\theta \\ -\operatorname{sen}\theta & \cos\theta \end{bmatrix}$, encontrar

(a) el determinante de S (b) los autovalores de S

(c) los autovectores de S (d) una razón por la que S es simétrica y definida positiva.

27 ¿ Para cuáles a y c es esta matriz definida positiva ? ¿ Para cuáles a y c es semidefinida positiva (esto incluye definida) ?

$$S = \begin{bmatrix} a & a & a \\ a & a+c & a-c \\ a & a-c & a+c \end{bmatrix}$$

Las 5 condiciones son posibles.
La energía $\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}$ es igual a
$a\,(x_1 + x_2 + x_3)^2 + c\,(x_2 - x_3)^2$.

28 ¡**Importante**! Sea S definida positiva con autovalores $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$.

(a) ¿ Cuáles son los autovalores de la matriz $\lambda_1 I - S$? ¿ Es semidefinida positiva?

(b) ¿ Cómo se prueba que $\lambda_1 \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} \geq \boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}$ para todo $\boldsymbol{x}$?

(c) Llegue a la siguiente conclusión: **el valor máximo de $\boldsymbol{x^{\mathrm{T}}Sx/x^{\mathrm{T}}x}$ es λ_1**.

Observación Otra manera de llegar a **28** (c) : **maximizar** $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x}$ sujeto a la condición $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = \mathbf{1}$.

Esto nos lleva a $\dfrac{\partial}{\partial \boldsymbol{x}}\left[\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x} - \lambda\left(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} - 1\right)\right] = \mathbf{0}$ y luego $S\boldsymbol{x} = \lambda\boldsymbol{x}$ y $\lambda = \lambda_1$.

I.8. Valores singulares y vectores singulares en la SVD

Las buenas matrices (matrices reales simétricas S) tienen autovalores reales y autovectores ortogonales. Pero para otras matrices, los autovalores pueden ser complejos o los autovectores no ser ortogonales. Si A no es cuadrada entonces $A\boldsymbol{x} = \lambda \boldsymbol{x}$ es imposible y los autovectores fallan (el lado izquierdo está en $\mathbf{R}^m$ y el lado derecho en $\mathbf{R}^n$). Se necesita una idea que funcione para cualquier matriz.

La descomposición en valores singulares (SVD por su sigla en inglés) cubre este hueco de una manera perfecta. En nuestras aplicaciones, A suele ser una matriz de datos. Las filas podrían indicar una la edad y otra la altura de 1000 niños. Así A es de 2 por 1000: definitivamente rectangular. A menos que la altura sea exactamente proporcional a la edad, el rango es $r = 2$ y la matriz A tiene dos valores singulares positivos σ_1 y σ_2.

El punto clave es que necesitamos **dos grupos de vectores singulares**, los $\boldsymbol{u}$ y los $\boldsymbol{v}$. Para una matriz real de m por n, los n vectores singulares a derecha $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ son ortogonales en $\mathbf{R}^n$. Los m vectores singulares a izquierda $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m$ son perpendiculares en $\mathbf{R}^m$. La conexión entre los n $\boldsymbol{v}$ y los m $\boldsymbol{u}$ no es $A\boldsymbol{x} = \lambda \boldsymbol{x}$. Esto es para autovectores. **Para vectores singulares, cada $\boldsymbol{Av}$ es igual a $\boldsymbol{\sigma u}$**:

$$\boxed{\boldsymbol{Av_1} = \boldsymbol{\sigma_1 u_1} \quad \cdots \quad \boldsymbol{Av_r} = \boldsymbol{\sigma_r u_r}} \qquad \boxed{\boldsymbol{Av_{r+1}} = \boldsymbol{0} \quad \cdots \quad \boldsymbol{Av_n} = \boldsymbol{0}} \tag{1}$$

Hemos separado los r primeros $\boldsymbol{v}$ y $\boldsymbol{u}$ del resto. Ese número r es el *rango de* A, el número de columnas (y filas) independientes. Así r es la dimensión del espacio columna y del espacio fila. **Tendremos $\boldsymbol{r}$ valores singulares en orden descendente $\boldsymbol{\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0}$.** Los últimos $n - r$ vectores $\boldsymbol{v}$ pertenecen al espacio nulo de A, y los últimos $m - r$ vectores $\boldsymbol{u}$ pertenecen al espacio nulo de A^{T}.

Nuestro primer paso es escribir las ecuaciones (1) en forma matricial. Todos los vectores singulares a derecha $\boldsymbol{v}_1$ a $\boldsymbol{v}_n$ son las columnas de V. Los autovectores a izquierda $\boldsymbol{u}_1$ a $\boldsymbol{u}_m$ son las columnas de U. Estas son **matrices cuadradas ortogonales** ($V^{\mathrm{T}} = V^{-1}$ y $U^{\mathrm{T}} = U^{-1}$) porque sus columnas son vectores unitarios ortogonales. La ecuación (1) es la SVD completa, con matrices cuadradas V y U:

$$\boxed{\boldsymbol{AV = U\Sigma} \qquad A \begin{bmatrix} \\ \boldsymbol{v}_1 \,..\, \boldsymbol{v}_r \,..\, \boldsymbol{v}_n \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \boldsymbol{u}_1 \,..\, \boldsymbol{u}_r \,..\, \boldsymbol{u}_m \\ \\ \end{bmatrix} \left[\begin{array}{ccc|c} \boldsymbol{\sigma_1} & & & \\ & \ddots & & \boldsymbol{0} \\ & & \boldsymbol{\sigma_r} & \\ \hline & \boldsymbol{0} & & \boldsymbol{0} \end{array}\right]} \tag{2}$$

Se ve la igualdad $A\boldsymbol{v}_k = \sigma_k \boldsymbol{u}_k$ en las primeras r columnas. Esto es una parte importante de la SVD. Muestra la base de los $\boldsymbol{v}$ para el espacio fila de A y de los $\boldsymbol{u}$ para el espacio columna. Sobre la diagonal principal de Σ están los números positivos $\sigma_1, \ldots, \sigma_r$, y el resto de la matriz es cero por los espacios nulos de A y A^{T}.

Los autovectores también satisfacen $AX = X\Lambda$. Pero $AV = U\Sigma$ necesita **dos grupos de vectores singulares**.

Ejemplo 1 $\quad \begin{bmatrix} \mathbf{3} & \mathbf{0} \\ \mathbf{4} & \mathbf{5} \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{1} & \mathbf{-1} \\ \mathbf{1} & \mathbf{1} \end{bmatrix} = \frac{1}{\sqrt{10}} \begin{bmatrix} \mathbf{1} & \mathbf{-3} \\ \mathbf{3} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{3\sqrt{5}} & \\ & \mathbf{\sqrt{5}} \end{bmatrix}$

$AV = U\Sigma$

La matriz A no es simétrica, así que V no es igual a U. El rango es 2, por lo que hay dos valores singulares $\sigma_1 = 3\sqrt{5}$ y $\sigma_2 = \sqrt{5}$. El producto $3 \cdot 5 = 15$ es el determinante de A (en este aspecto los valores singulares son como los autovalores). Las columnas de V son ortogonales y las columnas de U son ortogonales. Esas columnas son vectores unitarios cuando se dividen por $\sqrt{2}$ y $\sqrt{10}$, de esa manera **V y U son matrices ortogonales**: $V^{\mathrm{T}} = V^{-1}$ y $U^{\mathrm{T}} = U^{-1}$.

Esta ortogonalidad nos permite ir de $AV = U\Sigma$ a la expresión de la SVD más usual y famosa: multiplicando ambos miembros de $AV = U\Sigma$ por $V^{-1} = V^{\mathrm{T}}$ a la derecha.

$$\boxed{\textbf{La descomposición en valores singulares de } \boldsymbol{A} \textbf{ es } \quad \boldsymbol{A = U\Sigma V^{\mathrm{T}}}.} \tag{3}$$

Luego la multiplicación columna-fila de $U\Sigma$ por V^{T} separa A en r sumandos de rango 1:

$$\textbf{Sumandos de la SVD} \qquad \boxed{\boldsymbol{A = U\Sigma V^{\mathrm{T}} = \sigma_1 u_1 v_1^{\mathrm{T}} + \cdots + \sigma_r u_r v_r^{\mathrm{T}}}.} \tag{4}$$

En el ejemplo de 2 por 2, el primer sumando es más importante que el segundo sumando ya que $\sigma_1 = 3\sqrt{5}$ es más grande que $\sigma_2 = \sqrt{5}$. Para recuperar A, se suma $\sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^{\mathrm{T}} + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^{\mathrm{T}}$:

$$\frac{3\sqrt{5}}{\sqrt{10}\sqrt{2}} \begin{bmatrix} 1 \\ 3 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} + \frac{\sqrt{5}}{\sqrt{10}\sqrt{2}} \begin{bmatrix} -3 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \end{bmatrix} = \frac{3}{2} \begin{bmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{3} & \mathbf{3} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \mathbf{3} & \mathbf{-3} \\ \mathbf{-1} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{3} & \mathbf{0} \\ \mathbf{4} & \mathbf{5} \end{bmatrix}$$

Esto se simplifica ya que $\sqrt{5}/\sqrt{10}\sqrt{2}$ es igual a $1/2$. Observe que los vectores singulares a derecha $(1, 1)$ y $(-1, 1)$ en V son las traspuestas de las filas $\boldsymbol{v}_1^{\mathrm{T}}, \boldsymbol{v}_2^{\mathrm{T}}$ de V^{T}. ¡ No hemos explicado aún como se calculan V y U y Σ !

Forma reducida de la SVD

La forma completa $AV = U\Sigma$ en (2) puede tener muchos ceros en Σ cuando el rango de A es pequeño y el espacio nulo es grande. Esos ceros no contribuyen nada a la multiplicación matricial. El corazón de la SVD está en los primeros r vectores $\boldsymbol{v}$ y $\boldsymbol{u}$ y escalares σ. Podemos reducir $AV = U\Sigma$ a $AV_r = U_r\Sigma_r$ quitando las partes que producirán ceros. Esto nos lleva a la forma **reducida de la SVD donde ahora $\boldsymbol{\Sigma_r}$ es cuadrada:**

$$\boxed{\boldsymbol{AV_r = U_r\Sigma_r} \qquad A \begin{bmatrix} & & \\ \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_r \\ \text{espacio fila} & & \end{bmatrix} = \begin{bmatrix} & & \\ \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_r \\ \text{espacio columna} & & \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix}} \tag{5}$$

Ya que $\boldsymbol{v}$ y $\boldsymbol{u}$ son ortogonales, aún es cierto que $V_r^{\mathrm{T}} V_r = I_r$ y $U_r^{\mathrm{T}} U_r = I_r$. Pero cuando V_r y U_r no son cuadradas, no podemos seguir teniendo inversas bilaterales: $V_r V_r^{\mathrm{T}} \neq I$ y $U_r U_r^{\mathrm{T}} \neq I$.

Ejemplo $V_r = \begin{bmatrix} 1/3 \\ 2/3 \\ 2/3 \end{bmatrix}$ y $V_r^{\mathrm{T}} V_r = \begin{bmatrix} 1 \end{bmatrix}$ pero $V_r V_r^{\mathrm{T}} = \frac{1}{9} \begin{bmatrix} 1 & 2 & 2 \\ 2 & 4 & 4 \\ 2 & 4 & 4 \end{bmatrix} =$ rango 1.

El Problema 22 muestra que **aún es cierto que $\boldsymbol{A = U_r \Sigma_r V_r^{\mathrm{T}}}$**. El resto de $U \Sigma V^{\mathrm{T}}$ no contribuye a A, porque son bloques de ceros en Σ. La fórmula importante sigue siendo $A = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^{\mathrm{T}} + \cdots + \sigma_r \boldsymbol{u}_r \boldsymbol{v}_r^{\mathrm{T}}$. La SVD solo ve los r valores no nulos en la matriz diagonal Σ.

Hecho importante en ciencia de datos

¿Por qué la SVD es tan importante para este tema y para este libro? Como las otras factorizaciones $A = LU$ y $A = QR$ y $S = Q \Lambda Q^{\mathrm{T}}$, también separa la matriz en suma de matrices de rango uno. Una propiedad particular de la SVD es que **esos sumandos aparecen en orden de importancia**. El primer sumando $\sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^{\mathrm{T}}$ es la matriz de rango uno más cercana a A. Aún más: *la suma de los primeros k sumandos es la matriz más cercana posible de rango k.*

$\boldsymbol{A_k = \sigma_1 u_1 v_1^{\mathrm{T}} + \cdots + \sigma_k u_k v_k^{\mathrm{T}}}$ es la mejor aproximación a $\boldsymbol{A}$ de rango $\boldsymbol{k}$:

Eckart-Young **Si $\boldsymbol{B}$ tiene rango $\boldsymbol{k}$ entonces $\boldsymbol{||A - A_k|| \leq ||A - B||}$.** (6)

Para interpretar esta afirmación tenemos que darle un sentido al símbolo $||A - B||$. Esto es la "**norma**" de la matriz $A - B$, una medida de su tamaño (como lo es el valor absoluto de un número). El teorema de Eckart-Young se probará en la Sección I.9.

Para obtener la SVE, nuestro primer trabajo es encontrar los $\boldsymbol{v}$ y los $\boldsymbol{u}$ ortogonales para los que se cumple $A\boldsymbol{v} = \sigma \boldsymbol{u}$ en la ecuación (1).

Primera prueba de la SVD

Nuestro objetivo es $A = U \Sigma V^{\mathrm{T}}$. Queremos identificar los dos grupos de vectores singulares, los $\boldsymbol{u}$ y los $\boldsymbol{v}$. Una manera de encontrar esos vectores es formar las matrices simétricas $A^{\mathrm{T}} A$ y $A A^{\mathrm{T}}$:

$$\boldsymbol{A^{\mathrm{T}} A} = (V \Sigma^{\mathrm{T}} U^{\mathrm{T}})\ (U \Sigma V^{\mathrm{T}}) = \boldsymbol{V \Sigma^{\mathrm{T}} \Sigma V^{\mathrm{T}}} \tag{7}$$

$$\boldsymbol{A A^{\mathrm{T}}} = (U \Sigma V^{\mathrm{T}})\ (V \Sigma^{\mathrm{T}} U^{\mathrm{T}}) = \boldsymbol{U \Sigma \Sigma^{\mathrm{T}} U^{\mathrm{T}}} \tag{8}$$

Ambas expresiones (7) y (8) producen matrices simétricas. En general, $A^{\mathrm{T}} A$ y $A A^{\mathrm{T}}$ son diferentes. Ambos miembros derechos tienen la forma especial $Q \Lambda Q^{\mathrm{T}}$. Los autovalores están en $\Lambda = \Sigma^{\mathrm{T}} \Sigma$ o en $\Lambda = \Sigma \Sigma^{\mathrm{T}}$. **Los autovectores están en $\boldsymbol{Q = V}$ o $\boldsymbol{Q = U}$.** Así sabemos, a partir de (7) y (8), cómo V, U y Σ se relacionan con las matrices simétricas $A^{\mathrm{T}} A$ y $A A^{\mathrm{T}}$.

> **V contiene los autovectores ortonormales de $A^{\mathrm{T}}A$**
>
> **U contiene los autovectores ortonormales de AA^{T}**
>
> **σ_1^2 a σ_r^2 son los autovalores no nulos de $A^{\mathrm{T}}A$ y de AA^{T}**

No hemos terminado aún, por la siguiente razón: **la SVD requiere que $Av_k = \sigma_k u_k$.** Se debe conectar cada vector singular a derecha v_k a un vector singular a izquierda u_k, para $k = 1, \ldots, r$. Cada elección de v, determina el signo de u. Si $Su = \lambda u$ entonces también $S(-u) = \lambda(-u)$ y tenemos que saber el signo correcto. Además, hay un plano entero de autovectores cuando λ es un autovalor doble. Cuando elijo dos v en ese plano, $Av = \sigma u$ decidirá ambos vectores u. Esto está en la ecuación (9).

El plan es comenzar con los v. **Elegimos autovectores ortonormales $v_1, \ldots, v_r$** de $A^{\mathrm{T}}A$. **Luego elegimos $\sigma_k = \sqrt{\lambda_k}$. Para determinar cada u pedimos que $Av = \sigma u$:**

$$\boxed{v \text{ entonces } u \qquad A^{\mathrm{T}}Av_k = \sigma_k^2 v_k \quad \text{y así} \quad u_k = \frac{Av_k}{\sigma_k} \quad \text{para} \quad k = 1, \ldots, r} \tag{9}$$

¡ Esta es la demostración de la SVD ! Verifiquemos que esos u sean autovectores de AA^{T} :

$$AA^{\mathrm{T}}u_k = AA^{\mathrm{T}}\left(\frac{Av_k}{\sigma_k}\right) = A\left(\frac{A^{\mathrm{T}}A\,v_k}{\sigma_k}\right) = A\,\frac{\sigma_k^2 v_k}{\sigma_k} = \sigma_k^2 u_k. \tag{10}$$

Los v fueron elegidos ortonormales. Debemos chequear que los u también lo sean :

$$u_j^{\mathrm{T}}u_k = \left(\frac{Av_j}{\sigma_j}\right)^{\mathrm{T}}\left(\frac{Av_k}{\sigma_k}\right) = \frac{v_j^{\mathrm{T}}(A^{\mathrm{T}}Av_k)}{\sigma_j\,\sigma_k} = \frac{\sigma_k}{\sigma_j}\,v_j^{\mathrm{T}}\,v_k = \begin{cases} 1 & \text{si } j = k \\ 0 & \text{si } j \neq k \end{cases} \tag{11}$$

La igualdad $(AA^{\mathrm{T}})A = A(A^{\mathrm{T}}A)$ fue fundamental para la ecuación (10). La propiedad $(AB)C = A(BC)$ es fundamental para muchas pruebas en álgebra lineal. Mover los paréntesis es una idea muy poderosa. Esta es la *ley asociativa*.

Finalmente tenemos que elegir los últimos $n-r$ vectores v_{r+1} a v_n y los últimos $m-r$ vectores u_{r+1} a u_m. Esto es fácil. **Esos v y u están en los espacios nulos de A y de A^{T}.** Podemos elegir cualquier base ortonormal de estos espacios nulos. Automáticamente serán ortogonales a los primeros v que están en el espacio fila de A y a los primeros u que están en el espacio columna.
Esto se debe a que los espacios son ortogonales: $\mathbf{N}(A) \perp \mathbf{C}(A^{\mathrm{T}})$ y $\mathbf{N}(A^{\mathrm{T}}) \perp \mathbf{C}(A)$. *La prueba de la SVD está completa.*

Ahora tenemos las matrices U, V y Σ en la SVD completa de la ecuación (2). Habrá notado que los autovalores de $A^{\mathrm{T}}A$ están en $\Sigma^{\mathrm{T}}\Sigma$, y *los mismos números σ_1^2 a σ_r^2 son también autovalores de AA^{T} en $\Sigma\Sigma^{\mathrm{T}}$*. Un hecho sorprendente : **BA siempre tiene los mismos autovalores no nulos que AB : 5 páginas más adelante.**

Ejemplo 1 (completo) Encuentre las matrices U, Σ, V de $A = \begin{bmatrix} \mathbf{3} & \mathbf{0} \\ \mathbf{4} & \mathbf{5} \end{bmatrix}$.

Con rango 2, esta A tiene dos valores singulares positivos σ_1 y σ_2. Veremos que σ_1 es más grande que $\lambda_{\text{mx}} = 5$, y σ_2 es más pequeño que $\lambda_{\text{mn}} = 3$. Comencemos con $A^{\mathrm{T}}A$ y AA^{T} :

$$A^{\mathrm{T}}A = \begin{bmatrix} 25 & 20 \\ 20 & 25 \end{bmatrix} \qquad AA^{\mathrm{T}} = \begin{bmatrix} 9 & 12 \\ 12 & 41 \end{bmatrix}$$

Esas matrices tienen la misma traza (50) y los mismos autovalores $\sigma_1^2 = 45$ y $\sigma_2^2 = 5$. Las raíces cuadradas son $\sigma_1 = \sqrt{\mathbf{45}}$ y $\sigma_2 = \sqrt{\mathbf{5}}$. Entonces $\sigma_1\sigma_2 = 15$ que es el determinante de A.

Un paso clave es encontrar los autovectores de $A^{\mathrm{T}}A$ (con autovalores 45 y 5) :

$$\begin{bmatrix} 25 & 20 \\ 20 & 25 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \mathbf{45} \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix} \qquad \begin{bmatrix} 25 & 20 \\ 20 & 25 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \mathbf{5} \begin{bmatrix} \mathbf{-1} \\ \mathbf{1} \end{bmatrix}$$

Así $\boldsymbol{v}_1$ y $\boldsymbol{v}_2$ son aquellos autovectores ortogonales re-escalados a norma 1. Dividimos por $\sqrt{2}$.

Vectores singulares a derecha $\boldsymbol{v}_1 = \dfrac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ $\boldsymbol{v}_2 = \dfrac{1}{\sqrt{2}}\begin{bmatrix} -1 \\ 1 \end{bmatrix}$

Vectores singulares a izquierda $\boldsymbol{u}_i = \dfrac{A\boldsymbol{v}_i}{\sigma_i}$

Ahora calculemos $A\boldsymbol{v}_1$ y $A\boldsymbol{v}_2$ que serán iguales a $\sigma_1\boldsymbol{u}_1 = \sqrt{45}\,\boldsymbol{u}_1$ y $\sigma_2\boldsymbol{u}_2 = \sqrt{5}\,\boldsymbol{u}_2$:

$$A\boldsymbol{v}_1 = \frac{3}{\sqrt{2}}\begin{bmatrix} 1 \\ 3 \end{bmatrix} = \sqrt{45}\frac{\mathbf{1}}{\sqrt{\mathbf{10}}}\begin{bmatrix} \mathbf{1} \\ \mathbf{3} \end{bmatrix} = \sigma_1\,\boldsymbol{u}_1$$

$$A\boldsymbol{v}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} -3 \\ 1 \end{bmatrix} = \sqrt{5}\,\frac{\mathbf{1}}{\sqrt{\mathbf{10}}}\begin{bmatrix} \mathbf{-3} \\ \mathbf{1} \end{bmatrix} = \sigma_2\,\boldsymbol{u}_2$$

La división por $\sqrt{10}$ hace que $\boldsymbol{u}_1$ y $\boldsymbol{u}_2$ sean ortonormales. Así $\sigma_1 = \sqrt{45}$ y $\sigma_2 = \sqrt{5}$ como se esperaba. La descomposición en valores singulares de A es U por Σ por V^{T}.

$$\boldsymbol{U} = \frac{1}{\sqrt{10}}\begin{bmatrix} 1 & -3 \\ 3 & 1 \end{bmatrix} \qquad \boldsymbol{\Sigma} = \begin{bmatrix} \sqrt{45} & \\ & \sqrt{5} \end{bmatrix} \qquad \boldsymbol{V} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \tag{12}$$

U y V contienen bases ortonormales del espacio columna y del espacio fila de A (ambos espacios son $\mathbf{R}^2$). El verdadero logro de esas dos bases es diagonalizar A: AV es igual a $U\Sigma$. La matriz $A = U\Sigma V^{\mathrm{T}}$ se divide en dos matrices de rango uno, columnas por filas.

$$\sigma_1\boldsymbol{u}_1\boldsymbol{v}_1^{\mathrm{T}} + \sigma_2\boldsymbol{u}_2\boldsymbol{v}_2^{\mathrm{T}} = \frac{\sqrt{\mathbf{45}}}{\sqrt{\mathbf{20}}}\begin{bmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{3} & \mathbf{3} \end{bmatrix} + \frac{\sqrt{\mathbf{5}}}{\sqrt{\mathbf{20}}}\begin{bmatrix} \mathbf{3} & \mathbf{-3} \\ \mathbf{-1} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{3} & \mathbf{0} \\ \mathbf{4} & \mathbf{5} \end{bmatrix} = \boldsymbol{A}.$$

Toda matriz es suma de matrices de rango uno con vectores ortogonales $\boldsymbol{u}$ y vectores ortogonales $\boldsymbol{v}$.

Pregunta: **Si $S = Q\Lambda Q^{\mathrm{T}}$ es simétrica definida positiva, ¿ cuál es su SVD ?**

Respuesta: La SVD es exactamente $U\Sigma V^{\mathrm{T}} = Q\Lambda Q^{\mathrm{T}}$. La matriz $U = V = Q$ es ortogonal. La matriz de autovalores Λ es la matriz de valores singulares Σ.

Pregunta: Si $S = Q\Lambda Q^{\mathrm{T}}$ tiene un autovalor negativo ($S\boldsymbol{x} = -\alpha\,\boldsymbol{x}$), ¿ cuáles son los valores singulares y cuáles son los vectores $\boldsymbol{v}$ y $\boldsymbol{u}$?

Respuesta: El valor singular será $\boldsymbol{\sigma} = +\boldsymbol{\alpha}$ (positivo). Un vector singular ($\boldsymbol{u}$ o $\boldsymbol{v}$) debe ser $-\boldsymbol{x}$ (cambiamos el signo). Entonces $S\boldsymbol{x} = -\alpha\,\boldsymbol{x}$ es igual a $S\boldsymbol{v} = \sigma\boldsymbol{u}$. El cambio en los dos signos se cancela.

Pregunta: Si $A = Q$ es ortogonal, ¿ porqué todos los valores singulares son iguales a 1 ?

Respuesta: Todos los valores singulares son $\sigma = 1$ porque $A^{\mathrm{T}}A = Q^{\mathrm{T}}Q = I$. Entonces $\Sigma = I$. Pero $U = Q$ y $V = I$ es solo una de las posibles elecciones de vectores singulares para $\boldsymbol{u}$ y$\boldsymbol{v}$:

$$Q = U\Sigma V^{\mathrm{T}} \text{ puede ser } Q = QII^{\mathrm{T}} \text{ o cualquier } Q = (QQ_1)IQ_1^{\mathrm{T}}.$$

Pregunta: **¿Por qué los autovalores de una matriz cuadrada A son menores o iguales que $\boldsymbol{\sigma_1}$?**

Respuesta: Multiplicar por matrices ortogonales U y V^{T} no cambia la norma de los vectores :

$$||A\boldsymbol{x}|| = ||U\Sigma V^{\mathrm{T}}\boldsymbol{x}|| = ||\Sigma V^{\mathrm{T}}\boldsymbol{x}|| \le \sigma_1||V^{\mathrm{T}}\boldsymbol{x}|| = \boldsymbol{\sigma_1}||\boldsymbol{x}|| \text{ para todo } \boldsymbol{x}. \tag{13}$$

Un autovector cumple $||A\boldsymbol{x}|| = |\lambda|\,||\boldsymbol{x}||$. Luego de (13) se tiene $|\lambda|\,||\boldsymbol{x}|| \le \sigma_1\,||\boldsymbol{x}||$ y $\boldsymbol{|\lambda| \le \sigma_1}$.

Pregunta: Si $A = \boldsymbol{x}\boldsymbol{y}^{\mathrm{T}}$ tiene rango 1, ¿ quiénes son $\boldsymbol{u}_1$, $\boldsymbol{v}_1$ y σ_1 ? **Verificar que $\boldsymbol{|\lambda_1| \le \sigma_1}$.**

Respuesta: Los vectores singulares $\boldsymbol{u}_1 = \boldsymbol{x}/||\boldsymbol{x}||$ y $\boldsymbol{v}_1 = \boldsymbol{y}/||\boldsymbol{y}||$ tienen norma 1. Entonces $\boldsymbol{\sigma}_1 = ||\boldsymbol{x}||\,||\boldsymbol{y}||$ es el único número que no es cero en la matriz de valores singulares Σ. Esta es la SVD :

$$\textbf{Matriz de rango 1} \qquad \boldsymbol{x}\boldsymbol{y}^{\mathrm{T}} = \frac{\boldsymbol{x}}{||\boldsymbol{x}||}\left(||\boldsymbol{x}||\,||\boldsymbol{y}||\right)\frac{\boldsymbol{y}^{\mathrm{T}}}{||\boldsymbol{y}||} = \boldsymbol{u}_1\sigma_1\boldsymbol{v}_1^{\mathrm{T}}.$$

Observación El único autovalor no nulo de $A = \boldsymbol{x}\boldsymbol{y}^{\mathrm{T}}$ es $\lambda = \boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}$. El autovector es $\boldsymbol{x}$ ya que $(\boldsymbol{x}\boldsymbol{y}^{\mathrm{T}})\,\boldsymbol{x} = \boldsymbol{x}\,(\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}) = \lambda\boldsymbol{x}$. Entonces $|\lambda_1| = |\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}| \le \sigma_1 = ||\boldsymbol{y}||\,||\boldsymbol{x}||$.

La desigualdad clave $|\lambda_1| \le \sigma_1$ es precisamente la desigualdad de Schwarz.

Pregunta: **¿Qué es la transformación de Karhunen-Loève y cuál es su conexión con la SVD ?**

Respuesta: KL empieza con una matriz de covarianza V de un proceso aleatorio de media cero. V es simétrica y definida o semidefinida positiva. En general V puede ser una matriz infinita o una función de covarianza. En ese caso la expansión KL será una serie infinita.

Los autovectores de V, en orden decreciente de autovalores $\sigma_1^2 \ge \sigma_2^2 \ge \ldots \ge 0$, son las funciones de base $\boldsymbol{u}_i$ de la transformada KL. La expansión de cualquier vector $\boldsymbol{v}$ en una base ortonormal $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots$ es $\boldsymbol{v} = \sum(\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{v})\boldsymbol{u}_i$.

En este caso estocástico, esa transformación des-correlaciona el proceso aleatorio : los $\boldsymbol{u}_i$ son independientes. Más que eso, el orden de los autovalores significa que los primeros k términos, parando en $(\boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{v})\boldsymbol{u}_k$, minimizan el error cuadrático medio. Este hecho se corresponde con el teorema de Eckart-Young en la Sección I.9.

La transformada KL es la versión estocástica (aleatoria) del análisis de componentes principales.

La geometría de la SVD

La SVD separa una matriz en $A = U\Sigma V^{\mathrm{T}}$: (**ortogonal**) $\times$ (**diagonal**) $\times$ (**ortogonal**). Podemos dibujar esos pasos en dos dimensiones. Las matrices ortogonales U y V rotan el plano. La matriz diagonal Σ lo estira a lo largo de los ejes. La Figura I.10 muestra **rotación** por **estiramiento** por **rotación**. **Los vectores x en la circunferencia unitaria se transforman en una elipse al multiplicarlos por A.**

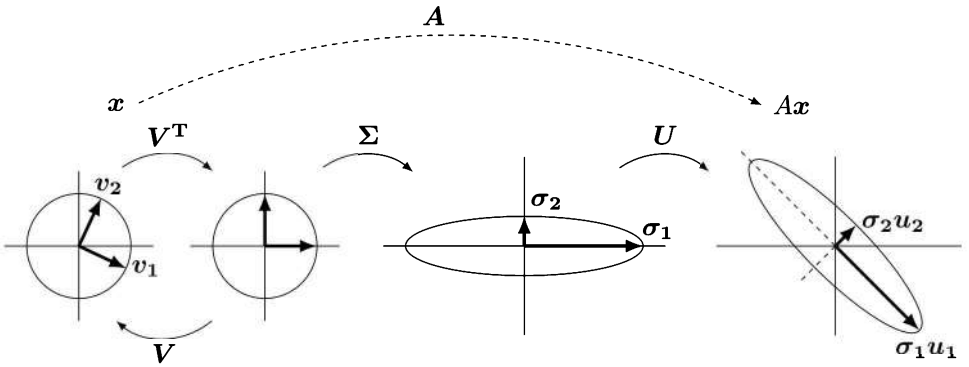


Figura I.10: U y V son rotaciones y posiblemente reflexiones. Σ estira un círculo en una elipse.

Esta figura es para matrices invertibles de 2 por 2 (porque $\sigma_1 > 0$ y $\sigma_2 > 0$). Primero es una rotación de cualquier $\boldsymbol{x}$ a $V^{\mathrm{T}}\boldsymbol{x}$. Luego Σ estira ese vector a $\Sigma V^{\mathrm{T}}\boldsymbol{x}$. Luego U lo rota a $A\boldsymbol{x} = U\Sigma V^{\mathrm{T}}\boldsymbol{x}$. Mantenemos todos los determinantes positivos para evitar reflexiones. Los cuatro números a, b, c, d en la matriz se conectan con los *dos ángulos* θ y ϕ y los *dos números* σ_1 y σ_2.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} \cos\theta & -\operatorname{sen}\theta \\ \operatorname{sen}\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \sigma_1 & \\ & \sigma_2 \end{bmatrix} \begin{bmatrix} \cos\phi & \operatorname{sen}\phi \\ -\operatorname{sen}\phi & \cos\phi \end{bmatrix}. \tag{14}$$

Pregunta. Si la matriz es simétrica entonces $b = c$ y A tiene solo 3 (no 4) parámetros. *¿Cómo se reducen a 3 los 4 números $\theta, \phi, \sigma_1, \sigma_2$ para una matriz simétrica S?*

El primer vector singular v_1

En la siguiente página estableceremos una nueva forma de ver a $\boldsymbol{v}_1$. En las páginas anteriores elegíamos los $\boldsymbol{v}$ como autovectores de $A^{\mathrm{T}}A$. Por supuesto esto sigue siendo cierto. Pero hay una manera valiosa de entender esos vectores singulares **uno por uno en lugar de todos juntos**. Comenzamos con $\boldsymbol{v}_1$ y el valor singular σ_1.

$$\boxed{\textbf{Maximizar el cociente } \frac{||A\boldsymbol{x}||}{||\boldsymbol{x}||}\textbf{. El máximo es } \boldsymbol{\sigma_1} \textbf{ en el vector } \boldsymbol{x} = \boldsymbol{v_1}\textbf{.}} \tag{15}$$

La elipse en la Figura I.10 muestra por qué el $\boldsymbol{x}$ que maximiza es $\boldsymbol{v}_1$. Cuando se sigue $\boldsymbol{v}_1$ de izquierda a derecha, se ve que termina en $A\boldsymbol{v}_1 = \sigma_1\boldsymbol{u}_1$ (el eje mayor de la elipse). Su norma empieza en $||\boldsymbol{v}_1|| = 1$ y termina en $||A\boldsymbol{v}_1|| = \sigma_1$.

¡ Pero nos interesa un enfoque independiente a la SVD ! Si no suponemos que ya conocemos U o Σ o V. ¿Cómo reconocemos que el cociente $||A\boldsymbol{x}||/||\boldsymbol{x}||$ tiene un máximo en $\boldsymbol{x} = \boldsymbol{v}_1$? El cálculo nos dice que las derivadas primeras deben ser cero. Las derivadas serán más fáciles si elevamos al cuadrado nuestra función :

$$\textbf{Problema : encontrar el máximo valor } \boldsymbol{\lambda} \textbf{ de } \frac{||A\boldsymbol{x}||^2}{||\boldsymbol{x}||^2} = \frac{\boldsymbol{x}^{\mathrm{T}} A^{\mathrm{T}} A\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}} = \frac{\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}. \tag{16}$$

Este "cociente de Rayleigh" depende de $x_1, \ldots, x_n$. El cálculo usa la regla del cociente, así que necesitamos

$$\frac{\partial}{\partial x_i}\left(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}\right) = \frac{\partial}{\partial x_i}\left(x_1^2 + \cdots + x_i^2 + \cdots + x_n^2\right) = 2(\boldsymbol{x})_i \tag{17}$$

$$\frac{\partial}{\partial x_i}\left(\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}\right) = \frac{\partial}{\partial x_i}\Big(\sum_i \sum_j S_{ij} x_i x_j\Big) = 2\sum_j S_{ij} x_j = 2\left(S\boldsymbol{x}\right)_i \tag{18}$$

Por la regla del cociente hallamos $\partial/\partial x_i\,(\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}/\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x})$. Igualamos esas n derivadas parciales de (16) a cero :

$$\left(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}\right) 2\Big(S\boldsymbol{x}\Big)_i - \left(\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}\right) 2\left(\boldsymbol{x}\right)_i = \boldsymbol{0} \text{ para } i = 1, \ldots, n \tag{19}$$

La ecuación (19) dice que el mejor $\boldsymbol{x}$ es un autovector de $\boldsymbol{S = A^{\mathrm{T}} A}$

$$\boxed{\textbf{2}S\boldsymbol{x} = \textbf{2}\boldsymbol{\lambda}\boldsymbol{x} \textbf{ y el máximo valor de } \frac{\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}} = \frac{||A\boldsymbol{x}||^2}{||\boldsymbol{x}||^2} \textbf{ es un autovalor } \boldsymbol{\lambda} \textbf{ de } \boldsymbol{S}.}$$

La búsqueda se reduce a los autovectores de $S = A^{\mathrm{T}} A$. El autovector que maximiza es $\boldsymbol{x} = \boldsymbol{v}_1$. El autovalor es $\lambda_1 = \sigma_1^2$. El cálculo ha confirmado la solución (15) del problema de máximo, el primer término de la SVD.

Para la SVD completa, necesitamos *todos* los vectores y valores singulares. Para encontrar $\boldsymbol{v}_2$ y σ_2, ajustamos el problema de maximizar a que solo busque vectores ortogonales $\boldsymbol{x}$ a $\boldsymbol{v}_1$.

$$\boxed{\textbf{Maximizar } \frac{||A\boldsymbol{x}||}{||\boldsymbol{x}||} \textbf{ bajo la condición } \boldsymbol{v_1^{\mathrm{T}} x = 0}. \textbf{ El máximo es } \boldsymbol{\sigma_2} \textbf{ en } \boldsymbol{x = v_2}.}$$

Los "multiplicadores de Lagrange" se inventaron para lidiar con restricciones en $\boldsymbol{x}$ como $\boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{x} = 0$. Y el Problema 3 da una manera directa y simple para tratar con esta condición $\boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{x} = 0$.

De la misma manera, cada vector singular $\boldsymbol{v}_{k+1}$ da el máximo cociente sobre todos los vectores $\boldsymbol{x}$ que son perpendiculares a los primeros $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$. Los vectores singulares a izquierda aparecerán de maximizar $||A^{\mathrm{T}}\boldsymbol{y}||/||\boldsymbol{y}||$. Siempre encontramos los ejes de una elipse y los autovectores de las matrices simétricas $A^{\mathrm{T}} A$ o AA^{T}.

Los vectores singulares de $A^{\mathbf{T}}$

La SVD conecta los $\boldsymbol{v}$ en el espacio fila con los $\boldsymbol{u}$ en el espacio columna. Cuando trasponemos $A = U\Sigma V^{\mathrm{T}}$, vemos que $\boldsymbol{A^{\mathrm{T}} = V\Sigma^{\mathrm{T}}U^{\mathrm{T}}}$ actúa en el sentido opuesto, desde los $\boldsymbol{u}$ a los $\boldsymbol{v}$:

$$\boldsymbol{A^{\mathrm{T}}u_k = \sigma_k v_k} \text{ para } \boldsymbol{k = 1, \ldots, r} \qquad \boldsymbol{A^{\mathrm{T}}u_k = 0} \text{ para } \boldsymbol{k = r+1, \ldots, m}. \tag{20}$$

Multiplicamos $A\boldsymbol{v}_k = \sigma_k \boldsymbol{u}_k$ por A^{T}. Recordamos que $A^{\mathrm{T}}A\boldsymbol{v}_k = \sigma_k^2 \boldsymbol{v}_k$ en (9). Dividimos por σ_k.

Una matriz simétrica diferente que también produce la SVD

Creamos la SVD a partir de dos matrices simétricas $A^{\mathrm{T}}A$ y AA^{T}. Otra buena manera usa una matriz S simétrica en bloques. *Esta matriz tiene r pares de autovalores opuestos*. Los autovalores no nulos de la matriz S son σ_k y $-\sigma_k$, y tiene tamaño $m + n$:

$$\boxed{\boldsymbol{S} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{A} \\ \boldsymbol{A^{\mathrm{T}}} & \boldsymbol{0} \end{bmatrix} \quad \text{tiene autovectores} \quad \begin{bmatrix} \boldsymbol{u}_k \\ \boldsymbol{v}_k \end{bmatrix} \text{ y } \begin{bmatrix} -\boldsymbol{u}_k \\ \boldsymbol{v}_k \end{bmatrix}.}$$

Podemos verificar esos autovectores directamente, recordando que $A\boldsymbol{v}_k = \sigma_k \boldsymbol{u}_k$ y $A^{\mathrm{T}}\boldsymbol{u}_k = \sigma_k \boldsymbol{v}_k$:

$$\begin{bmatrix} 0 & A \\ A^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \pm\boldsymbol{u}_k \\ \boldsymbol{v}_k \end{bmatrix} = \begin{bmatrix} A\boldsymbol{v}_k \\ \pm A^{\mathrm{T}}\boldsymbol{u}_k \end{bmatrix} = \sigma_k \begin{bmatrix} \boldsymbol{u}_k \\ \boldsymbol{v}_k \end{bmatrix} \text{ y } -\sigma_k \begin{bmatrix} -\boldsymbol{u}_k \\ \boldsymbol{v}_k \end{bmatrix}. \tag{21}$$

Eso da $2r$ autovalores. Los autovectores son ortogonales: $-\boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{u}_k + \boldsymbol{v}_k^{\mathrm{T}}\boldsymbol{v}_k = -1 + 1$. ¿Puede ver los otros $(m - r) + (n - r)$ autovectores correspondientes a $\lambda = 0$ para la matriz en bloques? Se deben involucrar el resto de los $\boldsymbol{u}$ y $\boldsymbol{v}$ del espacio nulo de A^{T} y A.

AB y BA: mismos autovalores no nulos

Si A es de $m \times n$ y B es de $n \times m$, entonces AB y BA tienen los mismos autovalores no nulos.

Comenzamos con $AB\boldsymbol{x} = \lambda\boldsymbol{x}$ y $\lambda \neq 0$. Multiplique ambos lados por B, para obtener $BAB\boldsymbol{x} = \lambda B\boldsymbol{x}$. Esto dice que $B\boldsymbol{x}$ es un autovector de BA con el mismo autovalor λ, esto es exactamente lo que queríamos. Necesitábamos $\lambda \neq 0$ para estar seguros que ese autovector $B\boldsymbol{x}$ no fuera nulo.

Notemos que si B es cuadrada e invertible, entonces $B^{-1}(BA)B = AB$. Esto dice que BA es *semejante* a AB: *los mismos autovalores*. Pero nuestra primera prueba permite que A y B sean de m por n y de n por m. Esto cubre el ejemplo importante de la SVD que ocurre cuando $B = A^{\mathrm{T}}$. En ese caso $A^{\mathrm{T}}A$ y AA^{T} nos conducen ambas a los valores singulares de A.

Si $m > n$, entonces AB tiene $m - n$ autovalores extra en comparación con BA.

Las submatrices tienen valores singulares más pequeños

El enfoque $||A|| = \sigma_1$ maximizando $||A\boldsymbol{x}||/||\boldsymbol{x}||$ hace fácil probar este hecho útil. La norma de una submatriz no puede ser mas grande que la norma de la matriz completa: $\boldsymbol{\sigma_1(B) \le \sigma_1(A)}$.

Si B consiste de $M \le m$ filas y $N \le n$ columnas de A, entonces $||B|| \le ||A||$. (22)

Prueba Tomemos vectores $\boldsymbol{y}$ con elementos no nulos solo en las N posiciones que corresponden a las columnas de B. Claramente, máximo $||B\boldsymbol{y}||/||\boldsymbol{y}|| \le$ máximo $||A\boldsymbol{x}||/||\boldsymbol{x}||$.

Reduzcamos aún mas $||B\boldsymbol{y}||$ mirando solamente las M componentes de $B\boldsymbol{y}$ que correspondan a las filas de B. Así eliminando columnas y filas no se puede aumentar la norma σ_1, y $||B|| \le ||A||$.

La SVD para Derivadas e Integrales

Este puede ser el ejemplo más claro de la SVD. No se comienza con una matriz (pero allí llegaremos). Historicamente, la primera SVD no fue para vectores sino para *funciones*. En este caso A no es una matriz sino un *operador*. Un ejemplo es el operador que integra cada función. Otro ejemplo es el operador (no acotado) D que deriva:

Operadores sobre funciones
Integral y derivada

$$A\boldsymbol{x}(s) = \int_0^s \boldsymbol{x}(t)\,dt \quad \text{y} \quad D\boldsymbol{x}(t) = \frac{d\boldsymbol{x}}{dt}. \tag{23}$$

Esos operadores son lineales (si no, el cálculo sería mucho mas difícil de lo que ya es). De alguna manera D es la inversa de A, por el Teorema Fundamental del Cálculo. Más precisamente, D es la inversa a izquierda con $\boldsymbol{DA = I}$: derivada de la integral es igual a la función original.

Pero $\boldsymbol{AD \ne I}$ ya que la derivada de una función constante es cero. Luego D tiene un espacio nulo, como si fuese una matriz con columnas dependientes. **¡ D es la seudoinversa o pseudoinversa de A!** Los senos y los cosenos son los $\boldsymbol{u}$ y $\boldsymbol{v}$ de los operadores $A =$ integral y $D =$ derivada:

$$\boxed{A\boldsymbol{v} = \sigma\boldsymbol{u} \text{ es } \boldsymbol{A(\cos kt) = \frac{1}{k}(\operatorname{sen} kt)} \qquad \text{Entonces } \boldsymbol{D(\operatorname{sen} kt) = k\,(\cos kt).}} \tag{24}$$

La simplicidad de esas ecuaciones es la razón para incluirlas en este libro. Estamos trabajando con *funciones periódicas*: $\boldsymbol{x}(t + 2\pi) = \boldsymbol{x}(t)$. El espacio de entrada de A contiene las *funciones pares* como $\cos t = \cos(-t)$. Las salidas de A (y las entradas de D) son las *funciones impares* como $\operatorname{sen} t = -\operatorname{sen}(-t)$. Esos espacios de entrada y de salida son como $\mathbf{R}^n$ y $\mathbf{R}^m$ para una matriz de m por n.

La propiedad especial de la SVD es que tanto los $\boldsymbol{v}$ como los $\boldsymbol{u}$ son ortogonales. Aquí esos vectores singulares se han convertido en funciones muy bonitas, *los cosenos son ortogonales entre sí como también lo son los senos. Sus productos internos son integrales iguales a cero*:

$$\boldsymbol{v}_k^{\mathrm{T}}\boldsymbol{v}_j = \int_0^{2\pi} (\cos kt)\,(\cos jt)\,dt = \mathbf{0} \quad \text{y} \quad \boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{u}_j = \int_0^{2\pi} (\operatorname{sen} kt)\,(\operatorname{sen} jt)\,dt = \mathbf{0}.$$

Notemos que el producto interno de las funciones $\boldsymbol{x}_1$ y $\boldsymbol{x}_2$ es la integral de $\boldsymbol{x}_1(t)\,\boldsymbol{x}_2(t)$. Esto es lo análogo en el espacio de funciones (*espacio de Hilbert*) al producto punto que suma $\boldsymbol{y}\cdot\boldsymbol{z} = \Sigma y_i z_i$. De hecho el símbolo $\int$ fue creado de alguna forma a partir de Σ (y las integrales son límites de sumas).

Diferencias finitas

La forma discreta de una derivada es una **diferencia finita**. La forma discreta de una integral es una **suma**. Aquí elegimos una matriz D de 4 por 3 que corresponde a la diferencia atrasada $f(x)-f(x-\Delta x)$:

$$\boldsymbol{D} = \begin{bmatrix} 1 & & \\ -1 & 1 & \\ & -1 & 1 \\ & & -1 \end{bmatrix} \quad \text{con} \quad \boldsymbol{D^{\mathrm{T}}} = \begin{bmatrix} 1 & -1 & & \\ & 1 & -1 & \\ & & 1 & -1 \end{bmatrix}. \tag{25}$$

Para hallar valores singulares y vectores singulares, calculemos $D^{\mathrm{T}}D$ (3 por 3) y DD^{T} (4 por 4):

$$\boldsymbol{D^{\mathrm{T}}D} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \quad \text{y} \quad \boldsymbol{DD^{\mathrm{T}}} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}. \tag{26}$$

¡Sus autovalores no nulos son siempre los mismos! DD^{T} también tiene un autovalor cero con autovector $\boldsymbol{u}_4 = \left(\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2}\right)$. Este es el equivalente discreto a la función $f(x) = \frac{1}{2}$ con $df/dx = 0$.

Los autovalores no nulos de ambas matrices simétricas $D^{\mathrm{T}}D$ y DD^{T} son

$$\boldsymbol{\lambda_1 = \sigma_1^2\,(D) = 2+\sqrt{2}} \qquad \boldsymbol{\lambda_2 = \sigma_2^2\,(D) = 2} \qquad \boldsymbol{\lambda_3 = \sigma_3^2\,(D) = 2-\sqrt{2}} \tag{27}$$

Los autovectores $\boldsymbol{v}$ de $D^{\mathrm{T}}D$ son los vectores singulares a derecha de D, son los **senos discretos**. Los autovectores $\boldsymbol{u}$ de DD^{T} son los vectores singulares a izquierda de D, son los **cosenos discretos**:

$$\sqrt{2}\,\boldsymbol{V} = \begin{bmatrix} \operatorname{sen}\frac{\pi}{4} & \operatorname{sen}\frac{2\pi}{4} & \operatorname{sen}\frac{3\pi}{4} \\ \operatorname{sen}\frac{2\pi}{4} & \operatorname{sen}\frac{4\pi}{4} & \operatorname{sen}\frac{6\pi}{4} \\ \operatorname{sen}\frac{3\pi}{4} & \operatorname{sen}\frac{6\pi}{4} & \operatorname{sen}\frac{9\pi}{4} \end{bmatrix} \qquad \sqrt{2}\,\boldsymbol{U} = \begin{bmatrix} \cos\frac{1}{2}\frac{\pi}{4} & \cos\frac{1}{2}\frac{2\pi}{4} & \cos\frac{1}{2}\frac{3\pi}{4} & 1 \\ \cos\frac{3}{2}\frac{\pi}{4} & \cos\frac{3}{2}\frac{2\pi}{4} & \cos\frac{3}{2}\frac{3\pi}{4} & 1 \\ \cos\frac{5}{2}\frac{\pi}{4} & \cos\frac{5}{2}\frac{2\pi}{4} & \cos\frac{5}{2}\frac{3\pi}{4} & 1 \\ \cos\frac{7}{2}\frac{\pi}{4} & \cos\frac{7}{2}\frac{2\pi}{4} & \cos\frac{7}{2}\frac{3\pi}{4} & 1 \end{bmatrix}.$$

Estas son las famosas matrices DST y DCT: **transformada discreta de senos** y **transformada discreta de cosenos**. La matriz DCT ha sido la columna vertebral de la compresión de imágenes JPEG. En realidad JPEG aumenta U a 8 por 8, lo que reduce el efecto "cuadriculado" de la imagen. Bloques de píxeles de 8 por 8 se transforman con una DCT bidimensional, que luego se comprimen y se transmiten. La ortogonalidad de esas matrices es la clave de la Sección IV.4.

Nuestro objetivo fue mostrar la forma discreta de la bella descomposición en valores singulares $D(\operatorname{sen} kt) = k(\cos kt)$. Podríamos decir acertadamente que esto es solo un ejemplo. Pero Fourier está siempre presente en las ecuaciones lineales con coeficientes constantes y siempre es importante.

En procesamiento de señales las letras claves son LTI: **Invariancia lineal con respecto al tiempo (Linear Time Invariance)**.

La descomposición polar $A = QS$

Todo número complejo $x + iy$ tiene la forma polar $re^{i\theta}$. Un número $r \geq 0$ multiplica un número $e^{i\theta}$ de la circunferencia unitaria. Tenemos $x + iy = r\cos\theta + ir\operatorname{sen}\theta = re^{i\theta}$. Pensemos estos números como matrices de 1 por 1. Entonces $e^{i\theta}$ es una *matriz ortogonal* Q y $r \geq 0$ es una *matriz semidefinida positiva* (llamémosla S). La ***descomposición polar*** extiende la misma idea a matrices de n por n: ortogonal por semidefinida positiva, $\boldsymbol{A = QS}$.

Toda matriz cuadrada real se puede factorizar como $\boldsymbol{A = QS}$, donde Q es ***ortogonal*** y S es ***simétrica semidefinida positiva***. Si A es invertible, S es definida positiva.

Descomposición polar $\quad \boldsymbol{A = U\Sigma V^{\mathrm{T}} = (UV^{\mathrm{T}})(V\Sigma V^{\mathrm{T}}) = (Q)(S).}$ (28)

El primer factor UV^{T} es Q. El producto de matrices ortogonales es ortogonal. El segundo factor $V\Sigma V^{\mathrm{T}}$ es S. Es semidefinido positivo ya que sus autovalores están en Σ.

Si A es invertible, entonces Σ y S también son invertibles. ***S es la raíz cuadrada simétrica definida positiva*** de $A^{\mathrm{T}}A$, ya que $S^2 = V\Sigma^2V^{\mathrm{T}} = A^{\mathrm{T}}A$. Así los autovalores de S son los valores singulares de A. Los autovectores de S son los vectores singulares $\boldsymbol{v}$ de A.

También existe una descomposición polar $A = KQ$ en el orden inverso. Q es la misma pero ahora $K = U\Sigma U^{\mathrm{T}}$. Entonces K es la raíz cuadrada simétrica definida positiva de AA^{T}.

Ejemplo Encuentre Q y S (rotación y estiramiento) en la descomposición polar de $A = \begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix}$.

Solución Las matrices U y Σ y V fueron encontradas antes de la ecuación (12):

$$Q = UV^{\mathrm{T}} = \frac{1}{\sqrt{20}}\begin{bmatrix} 1 & -3 \\ 3 & 1 \end{bmatrix}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \frac{1}{\sqrt{20}}\begin{bmatrix} 4 & -2 \\ 2 & 4 \end{bmatrix} = \frac{1}{\sqrt{5}}\begin{bmatrix} \mathbf{2} & \mathbf{-1} \\ \mathbf{1} & \mathbf{2} \end{bmatrix}$$

$$S = V\Sigma V^{\mathrm{T}} = \frac{\sqrt{5}}{2}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 3 & \\ & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} = \sqrt{5}\begin{bmatrix} \mathbf{2} & \mathbf{1} \\ \mathbf{1} & \mathbf{2} \end{bmatrix}. \text{ Entonces } \boldsymbol{A = QS}.$$

En mecánica, la descomposición polar separa la ***rotación*** (en Q) del ***estiramiento***. Los autovalores de S dan los factores del estiramiento en la Figura I.10. Los autovectores de S dan las direcciones de estiramiento (los ejes principales de la elipse). En la Sección IV.9 que trata sobre el problema ortogonal de Procrustes se afirma que **Q es la matriz ortogonal más cercana a la matriz A.**

La factorización $\boldsymbol{A = UTV^{\mathrm{T}}}$ con T matriz triangular es parecida a la SVD. Es más rápida de calcular. Martinsson describe el desarrollo de G. W. Stewart y propone un algoritmo aleatorizado de aproximación por matrices de rango bajo usando bloques (arXiv : 1703.00998).

Problemas de la Sección I.8

1 Una matriz simétrica $S = S^{\mathrm{T}}$ tiene autovectores ortonomales $\boldsymbol{v}_1$ a $\boldsymbol{v}_n$. Entonces cualquier vector $\boldsymbol{x}$ se puede escribir como una combinación de ellos: $\boldsymbol{x} = c_1\boldsymbol{v}_1 + \cdots + c_n\boldsymbol{v}_n$. Explique estas dos fórmulas :

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = c_1^2 + \cdots + c_n^2 \qquad\qquad \boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x} = \lambda_1 c_1^2 + \cdots + \lambda_n c_n^2.$$

2 El Problema 1 da una forma clara del cociente de Rayleigh $\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}/\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$:

$$\boxed{R(\boldsymbol{x}) = \frac{\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}} = \frac{\lambda_1 c_1^2 + \cdots + \lambda_n c_n^2}{c_1^2 + \cdots + c_n^2}.}$$

¿Por qué el valor máximo de este cociente es igual al autovalor más grande $\boldsymbol{\lambda_1}$? Esta debe ser la forma más simple de entender la "segunda construcción" de la SVD en la ecuación (15). Puede verse por qué el cociente $R(\boldsymbol{x})$ es máximo cuando $c_1 = 1$ y $c_2 = c_3 = \cdots = c_n = 0$.

3 Luego viene λ_2 cuando $\boldsymbol{x} = \boldsymbol{v}_2$. Maximizamos $R(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}/\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$ con la condición de que $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{v}_1 = 0$. ¿ *Que significa esta condición para* c_1 ? ¿ Por qué el cociente del Problema 2 ahora se maximiza cuando $c_2 = 1$ and $c_1 = c_3 = \cdots = c_n = 0$?

4 Siguiendo al Problema 3, ¿ Qué problema de máximo resuelve $\boldsymbol{x} = \boldsymbol{v}_3$? Los mejores $\boldsymbol{c}$ son $c_3 = 1$ y $c_1 = c_2 = c_4 \cdots = 0$.

¿ Sujeto a qué dos condiciones en $\boldsymbol{x}$ es el máximo de $R(\boldsymbol{x}) = \dfrac{\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}$ igual a $\boldsymbol{\lambda_3}$?

5 Muestre que A^{T} tiene los mismos valores singulares (no nulos) que A. Entonces $||A|| = ||A^{\mathrm{T}}||$ para todas las matrices. Pero no es cierto que para todos los vectores $||A\boldsymbol{x}|| = ||A^{\mathrm{T}}\boldsymbol{x}||$. Esto necesita $A^{\mathrm{T}}A = AA^{\mathrm{T}}$.

6 Encuentre los σ, los $\boldsymbol{v}$ y los $\boldsymbol{u}$ en la SVD para $A = \begin{bmatrix} 3 & 4 \\ 0 & 5 \end{bmatrix}$. Use la ecuación (12).

7 ¿ Cuál es la norma $||A - \sigma_1 \boldsymbol{u}_1\boldsymbol{v}_1^{\mathrm{T}}||$ cuando se le quita la matriz más grande de rango uno a A ? ¿ Cuáles son todos los valores singulares de esta matriz reducida ? ¿ Y su rango ?

8 Encuentre los σ y $\boldsymbol{v}$ y $\boldsymbol{u}$, y verifique que $A = \begin{bmatrix} 0 & 2 & 0 \\ 0 & 0 & 3 \\ 0 & 0 & 0 \end{bmatrix} = U\Sigma V^{\mathrm{T}}$. Para esta matriz, las matrices ortogonales U y V son matrices de permutación.

9 Para maximizar $\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}$ con $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = 1$, Lagrange trabajaría con $L = \boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x} + \lambda(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} - 1)$. Muestre que $\boldsymbol{\nabla L} = (\partial L/\partial x_1, \ldots, \partial L/\partial x_n) = 0$ es exactamente $2S\boldsymbol{x} = 2\lambda\boldsymbol{x}$. Nuevamente máx $R(\boldsymbol{x}) = \lambda_1$.

10 Pruebe que $||B|| \leq ||A||$ en la ecuación (22) con un enfoque ligeramente diferente. Elimine primero las $N - n$ columnas de A. La nueva matriz tiene $||C|| \leq ||A||$. (¿Por qué?). Luego trasponga C: no cambia su norma. Finalmente elimine $M - m$ columnas de C^{T} para generar B^{T} que no crece en norma. Todo junto $||B|| = ||B^{\mathrm{T}}|| \leq ||C^{\mathrm{T}}|| = ||C|| \leq ||A||$.

11 Corrobore que la traza de $S = \begin{bmatrix} 0 & A \\ A^{\mathrm{T}} & 0 \end{bmatrix}$ sumando los elementos de la diagonal, sea igual a la suma de sus autovalores que están en la ecuación (21). Si A es una matriz diagonal cuadrada con elementos $1, 2, \ldots, n$, ¿Cuáles son los $2n$ autovalores y autovectores de S?

12 Encuentre la SVD de la matriz de rango uno $A = \begin{bmatrix} 2 & 4 \\ 1 & 2 \end{bmatrix}$. Factorice $A^{\mathrm{T}}A$ como $Q\Lambda Q^{\mathrm{T}}$.

13 He aquí mi prueba casera de la SVD. El Paso **2** usa la factorización $A^{\mathrm{T}}A = V\Lambda V^{\mathrm{T}}$ y $AA^{\mathrm{T}} = U\Lambda U^{\mathrm{T}}$ (con los mismos autovalores en Λ).

1	$A(A^{\mathrm{T}}A)$	$= (AA^{\mathrm{T}})A$
2	$AV\Lambda V^{\mathrm{T}}$	$= U\Lambda U^{\mathrm{T}}A$
3	$(U^{\mathrm{T}}AV)\Lambda$	$= \Lambda(U^{\mathrm{T}}AV)$
4	$U^{\mathrm{T}}AV$ debe ser diagonal	

El Paso 3 multiplica el Paso 2 a la izquierda por _____ y a la derecha por _____. Entonces la matriz $U^{\mathrm{T}}AV$ commuta con la matriz diagonal Λ en el Paso 3. ¿Por qué esto implica que $U^{\mathrm{T}}AV = \Sigma$ debe ser una matriz diagonal? Pruebe el caso de 3 por 3.

$$\boldsymbol{\Sigma\Lambda} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{bmatrix} = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{bmatrix} \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} = \boldsymbol{\Lambda\Sigma}$$

Compare las primeras filas. ¿Cuándo podemos concluir que $\sigma_{12} = 0$ y $\sigma_{13} = 0$? Esta es la limitación de mi prueba: *necesita que los autovalores de* $A^{\mathrm{T}}A$ *sean* _____.

El mismo problema aparece en las pruebas simples del teorema espectral $S = Q\Lambda Q^{\mathrm{T}}$. Es fácil cuando S no tiene λ repetidos. La SVD es fácil cuando A no tiene σ repetidos.

Ambas factorizaciones $S = Q\Lambda Q^{\mathrm{T}}$ y $A = U\Sigma V^{\mathrm{T}}$ siguen siendo válidas si λ o σ aparecen repetidos. El problema es que esto produce un plano completo de autovectores o vectores singulares. Hay que elegir específicamente los vectores singulares $\boldsymbol{u}$ iguales a $A\boldsymbol{v}/\sigma$, que es la prueba en la ecuación (9).

14 La Figura I.10 muestra cómo una matriz de 2 por 2 con elementos a, b, c, d produce una SVD con cuatro parámetros $\theta, \phi, \sigma_1, \sigma_2$. Tome ahora $A = U\Sigma V^{\mathrm{T}} = 2$ por 3 con 6 elementos.

¿Cuántos σ tiene una matriz de 2 por 3? Luego U (2 por 2) sólo necesita un ángulo. Para recuperar A, ¿cuántos ángulos deja para la matriz ortogonal V de 3 por 3?

El espacio fila de A es un plano en $\mathbf{R}^3$.Necesita _____ ángulos para la posición de ese plano . Necesita _____ ángulo en el plano para encontrar $\boldsymbol{v}_1$ y $\boldsymbol{v}_2$. Un total de _____ ángulos para V.

15 Cada matriz de 3 por 3 tiene 9 elementos. Entonces $U\Sigma V^{\mathrm{T}}$ debe tener 9 parámetros. ¿ Cuántos parámetros hay en U, Σ y V ? *Responda las mismas preguntas para una de* 4 *por* 4. ¿Cuántos parámetros describen una rotación en el espacio de dimensión 4 ?

16 _____ números definirán la dirección de un vector unitario $\boldsymbol{v}_1$ en $\mathbf{R}^5$. Luego la dirección de un vector unitario ortogonal $\boldsymbol{v}_2$ necesita _____ números. ¿Cuántos para $\boldsymbol{v}_3$, $\boldsymbol{v}_4$, $\boldsymbol{v}_5$? Total _____.

17 **Si v es un autovector de $A^{\mathrm{T}}A$ con $\lambda \neq 0$, entonces _____ es un autovector de AA^{T}.**

18 Si $A = U\Sigma V^{\mathrm{T}}$ es cuadrada e invertible, entonces $A^{-1} =$ _____. Encuentre todos los valores singulares de $A^{\mathrm{T}}A$ (no de A).

19 Si $A = A^{\mathrm{T}}$ tiene columnas ortogonales $\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3$ en $\mathbf{R}^3$ de magnitudes $2, 3, 4$, encuentre su SVD.

20 Las razones para el éxito de autovalores y autovectores es que $\boldsymbol{A^k = X\Lambda^k X^{-1}}$:

a) Los autovalores de A^k son $\lambda_1^k, \ldots, \lambda_n^k$

b) Un autovector $\boldsymbol{x}$ de A es también autovector de A^k.

Muestre que a) y b) son falsas para los valores y los vectores singulares de $\begin{bmatrix} -2 & -6 \\ 6 & 2 \end{bmatrix}$.

21 Muestre que los valores singulares de $AA^{\mathrm{T}}A$ son $(\sigma_1)^3$ a $(\sigma_r)^3$.

22 La ecuación (5) es $AV_r = U_r\Sigma_r$. Mutiplique por V_r^{T} para obtener $A = U_r\Sigma_r V_r^{\mathrm{T}}$ (**SVD reducida**).

Para ese paso no podemos usar $V_r V_r^{\mathrm{T}} = I$ (que es falso cuando $m > r$). Muestre en cambio que la matriz $A = U_r\Sigma_r V_r^{\mathrm{T}}$ satisface la ecuación (1).

23 Muestre que una matriz de m por n de rango r tiene $r(m + n - r)$ parámetros libres en su SVD: $A = U\Sigma V^{\mathrm{T}} = (m \times r)\,(r \times r)\,(r \times n)$. ¿ Por qué r vectores ortonormales $\boldsymbol{u}_1$ a $\boldsymbol{u}_r$ tienen $(m-1) + (m-2) + \cdots + (m-r)$ parámetros ?

Otro enfoque utiliza $A = CR = (m \times r)\,(r \times n)$ de la Sección I.1. La matriz R contiene una matriz identidad de r por r, eliminando r^2 parámetros de los $rm + rn$. Este conteo se repite en un apéndice de este libro.

I.9. Componentes principales y la matriz más cercana de rango bajo

Los componentes principales de A son sus vectores singulares, las columnas $\boldsymbol{u}_j$ y $\boldsymbol{v}_j$ de las matrices ortogonales U y V. El **Análisis de los Componentes Principales** (**PCA**, por su sigla en inglés) usa los mayores σ conectados a los primeros $\boldsymbol{u}$ y $\boldsymbol{v}$ para sintetizar la información en una matriz de datos. Dada una matriz A, extraemos la parte A_k más importante (**los σ más grandes**):

$$\boldsymbol{A_k = \sigma_1 u_1 v_1^{\mathrm{T}} + \cdots + \sigma_k u_k v_k^{\mathrm{T}} \quad \text{de rango } (A_k) = k.}$$

A_k resuelve un problema de optimización de matrices. **La matriz de rango k más cercana a A es A_k.** En estadística identificamos las partes de A con mayor varianza. Esto ubica a la SVD en el centro de la ciencia de datos.

En ese mundo, el PCA es aprendizaje "no supervisado". El único instructor es el álgebra lineal: la SVD nos indica que elijamos A_k. Cuando el aprendizaje es supervisado, tenemos un gran conjunto de datos de entrenamiento. El aprendizaje profundo (Sección VII.1) construye una función F (¡no lineal!) que clasifica correctamente la mayoría de los datos de entrenamiento. Luego aplicamos F a nuevos datos, como veremos.

El análisis de los componentes principales está basado en la aproximación matricial por A_k. La prueba de que A_k es la mejor opción comenzó con Schmidt (1907). Su teorema fue escrito para operadores A en espacios de funciones, y se extiende directamente a matrices en espacios vectoriales. Eckart y Young presentaron una prueba nueva en 1936 (usando la norma de Frobenius para matrices). Luego Mirsky encontró una prueba más general en 1955 que permite cualquier norma $||A||$ que dependa solo de los valores singulares, como en las definiciones (2), (3), y (4) a continuación.

He aquí la propiedad clave de la matriz de rango k especial $A_k = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^{\mathrm{T}} + \cdots + \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^{\mathrm{T}}$:

Eckart-Young **Si B tiene rango k entonces $||A - B|| \geq ||A - A_k||$.** (1)

Tres opciones para la norma de matrices $||A||$ tienen especial importancia y sus propios nombres:

Norma espectral $\boldsymbol{||A||_2 = \max \dfrac{||Ax||}{||x||} = \sigma_1}$ (a menudo llamada **norma ℓ^2**) (2)

Norma de Frobenius $\boldsymbol{||A||_F = \sqrt{\sigma_1^2 + \cdots + \sigma_r^2}}$ (12) y (13) también definen $||A||_F$ (3)

Norma nuclear $\boldsymbol{||A||_N = \sigma_1 + \sigma_2 + \cdots + \sigma_r}$ (la traza) (4)

Estas normas tienen distintos valores aún para la matriz identidad de n por n:

$$||I||_2 = 1 \qquad ||I||_F = \sqrt{n} \qquad ||I||_N = n. \tag{5}$$

Reemplazando I por cualquier matriz ortogonal Q las normas se mantienen iguales (porque todo $\sigma_i = 1$):

$$||Q||_2 = 1 \qquad ||Q||_F = \sqrt{n} \qquad ||Q||_N = n. \tag{6}$$

Además, la norma espectral, la de Frobenius y la nuclear de cualquier matriz se mantienen iguales cuando A se multiplica (sin importar el lado) por una matriz ortogonal.

Los valores singulares no cambian cuando U y V cambian a Q_1U y Q_2V. Para matrices complejas la palabra *unitaria* reemplaza a *ortogonal*. Entonces $\overline{\boldsymbol{Q}}^{\mathbf{T}}\boldsymbol{Q} = \boldsymbol{I}$. Estas tres normas son **unitariamente invariantes**: $||Q_1 A \overline{Q}_2^{\mathrm{T}}|| = ||A||$. La prueba de Mirsky del teorema de Eckart-Young en (1) se aplica a todas las normas unitariamente invariantes: $||A||$ se calcula a partir de Σ.

$$\textbf{Las tres normas verifican} \quad \boldsymbol{||Q_1 A Q_2^{\mathrm{T}}|| = ||A||} \quad \textbf{para } \boldsymbol{Q_1} \textbf{ y } \boldsymbol{Q_2} \textbf{ ortogonales} \tag{7}$$

Ahora presentamos pruebas más simples de (1) para la norma L^2 y la norma de Frobenius.

Teorema de Eckart-Young: la mejor aproximación por A_k

Antes de enfocarnos en la prueba es útil ver lo que el teorema nos dice. En este ejemplo A es diagonal y $k = 2$:

$$\text{La matriz de rango dos más cercana a} \quad \boldsymbol{A} = \begin{bmatrix} \mathbf{4} & 0 & 0 & 0 \\ 0 & \mathbf{3} & 0 & 0 \\ 0 & 0 & \mathbf{2} & 0 \\ 0 & 0 & 0 & \mathbf{1} \end{bmatrix} \quad \text{es} \quad \boldsymbol{A_2} = \begin{bmatrix} \mathbf{4} & 0 & 0 & 0 \\ 0 & \mathbf{3} & 0 & 0 \\ 0 & 0 & \mathbf{0} & 0 \\ 0 & 0 & 0 & \mathbf{0} \end{bmatrix}.$$

¡ Esto debe ser cierto ! Se puede decir que esta matriz diagonal es demasiado simple y no es típica. Pero la norma L^2 y la norma de Frobenius no cambian cuando la matriz A se convierte en Q_1AQ_2 (para cualesquiera Q_1 y Q_2 ortogonales). Así, este ejemplo incluye *cualquier* matriz de 4 por 4 con valores singulares $4, 3, 2, 1$. El teorema de Eckart-Young nos sugiere mantener 4 y 3 porque son los más grandes. El error en L^2 es $||A - A_2||_2 = 2$. La norma de Frobenius es $\boldsymbol{||A - A_2||_F = \sqrt{5}}$.

La parte incómoda del problema es "las matrices de rango dos". *Ese conjunto no es convexo.* El promedio de A_2 y B_2 (ambas de rango 2) puede fácilmente tener rango 4. ¿ Es posible que esta B_2 esté más cerca de A que A_2 ?

$$\begin{array}{l} \text{¿ Podrá esta } B_2 \text{ ser} \\ \text{ una mejor aproximación} \\ \text{de rango 2 de } A \text{ ?} \end{array} \quad \begin{bmatrix} 3.5 & 3.5 & & \\ 3.5 & 3.5 & & \\ & & 1.5 & 1.5 \\ & & 1.5 & 1.5 \end{bmatrix}.$$

Los errores $A - B_2$ son solo 0.5 en la diagonal principal donde $A - A_2$ tiene errores 2 y 1. Por supuesto los errores 3.5 y 1.5 fuera de la diagonal serán muy grandes. ¿ Pero tal vez existe otra opción mejor que A_2 ?

No, A_2 es la mejor con rango $k = 2$. Lo demostramos para la norma L^2 y luego para Frobenius.

$$\boxed{\textbf{Eckart-Young en } \boldsymbol{L^2} \qquad \text{Si rango}(B) \le k \text{ entonces } \boldsymbol{||A - B||} = \max \frac{||(A-B)\,\boldsymbol{x}||}{||\boldsymbol{x}||} \ge \boldsymbol{\sigma_{k+1}}.} \tag{8}$$

Sabemos que $||A - A_k|| = \sigma_{k+1}$. Toda la prueba de $||A - B|| \geq \sigma_{k+1}$ depende de una buena elección del vector $\boldsymbol{x}$ al calcular la norma $||A - B||$:

$$\text{Elija } \boldsymbol{x} \neq \mathbf{0} \text{ de tal manera que } B\boldsymbol{x} = \mathbf{0} \text{ y } \boldsymbol{x} = \sum_{1}^{k+1} c_i \boldsymbol{v}_i \tag{9}$$

Primero, el espacio nulo de B tiene dimensión $\geq n - k$, porque B tiene rango $\leq k$. Segundo, las combinaciones de $\boldsymbol{v}_1$ a $\boldsymbol{v}_{k+1}$ producen un subespacio de dimensión $k + 1$. Esos dos subespacios deben intersectarse. Como las dimensiones suman $(n-k)+(k+1) = n+1$, los subespacios deben compartir (al menos) una recta. Pensemos en dos planos que atraviesen $(0, 0, 0)$ en $\mathbf{R}^3$: comparten una recta ya que $2 + 2 > 3$. Elijamos un vector distinto de cero $\boldsymbol{x}$ en esta recta.

Usemos ese $\boldsymbol{x}$ para estimar la norma de $A - B$ en (8). Recuerde que $B\boldsymbol{x} = \mathbf{0}$ y $A\boldsymbol{v}_i = \sigma_i \boldsymbol{u}_i$:

$$||(A - B)\,\boldsymbol{x}||^2 = ||A\boldsymbol{x}||^2 = ||\sum c_i \sigma_i \boldsymbol{u}_i||^2 = \sum_{1}^{k+1} c_i^2 \sigma_i^2. \tag{10}$$

La suma es al menos tan grande como $\left(\sum c_i^2\right) \sigma_{k+1}^2$, que es exactamente $||\boldsymbol{x}||^2 \sigma_{k+1}^2$. La ecuación (10) prueba que $||(A - B)\boldsymbol{x}|| \geq \sigma_{k+1} ||\boldsymbol{x}||$. Este vector $\boldsymbol{x}$ nos da la cota inferior que queremos para $||A - B||$:

$$\boxed{\frac{||(A - B)\,\boldsymbol{x}||}{||\boldsymbol{x}||} \geq \sigma_{k+1} \text{ significa que } \boldsymbol{||A - B|| \geq \sigma_{k+1} = ||A - A_k||.} \textbf{ ¡Demostrado!}} \tag{11}$$

La norma de Frobenius

Ahora nos referiremos a la norma de Frobenius, para mostrar que A_k también es la mejor aproximación para esta norma. Es útil ver tres fórmulas diferentes para esta norma. La primera fórmula considera a A como un vector largo, y toma la norma ℓ^2 usual de ese vector. La segunda fórmula reconoce que en la diagonal principal de $A^{\mathrm{T}}A$ se encuentra la norma ℓ^2 (al cuadrado) de cada columna de A.

Por ejemplo, el elemento $1, 1$ de $A^{\mathrm{T}}A$ es la suma $|a_{11}|^2 + \cdots + |a_{m1}|^2$ de la columna 1 de A. Así que (12) es lo mismo que (13), solo estamos tomando los números $|a_{ij}|^2$ de a una columna por vez.

La fórmula (14) para la norma de Frobenius usa los autovalores σ_i^2 de $A^{\mathrm{T}}A$ (la traza es siempre la suma de los autovalores). La fórmula (14) también se obtiene directamente de la SVD : la norma de Frobenius de $A = U\Sigma V^{\mathrm{T}}$ no se ve afectada por U y V, así $\boldsymbol{||A||_F^2 = ||\Sigma||_F^2}$. Esto es $\boldsymbol{\sigma_1^2 + \cdots + \sigma_r^2}$.

$$\boxed{\begin{aligned} &\boldsymbol{||A||_F^2} = |a_{11}|^2 + |a_{12}|^2 + \cdots + |a_{mn}|^2 \quad \text{(todos los } a_{ij}^2\text{)} && (12)\\ &\boldsymbol{||A||_F^2} = \text{traza de } A^{\mathrm{T}}A = (A^{\mathrm{T}}A)_{11} + \cdots + (A^{\mathrm{T}}A)_{nn} && (13)\\ &\boldsymbol{||A||_F^2} = \boldsymbol{\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_r^2} && (14) \end{aligned}}$$

Eckart-Young en la norma de Frobenius

Para la norma $||A–B||_F$, Pete Stewart encontró y generosamente nos compartió esta elegante prueba.

Suponga que la matriz B de rango $\leq k$ es la más cercana a A. Queremos demostrar que $B = A_k$. La sorpresa es que empezamos con la descomposición en valores singulares de B (*no* de A):

$$\boldsymbol{B} = U \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} V^{\mathrm{T}} \text{ donde la matriz diagonal } D \text{ es de } k \text{ por } k. \tag{15}$$

Esas matrices ortogonales U y V de B no necesariamente diagonalizarán a A:

$$\boldsymbol{A} = U \begin{bmatrix} L+E+R & F \\ G & H \end{bmatrix} V^{\mathrm{T}} \tag{16}$$

Aquí L es una matriz estrictamente triangular inferior en las primeras k filas, E es diagonal y R es una matriz estrictamente triangular superior. El paso 1 mostrará que L, R, y F son nulas al comparar A y B con esta matriz C que claramente tiene rango $\leq k$:

$$\boldsymbol{C} = U \begin{bmatrix} L+D+R & F \\ 0 & 0 \end{bmatrix} V^{\mathrm{T}} \tag{17}$$

Esta es la idea clave de Stewart, construir C con filas de ceros para mostrar su rango. Esas matrices ortogonales U y V^{T} no alteran la norma de Frobenius. Elevamos al cuadrado todos los elementos de la matriz y sumamos, teniendo en cuenta que $A - C$ tiene ceros donde $A - B$ tiene las matrices L, R, F:

$$||\boldsymbol{A - B}||_F^2 = ||\boldsymbol{A - C}||_F^2 + ||\boldsymbol{L}||_F^2 + ||\boldsymbol{R}||_F^2 + ||\boldsymbol{F}||_F^2. \tag{18}$$

Como $||A - B||_F^2$ era la más pequeña posible concluimos que L, R, F son cero. De manera similar encontramos que $G = 0$. Hasta aquí sabemos que $U^{\mathrm{T}}AV$ tiene dos bloques y E es diagonal (como D):

$$\boldsymbol{U^{\mathrm{T}}AV} = \begin{bmatrix} \boldsymbol{E} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H} \end{bmatrix} \quad \text{y} \quad \boldsymbol{U^{\mathrm{T}}BV} = \begin{bmatrix} \boldsymbol{D} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}.$$

Si B es la más cercana a A entonces $U^{\mathrm{T}}BV$ es la más cercana a $U^{\mathrm{T}}AV$. Ahora vemos la luz.

La matriz D debe ser la misma que $E = \operatorname{diag}(\sigma_1, \ldots, \sigma_k)$.

Los valores singulares de H deben ser los $n - k$ valores singulares más pequeños de A.

El menor error $||A - B||_F$ debe ser $||H||_F = \sqrt{\sigma_{k+1}^2 + \cdots + \sigma_r^2}$ = **Eckart-Young**.

En el ejemplo de 4 por 4 al principio de esta sección, A_2 es la mejor posible: $||A - A_2||_F = \sqrt{5}$. Es excepcional tener esta solución explícita A_k para un problema de optimización no convexa.

Minimizar la distancia de Frobenius $||A - B||_F^2$

He aquí un método distinto y más directo para demostrar Eckart-Young: *iguale las derivadas de* $||A - B||_F^2$ *a cero*. Toda matriz de rango k se factoriza en $\boldsymbol{B = CR} = (m \times k)\,(k \times n)$. Por la SVD, podemos pedir que C tenga k columnas ortogonales (así $C^{\mathrm{T}}C =$ matriz diagonal D) y que R tenga k filas ortonormales (así $RR^{\mathrm{T}} = I$). Buscamos obtener $C = U_k\Sigma_k$ y $R = V_k^{\mathrm{T}}$. **Derivemos $\boldsymbol{E = ||A - CR||_F^2}$ para encontrar las matrices $\boldsymbol{C}$ y $\boldsymbol{R}$ que minimizan $\boldsymbol{E}$:**

$$\frac{\partial E}{\partial C} = 2(CR - A)R^{\mathrm{T}} = 0 \qquad \left(\frac{\partial E}{\partial R}\right)^{\mathrm{T}} = 2(R^{\mathrm{T}}C^{\mathrm{T}} - A^{\mathrm{T}})\,C = 0 \tag{19}$$

¡ Vea la página web del libro de Dan Drucker para ver los cálculos ! La primera da $AR^{\mathrm{T}} = CRR^{\mathrm{T}} = C$. La segunda da $\boldsymbol{R^{\mathrm{T}}D} = A^{\mathrm{T}}C = \boldsymbol{A^{\mathrm{T}}AR^{\mathrm{T}}}$. Como D es diagonal, esto significa que:

Las columnas de R^{T} *son autovectores de* $A^{\mathrm{T}}A$. *Son vectores singulares a derecha* $\boldsymbol{v}_j$ *de* A. Análogamente las columnas de C son autovectores de AA^{T}: $\boldsymbol{AA^{\mathrm{T}}C} = AR^{\mathrm{T}}D = \boldsymbol{CD}$. Entonces C contiene los vectores singulares a izquierda $\boldsymbol{u}_j$. ¿ Qué vectores singulares minimizan el error E ?

E es la suma de todos los σ^2 que *no estaban involucrados en* C *y* R. Para minimizar, esos deberían ser los menores valores singulares de A. Eso deja los mayores valores singulares para producir el mejor $B = CR = A_k$, con $||A - CR||_F^2 = \sigma_{k+1}^2 + \cdots + \sigma_r^2$. Esta elegante prueba está en la tesis doctoral del MIT de Nathan Srebro: **ttic.uchicago.edu/~nati/Publications/thesis.pdf**

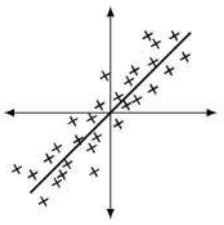
Análisis de componentes principales

Ahora comenzaremos a usar la SVD. *La matriz* A *está llena de datos*. Tenemos n muestras. Para cada muestra medimos m variables (como altura y peso). La matriz de datos A_0 tiene n columnas y m filas. En muchas aplicaciones esta es una matriz muy grande.

El primer paso es encontrar el promedio (la media muestral) de cada fila de A_0. Restamos la media a las m entradas de esa fila. Ahora cada fila de la matriz centrada A tiene *media cero*. Las columnas de A son n puntos en $\mathbf{R}^m$. Debido al centrado, la suma de los n vectores columna es cero. Entonces el promedio de las columnas es el vector nulo.

A menudo esos n puntos están agrupados cerca de una recta o un plano u otro subespacio de baja dimensión de $\mathbf{R}^m$. La Figura I.11 muestra un conjunto típico de puntos muestrales agrupados alrededor de una recta en $\mathbf{R}^2$ (después de centrar A_0 para mover los puntos de izquierda a derecha y de arriba a abajo para obtener media $(0, 0)$ en A).

¿ Cómo hará el álgebra lineal para encontrar la línea más cercana que pase por $(0, 0)$? **Esa línea está alineada con el primer vector singular $\boldsymbol{u_1}$ de $\boldsymbol{A}$.** ¡ Este es el punto clave del PCA !



A es de $\mathbf{2} \times \boldsymbol{n}$ (espacio nulo grande)

AA^{T} es de $\mathbf{2} \times \mathbf{2}$ (matriz pequeña)

$A^{\mathrm{T}}A$ es de $\boldsymbol{n} \times \boldsymbol{n}$ (matriz grande)

Dos valores singulares $\boldsymbol{\sigma_1 > \sigma_2 > 0}$

Figura I.11: puntos muestrales (columnas de A) generalmente están cerca de una recta en $\mathbf{R}^2$ o de un subespacio en $\mathbf{R}^m$.

Permítanme expresar el problema (que la SVD resuelve) primero en términos estadísticos y luego en términos geométricos. Después de eso vendrán el álgebra lineal y los ejemplos.

La estadística detrás del PCA

Los números clave en probabilidad y estadística son la **media** y la **varianza**. La "media" es un promedio de los datos (en cada fila de A_0). Restar las medias de cada fila de A_0 produce una A centrada. Las cantidades cruciales son las "varianzas" y "covarianzas". Las varianzas son las sumas de los cuadrados de las distancias a la media en cada fila de A.

Las varianzas son los elementos diagonales de la matriz $\boldsymbol{AA^{\mathrm{T}}}$.

Suponga que las columnas de A corresponden a la edad de un niño en el eje x y su altura en el eje y (esas edades y alturas indican la diferencia con la edad y altura promedio). Buscamos la recta que pase lo más cerca posible de los puntos muestrales en la figura. Además tenemos que tener en cuenta de *la distribución conjunta de edad-altura* de los datos.

Las covarianzas son los elementos de afuera de la diagonal de la matriz $\boldsymbol{AA^{\mathrm{T}}}$.

Esos son los productos punto (fila i de A) $\cdot$ (fila j de A). Una covarianza alta significa que la altura aumenta con la edad (la covarianza negativa indica que una variable aumenta cuando la otra disminuye). Nuestro ejemplo tiene solo dos filas: una para la edad y otra para la altura: la matriz simétrica AA^{T} es de 2 por 2. Al aumentar el número n de niños en la muestra, dividimos por $n-1$ para que la escala de AA^{T} sea estadísticamente correcta.

La matriz de covarianza muestral se define como $\boldsymbol{S = \dfrac{AA^{\mathrm{T}}}{n-1}}$.

El factor es $n-1$ porque un grado de libertad ya ha sido utilizado para media $= 0$. He aquí un ejemplo con seis edades y alturas ya centrado para que cada fila sume cero:

$$A = \begin{bmatrix} 3 & -4 & 7 & 1 & -4 & -3 \\ 7 & -6 & 8 & -1 & -1 & -7 \end{bmatrix}$$

Para estos datos, la matriz de covarianza muestral S se calcula fácilmente. Es definida positiva.

Varianzas y covarianzas $$S = \frac{1}{6-1} AA^{\mathrm{T}} = \begin{bmatrix} 20 & 25 \\ 25 & 40 \end{bmatrix}.$$

Los dos autovectores ortogonales de S son $\boldsymbol{u_1}$ y $\boldsymbol{u_2}$. Son vectores singulares a izquierda (*componentes principales*) de A. El teorema de Eckart-Young establece que **el vector $\boldsymbol{u_1}$ apunta hacia la recta más cercana en la Figura I.11. Los autovectores de $\boldsymbol{S}$ son vectores singulares de $\boldsymbol{A}$.**

El segundo vector singular $\boldsymbol{u_2}$ será perpendicular a la recta más cercana.

Nota importante El PCA puede describirse usando la matriz simétrica $S = AA^{\mathrm{T}}/(n-1)$ o la rectangular A. Sin duda S es una buena matriz. Pero dados los datos en A, calcular S sería un error computacional. Para matrices grandes, una SVD directa de A es más rápida y más precisa.

En el ejemplo, S tiene autovalores cercanos a 57 y 3. Su suma es $20 + 40 = 60$, la traza de S. La primera matriz de rango uno $\sqrt{57}\boldsymbol{u}_1\boldsymbol{v}_1^{\mathrm{T}}$ es mucho mayor que la segunda matriz $\sqrt{3}\boldsymbol{u}_2\boldsymbol{v}_2^{\mathrm{T}}$. **El autovector principal $\boldsymbol{u}_1 \approx (0.6, 0.8)$ nos dice que la recta más cercana en el gráfico de puntos tiene una pendiente cerca a $\mathbf{8/6}$.** La dirección en el gráfico casi produce un triángulo rectángulo $6 - 8 - 10$.

Ahora pasaré del álgebra del PCA a la geometría. ¿ En qué sentido la recta en la dirección de $\boldsymbol{u}_1$ será la *recta más cercana* a los datos centrados ?

La geometría detrás del PCA

La mejor recta en la Figura I.11 resuelve un problema de **mínimos cuadrados perpendiculares**. Esto también se llama *regresión ortogonal*. Es diferente del método de cuadrados mínimos estándar para n puntos de datos, o de la solución por cuadrados mínimos de un sistema lineal $A\boldsymbol{x} = \boldsymbol{b}$. Ese clásico problema de la Sección II.2 minimiza $||A\boldsymbol{x} - \boldsymbol{b}||^2$. Mide distancias verticales (hacia arriba y abajo) hasta la mejor recta. Nuestro problema minimiza distancias *perpendiculares*. El problema de cuadrados mínimos estándar nos lleva a un sistema lineal $A^{\mathrm{T}}A\widehat{\boldsymbol{x}} = A^{\mathrm{T}}\boldsymbol{b}$. Nuestro problema nos lleva a *autovalores* σ^2 y vectores singulares $\boldsymbol{u}_i$ (autovectores de S).

La suma de los cuadrados de las distancias de los datos a la recta $\boldsymbol{u}_1$ es mínima.

Para verlo, separamos cada columna $\boldsymbol{a}_j$ de A en sus componentes en dirección $\boldsymbol{u}_1$ y $\boldsymbol{u}_2$:

$$\sum_1^n ||\boldsymbol{a}_j||^2 = \sum_1^n |\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{u}_1|^2 + \sum_1^n |\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{u}_2|^2. \tag{20}$$

La suma de la izquierda del signo $=$ está determinada por los datos. La primera suma de la derecha tiene términos $\boldsymbol{u}_1^{\mathrm{T}}\boldsymbol{a}_j\boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{u}_1$, resultando en $\boldsymbol{u}_1^{\mathrm{T}}(AA^{\mathrm{T}})\boldsymbol{u}_1$. Así cuando maximizamos esa suma en PCA eligiendo el primer autovector $\boldsymbol{u}_1$ de AA^{T}, minimizamos la segunda suma. Esa segunda suma de las distancias al cuadrado desde los datos a la mejor recta (o el mejor subespacio) es la menor posible.

El álgebra lineal detrás de PCA

El análisis de componentes principales es una manera de entender los datos : n puntos muestrales $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ en un espacio m-dimensional. El gráfico de datos está centrado : todas las filas de A suman cero ($A\mathbf{1} = \mathbf{0}$). La conexión crucial con el álgebra lineal se encuentra en los valores singulares σ_i y los vectores singulares $\boldsymbol{u}_i$ de A. Esos se obtienen de los autovalores $\lambda_i = \sigma_i^2$ y los autovectores de la matriz de covarianza muestral $S = AA^{\mathrm{T}}/(n-1)$.

La varianza total de los datos viene de la norma de Frobenius (al cuadrado) de A :

$$\textbf{Varianza total } \boldsymbol{T} = \boldsymbol{||A||_F^2/(n-1)} = (||\boldsymbol{a}_1||^2 + \cdots + ||\boldsymbol{a}_n||^2)/(n-1). \tag{21}$$

Esta es la **traza de $\boldsymbol{S}$** : la suma de la diagonal. El álgebra lineal nos dice que la traza es igual a la **suma de los autovalores $\boldsymbol{\sigma_i^2/(n-1)}$ de la matriz de covarianza muestral $\boldsymbol{S}$.**

La traza de S conecta la varianza total con la suma de las varianzas de las componentes principales $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r$:

$$\textbf{Varianza total} \qquad \boldsymbol{T = (\sigma_1^2 + \cdots + \sigma_r^2)/(n-1).} \tag{22}$$

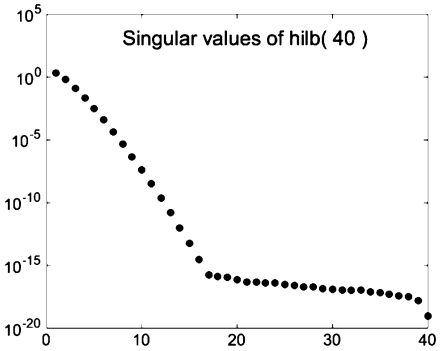
Exactamente como en la ecuación (20), la primera componente principal $\boldsymbol{u}_1$ explica una fracción σ_1^2/T de la varianza total. El siguiente vector singular $\boldsymbol{u}_2$ de A explica la siguiente fracción más grande σ_2^2/T. Cada vector singular hace lo que puede para capturar el significado de una matriz, y todos juntos lo logran.

El objetivo del teorema de Eckart-Young es que k vectores singulares (trabajando juntos) explican *más sobre los datos que cualquier otro conjunto de k vectores*. Así está justificada nuestra elección de $\boldsymbol{u}_1$ a $\boldsymbol{u}_k$ como base para el espacio k-dimensional más cercano a los n puntos muestrales.

El lector comprende que nuestra Figura I.11 muestra un agrupamiento de puntos de datos alrededor de una recta $(k = 1)$ en dimensión $m = 2$. En problemas reales es usual que $k > 1$ y $m > 2$.

El "rango efectivo" de A y S es el número de valores singulares que están por encima del punto donde el ruido tapa la verdadera señal de los datos. Generalmente este punto es visible en un "**gráfico de sedimentación**" que muestra la inflexión en el decaimiento de los valores singulares σ_i (o sus cuadrados σ_i^2). La Figura I.12 muestra el "codo" en el gráfico de sedimentación donde termina la señal y el ruido comienza a dominar.

En este ejemplo el ruido proviene de los errores de redondeo en el cálculo de valores singulares de la **Matriz de Hilbert** que es mal condicionada. El decaimiento en el resto de los verdaderos valores singulares es muy marcado. En la práctica, el ruido está en la misma matriz de datos : errores en la medición de A_0. La Sección III.3 de este libro estudia matrices como H con σ rápidamente decrecientes.



$$H_{ij} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdot & \cdot \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdot & \cdot \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} = \frac{1}{(i+j-1)}$$

Figura I.12: gráfico de sedimentación de $\sigma_1, \ldots, \sigma_{39}\,(\sigma_{40} = 0)$ para la maldita matriz de Hilbert, con el codo en el rango efectivo : $r \approx 17$ y $\sigma_r \approx 10^{-16}$.

Matrices de ceros y unos y sus propiedades

Alex Townsend y el autor de este libro comenzaron un estudio de las matrices con unos dentro de un círculo y ceros afuera. A medida que las matrices crecen, su rango aumenta. El gráfico de los valores singulares se acerca a un límite, que todavía no podemos predecir. Pero sabemos el rango.

En la Figura I.13 se encuentran dibujadas tres formas : **cuadrado**, **triángulo**, **cuarto de círculo**. Cualquier cuadrado de unos tendrá rango 1. El triangulo tiene todos los autovalores $\boldsymbol{\lambda = 1}$, y sus valores singulares son más interesantes. El rango de la matriz de "cuarto de círculo" fue nuestro primer acertijo, que resolvemos a continuación.

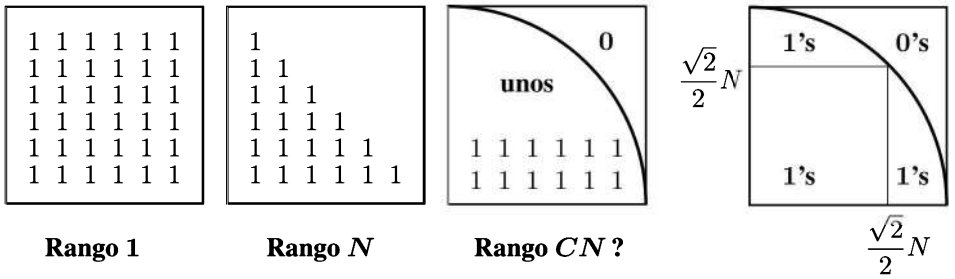


Figura I.13: cuadrado, triángulo y cuarto de círculo de unos en matrices con $N = 6$.

La reflexión de estas figuras a través del eje x producirá un rectángulo, un triángulo más grande y un semicírculo con lado $2N$. Los rangos no cambiarán porque las filas nuevas son copias de las viejas. La reflexión por el eje y producirá un cuadrado, un diamante y un círculo. Esta vez las nuevas columnas son copias de las viejas : de nuevo el mismo rango.

A partir del cuadrado y del triángulo nos damos cuenta de que el rango bajo acompaña el alineamiento horizontal-vertical. Las diagonales traen rangos altos y las diagonales de $45\,^\circ$ dan los rangos más altos.

¿Cuál es el "*rango asintótico*" del cuarto de círculo a medida que el radio $N = 6$ aumenta? Buscamos el término principal CN en el rango.

La cuarta figura muestra una manera de calcular C. Dibujamos un cuadrado de máximo tamaño en el cuarto de círculo. Esa submatriz cuadrada (todos 1) tiene rango 1. La figura sobre el cuadrado tiene $N - \frac{\sqrt{2}}{2}N$ filas (aproximadamente $0.3N$) y la figura a su lado tiene $N - \frac{\sqrt{2}}{2}N$ columnas. Esas filas y esas columnas son independientes. Sumar esos dos números produce el término principal en el rango y está confirmado numéricamente :

$$\textbf{Rango de una matriz de cuarto de círculo} \approx \boldsymbol{(2 - \sqrt{2})\, N} \text{ cuando } N \to \infty.$$

Nos concentramos en los valores singulares (no nulos) de esas matrices, trivial para el cuadrado, conocido para el triángulo, y calculable para el cuarto de círculo. Para estas y otras figuras, siempre hemos visto un **"salto singular"**. Los valores singulares no se acercan a cero. Todos los σ permanecen por encima de un límite L, y no sabemos por qué.

Los gráficos muestran los σ para el cuarto de círculo (valores calculados) y el triángulo (valores exactos). Para el triángulo de unos, la matriz inversa solo tiene una diagonal de unos sobre una diagonal de -1. Entonces $\boldsymbol{\sigma_i = \frac{1}{2}\,\mathrm{sen}\,\theta}$ en los N ángulos idénticamente espaciados $\theta_i = (2i - 1)\pi/(4N + 2)$. Por lo tanto el salto sin valores singulares alcanza $\sigma_{\text{mín}} \approx \frac{1}{2}\,\mathrm{sen}\,\frac{\pi}{2} = \frac{1}{2}$. El cuarto de círculo también tiene $\sigma_{\text{mín}} \approx \frac{1}{2}$. Aún no se conoce un "teorema del salto".

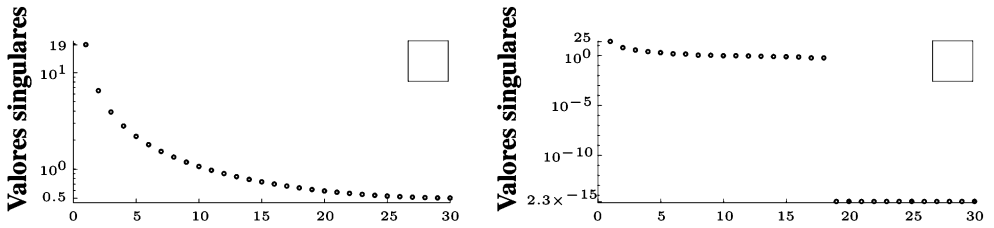


Figura I.14: los valores singulares (no nulos) para las matrices "triángulo" y "cuarto de círculo".

Problemas de la Sección I.9

1 ¿Cuáles son los valores singulares (en orden descendente) de $\boldsymbol{A - A_k}$? Omita los ceros.

2 Encuentre la aproximación de rango 1 más cercana a estas matrices (L^2 o norma de Frobenius):

$$A = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad A = \begin{bmatrix} 0 & 3 \\ 2 & 0 \end{bmatrix} \qquad A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

3 Encuentre la aproximación de rango 1 más cercana en la norma L^2 a $A = \begin{bmatrix} \cos\theta & -\operatorname{sen}\theta \\ \operatorname{sen}\theta & \cos\theta \end{bmatrix}$

4 El teorema de Eckart-Young es falso en la norma matricial $||A||_\infty$ = máximo de las sumas por filas:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \text{ tiene } ||A||_\infty = \text{máx}\frac{||A\boldsymbol{x}||_\infty}{||\boldsymbol{x}||_\infty} = \text{máx}(|a| + |b|, |c| + |d|)$$

Encuentre una matriz de rango 1 más cercana a $A = \begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix}$ que $A_1 = \frac{3}{2}\begin{bmatrix} 1 & 1 \\ 3 & 3 \end{bmatrix}$.

5 Muestre que esta norma $||A||_\infty = \text{máx}(|a| + |b|, |c| + |d|)$ no es ortogonalmente invariante:

Encuentre una matriz diagonal A donde $||QA||_\infty \neq ||A||_\infty$ con $Q = \begin{bmatrix} \cos\theta & -\operatorname{sen}\theta \\ \operatorname{sen}\theta & \cos\theta \end{bmatrix}$.

6 Si $S = Q\Lambda Q^{\mathrm{T}}$ es una matriz simétrica definida positiva, explique usando Eckart-Young por qué $\boldsymbol{q}_1\lambda_1\boldsymbol{q}_1^{\mathrm{T}}$ es la aproximación de rango 1 más cercana en la norma L^2 matricial.

7 Deduzca las derivadas $\partial E/\partial C$ y $\partial E/\partial R$ en la ecuación (19) para $n = 2$.

8 ¿Qué matrices de rango 3 tienen $||A - A_1||_2 = ||A - A_2||_2$? A_2 es $\sigma_1\boldsymbol{u}_1\boldsymbol{v}_1^{\mathrm{T}} + \sigma_2\boldsymbol{u}_2\boldsymbol{v}_2^{\mathrm{T}}$.

9 Reemplace el cuarto de círculo en la Figura I.13 con la parábola $\boldsymbol{y = 1 - x^2}$. Estime el rango CN con todos los 1 debajo de la parábola (N unos sobre los ejes). Primero quite un rectángulo de unos, que esté tocando la parábola donde la pendiente es $= -1$.

10 Si A es una matriz de 2 por 2 con $\sigma_1 \geq \sigma_2 > 0$, encuentre $||A^{-1}||_2$ y $||A^{-1}||_F^2$.

I.10. Cocientes de Rayleigh y autovalores generalizados

Esta sección retoma y extiende un tema de la Sección I.8, donde relacionamos los autovalores y autovectores de una matriz simétrica S con el **cociente de Rayleigh $R(\boldsymbol{x})$**:

$$R(\boldsymbol{x}) = \frac{\boldsymbol{x}^{\mathrm{T}} S \boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}}. \tag{1}$$

El valor máximo de $R(\boldsymbol{x})$ es el mayor autovalor $\boldsymbol{\lambda_1}$ de $\boldsymbol{S}$. Ese máximo se alcanza en el autovector $\boldsymbol{x} = \boldsymbol{q}_1$ donde $S\boldsymbol{q}_1 = \lambda_1 \boldsymbol{q}_1$:

$$\textbf{Máximo} \qquad R(\boldsymbol{q}_1) = \frac{\boldsymbol{q}_1^{\mathrm{T}} S \boldsymbol{q}_1}{\boldsymbol{q}_1^{\mathrm{T}} \boldsymbol{q}_1} = \frac{\boldsymbol{q}_1^{\mathrm{T}} \lambda_1 \boldsymbol{q}_1}{\boldsymbol{q}_1^{\mathrm{T}} \boldsymbol{q}_1} = \boldsymbol{\lambda_1}. \tag{2}$$

Análogamente el valor mínimo de $R(\boldsymbol{x})$ es igual al menor autovalor λ_n de S. Ese mínimo se alcanza en el "autovector inferior" $\boldsymbol{x} = \boldsymbol{q}_n$. Además, todos los autovectores $\boldsymbol{x} = \boldsymbol{q}_k$ de S correspondientes a los autovalores *entre* λ_n y λ_1 son puntos de ensilladura de $R(\boldsymbol{x})$. En los puntos de ensilladuras se tienen derivadas primeras iguales a cero pero no son máximos ni mínimos.

$$\textbf{Punto de ensilladura} \qquad \text{Todas } \frac{\partial R}{\partial x_i} = 0 \text{ en } \boldsymbol{x} = \boldsymbol{q}_k. \text{ Entonces } R(\boldsymbol{q}_k) = \frac{\boldsymbol{q}_k^{\mathrm{T}} \lambda_k \boldsymbol{q}_k}{\boldsymbol{q}_k^{\mathrm{T}} \boldsymbol{q}_k} = \lambda_k. \tag{3}$$

Estos hechos se conectan con la descomposición en valores singulares de A. La conexión se realizó a través de $S = A^{\mathrm{T}} A$. Para esa matriz S definida (o semidefinida) positiva, el cociente de Rayleigh se conecta con a la norma de A (al cuadrado). Y el mayor autovalor de S es $\sigma_1^2(A)$:

$$\boxed{||A||^2 = \max \frac{||A\boldsymbol{x}||^2}{||\boldsymbol{x}||^2} = \max \frac{\boldsymbol{x}^{\mathrm{T}} A^{\mathrm{T}} A \boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}} = \max \frac{\boldsymbol{x}^{\mathrm{T}} S \boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}} = \lambda_1(S) = \sigma_1^2(A).} \tag{4}$$

De esta manera un problema simétrico de autovalores es también un problema de optimización: **maximizar $R(\boldsymbol{x})$.**

Autovalores y autovectores generalizados

Las aplicaciones en estadística y ciencia de datos nos llevan al próximo paso. Las aplicaciones en ingeniería y mecánica apuntan en la misma dirección. *Una segunda matriz simétrica* M entra en el denominador de $R(\boldsymbol{x})$:

$$\textbf{Cociente de Rayleigh generalizado} \qquad R(\boldsymbol{x}) = \frac{\boldsymbol{x}^{\mathrm{T}} S \boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}} M \boldsymbol{x}} \tag{5}$$

En problemas dinámicos M generalmente es la "matriz de masa" o la "matriz de inercia". En estadística M es generalmente la **matriz de covarianza**. La construcción de las matrices de covarianza y su aplicación en la clasficación de datos se desarrollará en el capítulo sobre probabilidad y estadística.

Aquí nuestro objetivo es mostrar cómo el problema de autovalores $S\boldsymbol{x} = \lambda \boldsymbol{x}$ cambia a $S\boldsymbol{x} = \lambda M\boldsymbol{x}$ cuando $R(\boldsymbol{x})$ es $\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}/\boldsymbol{x}^{\mathrm{T}} M\boldsymbol{x}$. Este es el problema simétrico de autovalores *generalizado* .

Si M es definida positiva, el máximo de $R(x)$ es el mayor autovalor de $M^{-1}S$.

Reduciremos este problema generalizado $S\boldsymbol{x} = \lambda M\boldsymbol{x}$ a un problema de autovalores ordinario $H\boldsymbol{y} = \lambda \boldsymbol{y}$. Pero tenemos que ver que la elección $H = M^{-1}S$ no es perfecta. La razón es simple : ¡ $M^{-1}S$ *suele ser no simétrica* ! Incluso una matriz diagonal M lo deja en claro. La raíz cuadrada $M^{1/2}$ de la misma matriz diagonal sugiere la manera correcta de mantener la simetría.

$$M^{-1}S = \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}^{-1} \begin{bmatrix} a & b \\ b & c \end{bmatrix} = \begin{bmatrix} a/m_1 & \boldsymbol{b/m_1} \\ \boldsymbol{b/m_2} & c/m_2 \end{bmatrix} \quad \textbf{no es simétrica},$$

$$H = M^{-1/2} S M^{-1/2} = \begin{bmatrix} a/m_1 & \boldsymbol{b/\sqrt{m_1 m_2}} \\ \boldsymbol{b/\sqrt{m_1 m_2}} & c/m_2 \end{bmatrix} \quad \textbf{es simétrica}.$$

Esas matrices $M^{-1}S$ y $H = M^{-1/2}SM^{-1/2}$ tienen los mismos autovalores. Esta H luce rara, pero la simetría se mantiene cuando elegimos la raíz cuadrada simétrica de M y M^{-1}. **Toda M definida positiva tiene una raíz cuadrada definida positiva.**

El ejemplo diagonal mencionado tenía $M^{1/2} = \operatorname{diag}(\sqrt{m_1}, \sqrt{m_2})$. Su inversa es $M^{-1/2}$. En todos los casos, diagonalizamos M y tomamos la raíz cuadrada de cada autovalor :

$$\textbf{Si } M = Q\Lambda Q^{\mathrm{T}} \textbf{ tiene } \Lambda > 0 \textbf{ entonces } M^{1/2} = Q\Lambda^{1/2} Q^{\mathrm{T}} \textbf{ tiene } \Lambda^{1/2} > 0. \quad (6)$$

Elevando $M^{1/2}$ al cuadrado recuperamos $Q\Lambda^{1/2}Q^{\mathrm{T}}Q\Lambda^{1/2}Q^{\mathrm{T}} = Q\Lambda Q^{\mathrm{T}}$ que es M. ¡ No usaremos $M^{1/2}$ o $M^{-1/2}$ numéricamente !. El problema de autovalores generalizados $S\boldsymbol{x} = \lambda M\boldsymbol{x}$ se resuelve en MATLAB con el comando **eig**(S, M). **Julia** y **Python** y **R** y todos los paquetes de álgebra lineal completos incluyen esta extensión a $S\boldsymbol{x} = \lambda M\boldsymbol{x}$.

Un cociente de Rayleigh con $\boldsymbol{x}^{\mathrm{T}} M\boldsymbol{x}$ fácilmente se convierte en un cociente con $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}$:

$$\textbf{Sea } \boldsymbol{x} = \boldsymbol{M^{-1/2}y}. \quad \text{Entonces} \quad \frac{\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}} M\boldsymbol{x}} = \frac{\boldsymbol{y}^{\mathrm{T}}(M^{-1/2})^{\mathrm{T}} S M^{-1/2}\boldsymbol{y}}{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}} = \frac{\boldsymbol{y}^{\mathrm{T}} H\boldsymbol{y}}{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}}. \quad (7)$$

Esto cambia el problema generalizado $S\boldsymbol{x} = \lambda M\boldsymbol{x}$ a un problema simétrico ordinario $H\boldsymbol{y} = \lambda \boldsymbol{y}$. Si S y M son definidas positivas, entonces también lo es $\boldsymbol{H = M^{-1/2}SM^{-1/2}}$. El mayor cociente de Rayleigh sigue siendo el mayor autovalor λ_1. Además podemos mostrar el autovector principal $\boldsymbol{y}_1$ de H y el autovector principal $\boldsymbol{x}_1$ de $M^{-1}S$:

$$\max \frac{\boldsymbol{y}^{\mathrm{T}} H\boldsymbol{y}}{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}} = \lambda_1 \text{ cuando } \boldsymbol{Hy_1 = \lambda_1 y_1}. \text{ Entonces } \boldsymbol{Sx_1 = \lambda_1 M x_1} \text{ para } \boldsymbol{x_1 = M^{-1/2} y_1}.$$

Ejemplo 1 Resuelva $S\boldsymbol{x} = \lambda M\boldsymbol{x}$ para $S = \begin{bmatrix} \mathbf{4} & \mathbf{-2} \\ \mathbf{-2} & \mathbf{4} \end{bmatrix}$ y $M = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{2} \end{bmatrix}$.

Solución Nuestros problemas de autovalores son $(S - \lambda M)\,\boldsymbol{x} = \mathbf{0}$ y $(H - \lambda I)\,\boldsymbol{y} = \mathbf{0}$. Los **mismos** $\boldsymbol{\lambda}$ se encuentran en ambos determinantes : $\det(S - \lambda M) = 0$ y $\det(H - \lambda I) = 0$.

$$\det(S - \lambda M) = \det \begin{bmatrix} 4 - \lambda & -2 \\ -2 & 4 - 2\lambda \end{bmatrix} = 2\lambda^2 - 12\lambda + 12 = 0 \quad \text{da} \quad \boldsymbol{\lambda = 3 \pm \sqrt{3}}\,.$$

Si preferimos trabajar con una matriz $H = M^{-1/2} S M^{-1/2}$, primero debemos calcularla :

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 4 & -2 \\ -2 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 4 & -\sqrt{2} \\ -\sqrt{2} & 2 \end{bmatrix}.$$

Así sus autovalores se obtienen del determinante de $H - \lambda I$:

$$\det \begin{bmatrix} 4 - \lambda & -\sqrt{2} \\ -\sqrt{2} & 2 - \lambda \end{bmatrix} = \lambda^2 - 6\lambda + 6 = 0 \;\text{ también da }\; \lambda = 3 \pm \sqrt{3}.$$

Esta ecuación es solo la mitad de la anterior $2\lambda^2 - 12\lambda + 12 = 0$: los mismos λ para H y $M^{-1}S$.

En ingeniería mecánica esos λ nos indicarían las frecuencias $\omega = \sqrt{\lambda}$ para dos masas oscilantes $m_1 = 1$ y $m_2 = 2$ en una línea de resortes. S nos dice la rigidez en los tres resortes que conectan las dos masas con puntos finales fijos.

La ecuación diferencial es la ley de Newton $M d^2\boldsymbol{u}/dt^2 = -S\boldsymbol{u}$.

Los autovectores generalizados son M-ortogonales

Un dato crucial sobre una matriz simétrica S es que dos autovectores cualesquiera son ortogonales (cuando los autovalores correspondientes son diferentes). ¿ Se extiende esto a $S\boldsymbol{x}_1 = \lambda_1 M\boldsymbol{x}_1$ con dos matrices simétricas ? La respuesta inmediata es **no**, pero la respuesta correcta es **si**. Para esa respuesta tenemos que suponer que M es definida positiva y tenemos que cambiar de $\boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_2 = 0$ a "M-ortogonalidad" de $\boldsymbol{x}_1$ y $\boldsymbol{x}_2$. **Dos vectores son $\boldsymbol{M}$-ortogonales si $\boldsymbol{x_1^{\mathrm{T}} M x_2 = 0}$.**

$$\boxed{\boldsymbol{x_1^{\mathrm{T}} M x_2 = 0 \;\text{ si }\; S x_1 = \lambda_1 M x_1 \;\text{ y }\; S x_2 = \lambda_2 M x_2 \;\text{ con }\; \lambda_1 \neq \lambda_2.}} \tag{8}$$

Prueba. Multiplicamos una ecuación por $\boldsymbol{x}_2^{\mathrm{T}}$ y multiplicamos la otra ecuación por $\boldsymbol{x}_1^{\mathrm{T}}$:

$$\boldsymbol{x}_2^{\mathrm{T}} S\boldsymbol{x}_1 = \lambda_1 \boldsymbol{x}_2^{\mathrm{T}} M\boldsymbol{x}_1 \quad \text{y} \quad \boldsymbol{x}_1^{\mathrm{T}} S\boldsymbol{x}_2 = \lambda_2 \boldsymbol{x}_1^{\mathrm{T}} M\boldsymbol{x}_2$$

Como S y M son simétricas, trasponer la primera ecuación da $\boldsymbol{x}_1^{\mathrm{T}} S\boldsymbol{x}_2 = \lambda_1 \boldsymbol{x}_1^{\mathrm{T}} M\boldsymbol{x}_2$. Restamos la segunda ecuación :

$$(\lambda_1 - \lambda_2)\,\boldsymbol{x}_1^{\mathrm{T}} M\boldsymbol{x}_2 = 0 \quad \text{y como} \quad \lambda_1 \neq \lambda_2 \quad \text{esto requiere} \quad \boldsymbol{x}_1^{\mathrm{T}} M\boldsymbol{x}_2 = 0. \tag{9}$$

Además $\boldsymbol{x}_1^{\mathrm{T}} S\boldsymbol{x}_2 = 0$. Podemos verificar esta concusión en las matrices S y M del Ejemplo 1.

Ejemplo 2 Encuentre los autovectores asociados a $\lambda_1 = 3+\sqrt{3}$ y a $\lambda_2 = 3-\sqrt{3}$. Verifique que $\boldsymbol{x}^{\mathrm{T}} M \boldsymbol{y} = 0$.

Solución Los autovectores $\boldsymbol{x}$ e $\boldsymbol{y}$ están en los espacios nulos: $(S-\lambda_1 M)\boldsymbol{x}=\boldsymbol{0}$ y $(S-\lambda_2 M)\boldsymbol{y}=\boldsymbol{0}$.

$$(S-\lambda_1 M)\,\boldsymbol{x} = \begin{bmatrix} 4-(3+\sqrt{3}) & -2 \\ -2 & 4-2(3+\sqrt{3}) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \text{da} \quad \boldsymbol{x} = c \begin{bmatrix} \mathbf{2} \\ \mathbf{1+\sqrt{3}} \end{bmatrix}$$

$$(S-\lambda_2 M)\,\boldsymbol{y} = \begin{bmatrix} 4-(3-\sqrt{3}) & -2 \\ -2 & 4-2(3-\sqrt{3}) \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \quad \text{da} \quad \boldsymbol{y} = c \begin{bmatrix} \mathbf{2} \\ \mathbf{1-\sqrt{3}} \end{bmatrix}$$

Esos autovectores $\boldsymbol{x}$ y $\boldsymbol{y}$ no son ortogonales, pero son M-ortogonales porque

$$\boldsymbol{x}^{\mathrm{T}} M \boldsymbol{y} = \begin{bmatrix} 2 & 1+\sqrt{3} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ 1-\sqrt{3} \end{bmatrix} = 0.$$

M semidefinida positiva : no invertible

Existen aplicaciones importantes en las que la matriz M es solo **semidefinida positiva**. **¡Entonces $\boldsymbol{x}^{\mathrm{T}} M \boldsymbol{x}$ puede ser cero!** La matriz M no será invertible. El cociente $\boldsymbol{x}^{\mathrm{T}} S \boldsymbol{x} / \boldsymbol{x}^{\mathrm{T}} M \boldsymbol{x}$ puede ser infinito. Las matrices $M^{-1/2}$ y H ni siquiera existen. Sigue siendo necesario resolver el problema de autovalores $S\boldsymbol{x} = \lambda M \boldsymbol{x}$, pero ahora es posible un autovalor infinito $\lambda = \infty$.

En estadística M es generalmente una matriz de covarianza. Sus elementos diagonales nos dicen las varianzas de dos o más mediciones. Sus elementos por fuera de la diagonal nos dicen las "covarianzas entre las mediciones". Si absurdamente repetimos las mismas observaciones (o si un experimento está completamente determinado por otro) entonces la matriz de covarianza M es singular. Su determinante es cero y no es invertible. El cociente de Rayleigh (que divide por $\boldsymbol{x}^{\mathrm{T}} M \boldsymbol{x}$) puede volverse infinito.

Una manera de ver esto matemáticamente es escribir $S\boldsymbol{x} = \lambda M \boldsymbol{x}$ con α y β.

$$\boldsymbol{\alpha S x = \beta M x} \ \text{ con } \ \boldsymbol{\alpha \geq 0} \ \text{ y } \ \boldsymbol{\beta \geq 0} \ \text{ y autovalores } \ \boldsymbol{\lambda = \frac{\beta}{\alpha}}. \tag{10}$$

λ será un autovalor positivo ordinario si $\alpha > 0$ y $\beta > 0$. Hasta podemos normalizar esos dos números pidiendo $\alpha^2 + \beta^2 = 1$, pero ahora vemos otras tres posibilidades en la ecuación (10) :

$\alpha > 0$ y $\beta = 0$ Entonces $\boldsymbol{\lambda = 0}$ y $S\boldsymbol{x} = 0\boldsymbol{x}$: un autovalor cero de S

$\alpha = 0$ y $\beta > 0$ Entonces $\boldsymbol{\lambda = \infty}$ y $M\boldsymbol{x} = 0$: M no es invertible

$\alpha = 0$ y $\beta = 0$ Entonces $\boldsymbol{\lambda = \frac{0}{0}}$ es indeterminado : $M\boldsymbol{x} = \boldsymbol{0}$ y también $S\boldsymbol{x} = \boldsymbol{0}$.

$\alpha = 0$ puede ocurrir cuando tenemos agrupamientos (clusters) de datos y el número de muestras en un agrupamiento es menor que el número de características que medimos. Este es el problema de **tamaño de muestra pequeño**. Suele ocurrir.

Comprenderá que las matemáticas se vuelven más delicadas. Utilizar la SVD (cuando se factoriza una matriz de datos en $A = U\Sigma V^{\mathrm{T}}$ con vectores singulares $\boldsymbol{v}$ que vienen de los autovectores de $S = A^{\mathrm{T}} A$) no es suficiente. **Hay que generalizar la SVD. Necesitamos permitir la presencia de una segunda matriz M, lo que nos lleva a la GSVD.**

SVD Generalizada (Simplificada)

En su forma más general, esta factorización es complicada. Involucra dos matrices S y M y permite que sean singulares. En este libro tiene sentido trabajar con el caso usual, que suele ser el mejor, cuando las matrices simétricas son definidas positivas. Así podemos ver que el propósito principal de la GSVD es factorizar dos matrices al mismo tiempo.

Recordemos que la SVD clásica factoriza una matriz rectangular A en $U\Sigma V^{\mathrm{T}}$. Comienza con A en lugar de $S = A^{\mathrm{T}}A$. Análogamente aquí **comenzamos con dos matrices $\boldsymbol{A}$ y $\boldsymbol{B}$**. Nuestra simplificación consiste en suponer que ambas matrices son de rango n, altas y delgadas. Sus tamaños son m_A por n y m_B por n. Entonces $\boldsymbol{S = A^{\mathrm{T}}A}$ y $\boldsymbol{M = B^{\mathrm{T}}B}$ son de n por n y definidas positivas.

Descomposición en valores singulares generalizada

A y B pueden factorizarse como $A = U_A\Sigma_A Z$ y $B = U_B\Sigma_B Z$ (misma Z)

U_A y U_B son matrices ortogonales (tamaño m_A y m_B)

Σ_A y Σ_B son matrices diagonales positivas (con $\Sigma_A^{\mathrm{T}}\Sigma_A + \Sigma_B^{\mathrm{T}}\Sigma_B = I_{n\times n}$)

Z es una matriz invertible (tamaño n)

Observemos que Z probablemente no sea una matriz ortogonal. Eso sería pedir demasiado. La notable propiedad de Z es **diagonalizar simultáneamente** a $S = A^{\mathrm{T}}A$ y a $M = B^{\mathrm{T}}B$:

$$A^{\mathrm{T}}A = Z^{\mathrm{T}}\Sigma_A^{\mathrm{T}}U_A^{\mathrm{T}}U_A\Sigma_A Z = \boldsymbol{Z^{\mathrm{T}}\left(\Sigma_A^{\mathrm{T}}\Sigma_A\right)Z} \quad \text{y} \quad B^{\mathrm{T}}B = \boldsymbol{Z^{\mathrm{T}}\left(\Sigma_B^{\mathrm{T}}\Sigma_B\right)Z}. \tag{11}$$

Este es un hecho del álgebra lineal: dos matrices definidas positivas cualesquiera se pueden diagonalizar por la misma matriz Z. De la ecuación (9), sus columnas pueden ser $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. Eso se sabía antes de que se inventara la GSVD. Y como no requerimos de la ortogonalidad, podemos escalar Z para que $\Sigma_A^{\mathrm{T}}\Sigma_A + \Sigma_B^{\mathrm{T}}\Sigma_B = I$. También podemos ordenar sus columnas $\boldsymbol{x}_k$ para que los n números positivos σ_A estén en orden descendiente (en Σ_A).

Observemos también el significado de "diagonalizar". La ecuación (11) no contiene Z^{-1} y Z, sino que contiene Z^{T} y Z. Con Z^{-1} tenemos una tranformación de **similaridad** o de **semejanza**, que preserva los autovalores. Con Z^{T} tenemos una transformación de **congruencia** $Z^{\mathrm{T}}SZ$, que preserva la simetría. (Así los autovalores de S y $Z^{\mathrm{T}}SZ$ tienen los **mismos signos**. Esta es la ley de inercia de Sylvester en los problemas de la sección III.2. Aquí los signos son todos positivos.) La simetría de $S = A^{\mathrm{T}}A$ y el hecho que $M = B^{\mathrm{T}}B$ sea definida positiva permite que una Z diagonalice ambas matrices.

El problema 5 de esta sección le guiará a una prueba de esta GSVD simplificada.

El análisis de discriminante lineal de Fisher (LDA)

He aquí una linda aplicación en estadística y aprendizaje automático. Tenemos muestras de dos poblaciones distintas y están mezcladas. Conocemos los datos básicos de cada una: su valor medio m y su dispersión σ alrededor de la media m. Así tenemos una media m_1 y una varianza σ_1^2 para la primera población, y m_2 y σ_2^2 para la segunda población. Si todas las muestras están mezcladas y eligiéramos una, *¿cómo nos damos cuenta a cuál población pertenece* ?

El "discriminante lineal" de Fisher responde la pregunta.

De hecho, el problema es un poco más complicado. Cada muestra tiene varios predictores, como la edad, altura y peso de un niño. Esto es normal en aprendizaje automático, tener un "vector de predictores" como $\boldsymbol{f}$ = (edad, altura, peso) para cada muestra. Si una muestra tiene un vector de predictores $\boldsymbol{f}$, ¿ de qué población probablemente vino ? Comenzamos con vectores, no escalares.

Tenemos una edad promedio $\boldsymbol{m_a}$, una altura promedio $\boldsymbol{m_h}$ y un peso promedio $\boldsymbol{m_w}$ para cada población. **La media (el promedio) de la población 1 es un vector** $\boldsymbol{m_1} = (\boldsymbol{m_{a1}}, \boldsymbol{m_{h1}}, \boldsymbol{m_{w1}})$. La población 2 también tiene un vector $\boldsymbol{m_2}$ de edad promedio, altura promedio y peso promedio. Y la varianza σ para cada población se vuelve una **matriz** Σ de $\mathbf{3 \times 3}$ para medir su dispersión alrededor de la media. Esta "matriz de covarianza" será clave en el Capítulo V sobre estadística. Por ahora tenemos $\boldsymbol{m_1}, \boldsymbol{m_2}, \Sigma_1, \Sigma_2$ y queremos una regla para discriminar entre las dos poblaciones.

El test de Fisher tiene una forma sencilla : *encuentra un vector de separación* $\boldsymbol{v}$. Si la muestra satisface $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{f} > c$ entonces nuestra mejor conjetura es la población 1. Si $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{f} < c$ entonces la muestra probablemente pertenece a la población 2. El vector $\boldsymbol{v}$ trata de separar las dos poblaciones (tanto como le sea posible), maximiza el cociente de separación R :

$$\textbf{Cociente de separación} \qquad R = \frac{(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{m}_1 - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{m}_2)^2}{\boldsymbol{x}^{\mathrm{T}}\Sigma_1\boldsymbol{x} + \boldsymbol{x}^{\mathrm{T}}\Sigma_2\boldsymbol{x}} \tag{12}$$

Ese cociente R tiene la forma $\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}/\boldsymbol{x}^{\mathrm{T}}M\boldsymbol{x}$. La matriz S es $(\boldsymbol{m_1} - \boldsymbol{m_2})(\boldsymbol{m_1} - \boldsymbol{m_2})^{\mathrm{T}}$. La matriz M es $\Sigma_1 + \Sigma_2$. **Conocemos la regla** $\boldsymbol{Sv} = \boldsymbol{\lambda M v}$ **para el vector** $\boldsymbol{x} = \boldsymbol{v}$ **que maximiza el cociente de separación** $\boldsymbol{R}$**.**

Fisher pudo encontrar ese autovector $\boldsymbol{v}$ de $M^{-1}S$. También podemos nosotros, porque la matriz $S = (\boldsymbol{m_1} - \boldsymbol{m_2})(\boldsymbol{m_1} - \boldsymbol{m_2})^{\mathrm{T}}$ *tiene rango uno*. Así que $S\boldsymbol{v}$ siempre está en la dirección $\boldsymbol{m_1} - \boldsymbol{m_2}$. Luego $M\boldsymbol{v}$ también debe estar en esa dirección, para que se verifique $S\boldsymbol{v} = \lambda M\boldsymbol{v}$. *Entonces* $\boldsymbol{v} = \boldsymbol{M^{-1}}(\boldsymbol{m_1} - \boldsymbol{m_2})$.

Este fue un lindo problema porque encontramos el autovector $\boldsymbol{v}$. Tiene sentido que cuando una muestra desconocida tenga un vector de predictores $\boldsymbol{f}$ = (edad, altura, peso) miremos los números $\boldsymbol{m_1^{\mathrm{T}}f}$ y $\boldsymbol{m_2^{\mathrm{T}}f}$. Si estuvieramos listos para una discusión estadística plena (*y no lo estamos*), entonces podríamos ver cómo la matriz de pesos $M = \Sigma_0 + \Sigma_1$ se usa en el test final de $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{f}$. Por ahora nos alcanza con decir : los vectores de predictores $\boldsymbol{f}$ de las dos poblaciones se encuentran separados tanto como sea posible por un plano perpendicular a $\boldsymbol{v}$.

Resumen. Tenemos dos nubes de puntos en un espacio de predictores tridimensional. Tratamos de separarlos con un plano, lo que no siempre es posible. Fisher propuso un plano razonable.

Las redes neuronales tendrán éxito al permitir la separación por superficies *que no son necesariamente planos.*

Problemas de la Sección I.10

1 Resuelva $(S - \lambda M)\,\boldsymbol{x} = \boldsymbol{0}$ y $(H - \lambda I)\,\boldsymbol{y} = \boldsymbol{0}$ para la matriz $H = M^{-1/2} S M^{-1/2}$:

$$S = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix} \qquad M = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$$

El primer paso es encontrar λ_1 y λ_2 a partir de $\det(S - \lambda M) = 0$. La ecuación $\det(H - \lambda I) = 0$ debería producir los mismos λ_1 y λ_2. Esos autovalores producirán dos autovectores $\boldsymbol{x}_1$ y $\boldsymbol{x}_2$ de $S - \lambda M$ y dos autovectores $\boldsymbol{y}_1$ y $\boldsymbol{y}_2$ de $H - \lambda I$.

Verifique que $\boldsymbol{x}_1^{\mathrm{T}} \boldsymbol{x}_2$ no es cero pero que $\boldsymbol{x}_1^{\mathrm{T}} M \boldsymbol{x}_2 = 0$. H es simétrica así que $\boldsymbol{y}_1^{\mathrm{T}} \boldsymbol{y}_2 = 0$.

2 (a) Para $\boldsymbol{x} = (a, b)$ e $\boldsymbol{y} = (c, d)$ escriba los cocientes de Rayleigh del Problema 1 como

$$R^*(\boldsymbol{x}) = \frac{\boldsymbol{x}^{\mathrm{T}} S \boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}} M \boldsymbol{x}} = \frac{(5a^2 + 8ab + 5b^2)}{(\cdots + \cdots + \cdots)} \quad \text{y} \quad R(\boldsymbol{y}) = \frac{\boldsymbol{y}^{\mathrm{T}} H \boldsymbol{y}}{\boldsymbol{y}^{\mathrm{T}} \boldsymbol{y}} = \frac{20c^2 + 16cd + 5d^2}{(\cdots + \cdots)}$$

(b) Derive $R(\boldsymbol{y})$ con respecto a c y d para encontrar su máximo y mínimo.

(c) Derive $R^*(\boldsymbol{x})$ con respecto a a y b para encontrar su máximo y mínimo.

(d) Verifique que esos máximos se alcanzan en autovectores de $(S - \lambda M)\,\boldsymbol{x} = \boldsymbol{0}$ y $(H - \lambda I)\,\boldsymbol{y} = \boldsymbol{0}$.

3 ¿Cómo se relacionan los autovectores $\boldsymbol{x}_1$ y $\boldsymbol{x}_2$ con los autovectores $\boldsymbol{y}_1$ e $\boldsymbol{y}_2$?

4 Cambie M por $\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ y resuelva $S\boldsymbol{x} = \lambda M \boldsymbol{x}$. Ahora M es singular y uno de los autovalores λ es infinito, pero su autovector $\boldsymbol{x}_2$ todavía es M-ortogonal al autovector $\boldsymbol{x}_1$.

5 Comience con matrices S y M simétricas y definidas positivas. Los autovectores de S conforman una matriz ortogonal Q de manera que $Q^{\mathrm{T}} S Q = \Lambda$ sea diagonal. ¿Cuál es la matriz diagonal D tal que $D^{\mathrm{T}} \Lambda D = I$? Ahora tenemos $D^{\mathrm{T}} Q^{\mathrm{T}} S Q D = I$ y miramos a $D^{\mathrm{T}} Q^{\mathrm{T}} M Q D$.

Su matriz de autovectores Q_2 da $Q_2^{\mathrm{T}} I Q_2 = I$ y $Q_2^{\mathrm{T}} D^{\mathrm{T}} Q^{\mathrm{T}} M Q D Q_2 = \Lambda_2$.

Muestre que $Z = QDQ_2$ diagonaliza ambas congruencias: $Z^{\mathrm{T}} S Z$ y $Z^{\mathrm{T}} M Z$ en la GSVD.

6 (a) ¿Por qué toda congruencia $Z^{\mathrm{T}} S Z$ preserva la simetría de S ?

(b) ¿Por qué es $Z^{\mathrm{T}} S Z$ definida positiva cuando S es definida positiva y Z es cuadrada e invertible ? Aplique el test de energía a $Z^{\mathrm{T}} S Z$. No se olvide de explicar por qué $Z\boldsymbol{x}$ no es el vector cero.

7 Para Z invertible, ¿qué matrices $Z^{\mathrm{T}} I Z$ son congruentes a la matriz identidad ?

8 Resuelva este problema matricial esencial del análisis discriminante lineal de Fisher :

$$\text{Si } R(\boldsymbol{x}) = \frac{\boldsymbol{x}^{\mathrm{T}} S \boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}} M \boldsymbol{x}} \text{ y } S = \boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} \quad \text{¿qué vector } \boldsymbol{x} \text{ maximiza } R(\boldsymbol{x}) \text{ ?}$$

I.11. Normas de vectores, de funciones y de matrices

La norma de un vector no nulo $\boldsymbol{v}$ es un número positivo $||\boldsymbol{v}||$. Ese número mide la "longitud" del vector. Hay muchas maneras útiles de medir longitud (diferentes normas). Cada norma para vectores o funciones o matrices debe compartir estas dos propiedades del valor absoluto $|c|$ de un número:

Toda norma

Multiplicar $\boldsymbol{v}$ por $\boldsymbol{c}$ **(Reescalado)** $\quad \boxed{||c\,v|| = |c|\,||v||}$ (1)

Sumar $\boldsymbol{v}$ y $\boldsymbol{w}$ **(Desigualdad triangular)** $\quad \boxed{||v+w|| \le ||v|| + ||w||}$ (2)

Comenzamos con tres normas especiales: las más importantes por lejos. Son la norma $\boldsymbol{\ell^2}$, la norma $\boldsymbol{\ell^1}$ y la norma $\boldsymbol{\ell^\infty}$ del vector $\boldsymbol{v} = (v_1, \ldots, v_n)$. El vector $\boldsymbol{v}$ pertenece a $\mathbf{R}^n$ (v_i real) o a $\mathbf{C}^n$ (v_i complejo):

norma ℓ^2 = **norma euclídea**	$\|\|v\|\|_2$	$= \sqrt{\|v_1\|^2 + \cdots + \|v_n\|^2}$
norma ℓ^1 = **norma uno**	$\|\|v\|\|_1$	$= \|v_1\| + \|v_2\| + \cdots + \|v_n\|$
norma ℓ^∞ = **norma del máximo**	$\|\|v\|\|_\infty$	$=$ **máximo de** $\|v_1\|, \ldots, \|v_n\|$

El vector de todos unos $\boldsymbol{v} = (1, 1, \ldots, 1)$ tiene normas $||\boldsymbol{v}||_2 = \sqrt{n}$, $||\boldsymbol{v}||_1 = n$ y $||\boldsymbol{v}||_\infty = 1$.

Estas tres normas son los casos particulares $p = 2$, $p = 1$ y $p = \infty$ de la norma ℓ^p, $||\boldsymbol{v}||_p = (|v_1|^p + \cdots + |v_n|^p)^{1/p}$. Esta figura muestra vectores con **norma 1**: $p = \frac{1}{2}$ es ilegal.

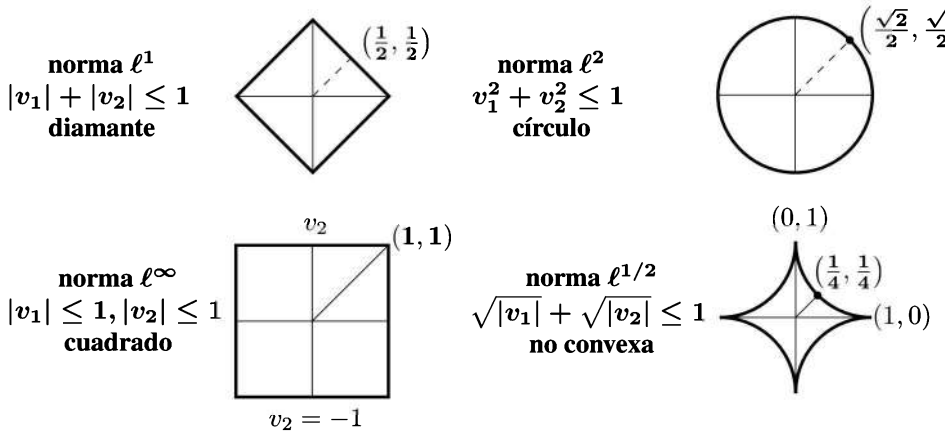


Figura I.15: las normas vectoriales importantes $||\boldsymbol{v}||_1, ||\boldsymbol{v}||_2, ||\boldsymbol{v}||_\infty$ y una falla ($p = 0$ también falla).

La falla para $p = \frac{1}{2}$ está en la desigualdad triangular: $(1, 0)$ y $(0, 1)$ tienen norma 1, pero su suma $(1, 1)$ tiene norma $2^{1/p} = 4$. Solo $1 \le p \le \infty$ producen normas $||\boldsymbol{v}||_p$ aceptables.

El mínimo de $||v||_p$ en la recta $a_1v_1 + a_2v_2 = 1$

¿ Qué punto en una recta diagonal como $3v_1 + 4v_2 = 1$ se encuentra más cercano a $(0, 0)$? La respuesta (y el significado de "cercano") dependerá de la norma. Esta es otra manera de ver diferencias importantes entre ℓ^1 y ℓ^2 y ℓ^∞. Veamos un primer ejemplo con una característica muy especial : **la minimización en ℓ^1 produce soluciones ralas**.

Para encontrar el punto más cercano a $(0, 0)$, expandimos el diamante ℓ^1 y el círculo ℓ^2 y el cuadrado ℓ^∞ **hasta que tocan la recta diagonal**. Para cada p, el punto de contacto $\boldsymbol{v^*}$ resolverá nuestro problema de optimización :

Minimizamos $||\boldsymbol{v}||_p$ entre los vectores (v_1, v_2) pertenecientes a la recta $3v_1 + 4v_2 = 1$

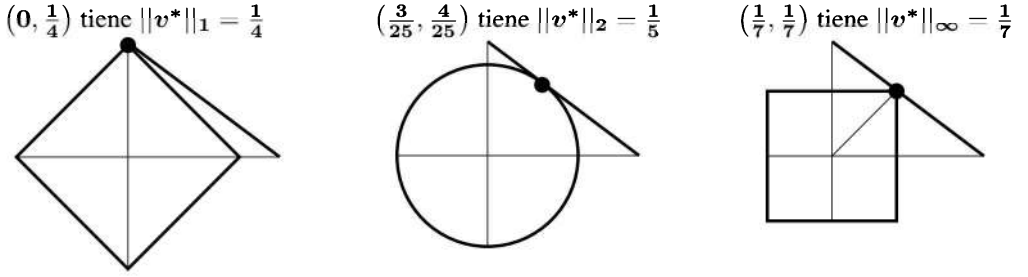


Figura I.16: las soluciones $\boldsymbol{v^*}$ para las minimizaciones de ℓ^1, ℓ^2 y ℓ^∞. La primera es **rala**.

La primera figura presenta una propiedad muy importante de la solución minimizadora para el problema ℓ^1 : **esa solución $\boldsymbol{v^*}$ tiene componentes nulas**. El vector $\boldsymbol{v^*}$ es "ralo". Esto se debe a que **un diamante toca una recta en una punta**. La recta (o el hiperplano en dimensiones superiores) contiene a los vectores que resuelven las m restricciones $A\boldsymbol{v} = \boldsymbol{b}$. La superficie del diamante contiene vectores con la misma norma ℓ^1. El diamante se expande hasta alcanzar a tocar la recta con uno de sus vértices. Los problemas de esta sección y la Sección III.4 volverán sobre este problema de "**selección de base**" y problemas relacionados con la norma ℓ^1.

Lo esencial es que **las soluciones a esos problemas son ralas**. Tienen pocas componentes distintas de cero y esas componentes tienen su significado. En contraste, la solución por cuadrados mínimos (que usa ℓ^2) tiene muchas componentes pequeñas y que no son interesantes. Al elevar al cuadrado, esas componentes se vuelven muy pequeñas y apenas afectan la distancia ℓ^2.

Una observación final : **la "norma $\boldsymbol{\ell^0}$"** de un vector $\boldsymbol{v}$ cuenta la cantidad de componentes no nulas, pero esta no es una verdadera norma. Los puntos con $||\boldsymbol{v}||_0 = 1$ son todos los puntos del eje x o del eje y: solo una componente distinta de cero. El dibujo para $p = \frac{1}{2}$ en la página anterior se vuelve más extremo: solo una cruz o un esqueleto sobre los dos ejes.

Por supuesto este esqueleto no es para nada convexo. La "norma cero" viola el requerimiento fundamental de que $||2\boldsymbol{v}|| = 2\,||\boldsymbol{v}||$. De hecho $||2\boldsymbol{v}||_0 = ||\boldsymbol{v}||_0 =$ cantidad de componentes no nulas en $\boldsymbol{v}$.

La maravillosa observación es que podemos encontrar la solución más rala de $A\boldsymbol{v} = \boldsymbol{b}$ *usando la norma* ℓ^1. Hemos "convexificado" el esqueleto ℓ^0 sobre los dos ejes. Rellenamos el esqueleto y el resultado es el diamante ℓ^1.

Productos internos y ángulos

La norma $\boldsymbol{\ell^2}$ tiene un lugar especial. Cuando escribimos $||\boldsymbol{v}||$ sin subíndice, esta es la norma a la que nos referimos. Se conecta con la geometría ordinaria de los productos internos $(\boldsymbol{v}, \boldsymbol{w}) = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{w}$ y ángulos θ entre vectores :

$$\textbf{Producto interno } = \textbf{ longitud al cuadrado} \qquad \boldsymbol{v} \cdot \boldsymbol{v} = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{v} = ||\boldsymbol{v}||^{\boldsymbol{2}} \tag{3}$$

$$\textbf{Ángulo } \boldsymbol{\theta} \textbf{ entre vectores } \boldsymbol{v} \textbf{ y } \boldsymbol{w} \qquad \boldsymbol{v}^{\mathrm{T}}\boldsymbol{w} = ||\boldsymbol{v}||\,||\boldsymbol{w}|| \cos\theta \tag{4}$$

Así $\boldsymbol{v}$ es ortogonal a $\boldsymbol{w}$ cuando $\theta = 90°$ y $\cos\theta = 0$ y $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{w} = 0$.

Esas conexiones (3) y (4) llevan a las desigualdades más importantes en matemática :

$$\textbf{Cauchy-Schwarz } |\boldsymbol{v}^{\mathrm{T}}\boldsymbol{w}| \le ||\boldsymbol{v}||\,||\boldsymbol{w}|| \quad \textbf{Desigualdad Triangular } ||\boldsymbol{v}+\boldsymbol{w}|| \le ||\boldsymbol{v}|| + ||\boldsymbol{w}||$$

Los problemas de esta sección incluyen una prueba directa de Cauchy-Schwarz. Aquí en el texto lo conectamos con la ecuación (4) vía el coseno : $|\cos\theta| \le 1$ significa que $|\boldsymbol{v}^{\mathrm{T}}\boldsymbol{w}| \le ||\boldsymbol{v}||\,||\boldsymbol{w}||$. Y esto a su vez conduce a la desigualdad triangular en la ecuación (2), conectando los lados $\boldsymbol{v}$, $\boldsymbol{w}$, y $\boldsymbol{v}+\boldsymbol{w}$ de un triángulo ordinario en n dimensiones :

$$\begin{aligned} &\text{Igualdad} && ||\boldsymbol{v}+\boldsymbol{w}||^2 = (\boldsymbol{v}+\boldsymbol{w}, \boldsymbol{v}+\boldsymbol{w}) = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{v} + \boldsymbol{v}^{\mathrm{T}}\boldsymbol{w} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{v} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} \\ &\text{Desigualdad} && ||\boldsymbol{v}+\boldsymbol{w}||^2 \le ||\boldsymbol{v}||^2 + 2\,||\boldsymbol{v}||\,||\boldsymbol{w}|| + ||\boldsymbol{w}||^2 = (||\boldsymbol{v}|| + ||\boldsymbol{w}||)^2 \end{aligned} \tag{5}$$

Esto confima nuestra intuición : la longitud de cualquier lado de un triángulo es menor que la suma de las longitudes de sus otros dos lados : $||\boldsymbol{v}+\boldsymbol{w}|| \le ||\boldsymbol{v}|| + ||\boldsymbol{w}||$. La igualdad en la norma ℓ^2 es solo posible cuando el triángulo está colapsado en un segmento y para todos los ángulos vale que $|\cos\theta| = 1$.

Productos internos y normas S

Una pregunta final sobre las normas vectoriales. ¿Es ℓ^2 la única norma que tiene conexión con los productos internos (productos punto) y los ángulos ? No hay productos punto en ℓ^1 y ℓ^∞ pero podemos encontrar otros productos internos que concuerdan con otras normas :

Elijamos una matriz simétrica y definida positiva $\boldsymbol{S}$

$$||\boldsymbol{v}||_{\boldsymbol{S}}^{\boldsymbol{2}} = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{v} \text{ da una norma para } \boldsymbol{v} \text{ en } \mathbf{R}^n \text{ (llamada la norma } \boldsymbol{S}) \tag{6}$$

$$(\boldsymbol{v}, \boldsymbol{w})_{\boldsymbol{S}} = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{w} \text{ da el producto interno } \boldsymbol{S} \text{ de } \boldsymbol{v}, \boldsymbol{w} \text{ en } \mathbf{R}^n \tag{7}$$

El producto interno $(\boldsymbol{v}, \boldsymbol{v})_{\boldsymbol{S}}$ coincide con $||\boldsymbol{v}||_{\boldsymbol{S}}^{\boldsymbol{2}}$. Tenemos una noción de ángulo a partir de (4) y desigualdades a partir de (5). La prueba está en (5) cuando en todas las nomas se incluye la matriz S.

Sabemos que cada matriz S definida positiva puede factorizarse en $A^{\mathrm{T}}A$. Así la norma S y el producto interno S de $\boldsymbol{v}$ y $\boldsymbol{w}$ son exactamente la norma ℓ^2 estándar y el producto interno estándar de $A\boldsymbol{v}$ y $A\boldsymbol{w}$.

$$(\boldsymbol{v}, \boldsymbol{w})_{\boldsymbol{S}} = \boldsymbol{v}^{\mathrm{T}} S \boldsymbol{w} = (A\boldsymbol{v})^{\mathrm{T}}(A\boldsymbol{w}) \text{ porque } S = A^{\mathrm{T}}A \tag{8}$$

Esto no es una idea impresionante pero es conveniente. Las matrices S y A "pesan" los vectores y sus longitudes. Entonces mínimos cuadrados pesados es simplemente mínimos cuadrados ordinario en esta norma pesada.

Einstein necesitaba una nueva definición de longitud y distancia en el espacio tiempo de 4 dimensiones. Lorentz propuso esta, que Einstein aceptó (c = velocidad de la luz):

$\boldsymbol{v} = (x, y, z, t)$ $\quad ||\boldsymbol{v}||^2 = \boldsymbol{x^2 + y^2 + z^2 - c^2 t^2}$ $\quad$ *¿ Es ésta una verdadera norma en* $\mathbf{R}^4$?

Normas y Productos Internos de Funciones

Una función $f(x)$ es un **"vector en un espacio de funciones"**. Esta sencilla idea le da al álgebra lineal una importancia que va más alla del espacio n dimensional $\mathbf{R}^n$. Toda la intuición asociada con la linealidad nos lleva de dimensiones finitas a dimensiones infinitas. El requerimiento fundamental para un espacio vectorial es permitir *combinaciones lineales* $c\boldsymbol{v} + d\boldsymbol{w}$ de vectores $\boldsymbol{v}$ y $\boldsymbol{w}$. Esta idea se extiende directamente a las combinaciones lineales $cf + dg$ de funciones f y g.

Es precisamente con las normas que surgen nuevas preguntas en dimensiones infinitas. Pensemos en los vectores particulares $\boldsymbol{v}_n = \left(1, \frac{1}{2}, \ldots, \left(\frac{1}{2}\right)^n, 0, 0, \ldots\right)$ con la norma ℓ_2 usual. Esos vectores se acercan entre sí debido a que $||\boldsymbol{v}_n - \boldsymbol{v}_N|| \to 0$ a medida que $n \to \infty$ y $N \to \infty$. **Para que un espacio vectorial sea "completo", cada sucesión convergente $\boldsymbol{v}_n$ debe tener un límite $\boldsymbol{v}_\infty$ en el espacio**: $||\boldsymbol{v}_n - \boldsymbol{v}_\infty|| \to 0$.

1. El espacio de vectores infinitos $\boldsymbol{v} = (v_1, \ldots, v_N, 0, 0, \ldots)$ que termina con todos ceros *no es completo.*

2. El espacio de los vectores con $||\boldsymbol{v}||^2 = |v_1|^2 + |v_2|^2 + \cdots < \infty$ *es completo.* Un vector como $\boldsymbol{v}_\infty = \left(1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots\right)$ esta incluido en este espacio, pero no termina con ceros.

Dos nombres famosos están asociados a los espacios vectoriales completos de dimensión infinita:

Un espacio de Banach es un espacio vectorial completo con una norma $||\boldsymbol{v}||$ que satisface las reglas (1) y (2)

Un espacio de Hilbert es un espacio de Banach que también tiene un producto interno con $(\boldsymbol{v}, \boldsymbol{v})$ igual a $||\boldsymbol{v}||^2$

Estos espacios son de dimensión infinita cuando los vectores tienen infinitas componentes:

$\boldsymbol{\ell^1}$ es un espacio de Banach con norma $||\boldsymbol{v}||_1 = |v_1| + |v_2| + \cdots$

$\boldsymbol{\ell^2}$ es un espacio de Hilbert porque tiene un producto interno $(\boldsymbol{v}, \boldsymbol{w}) = v_1 w_1 + v_2 w_2 + \cdots$

$\boldsymbol{\ell^\infty}$ es un espacio de Banach con norma $||\boldsymbol{v}||_\infty =$ supremo de los números $|v_1|, |v_2|, \ldots$

Nuestro interés especial está en los espacios de funciones. Los vectores pueden ser funciones $f(x)$ para $0 \le x \le 1$.

$\boldsymbol{L^1}[0, 1]$ es un espacio de Banach con $||\boldsymbol{f}||_1 = \int_0^1 |f(x)| dx$

$\boldsymbol{L^2}[0, 1]$ es un espacio de Hilbert con $(\boldsymbol{f}, \boldsymbol{g}) = \int_0^1 f(x)\, g(x)\, dx$ y $||\boldsymbol{f}||_2^2 = \int_0^1 |f(x)|^2 dx$

$\boldsymbol{L^\infty}[0, 1]$ es un espacio de Banach con $||\boldsymbol{f}||_\infty =$ supremo de $|f(x)|$.

Observe el paralelismo entre las sumas de las componentes en $\boldsymbol{\ell^1}$ y las integrales de funciones en $\boldsymbol{L^1}$. De manera similar, sumas de cuadrados en $\boldsymbol{\ell^2}$ e integrales de $|f(x)|^2$ en $\boldsymbol{L^2}$. *Sumar o integrar.*

Suavidad de funciones

Existen muchos tipos de espacios de funciones. Esta parte de la matemática es el "*análisis funcional*". Generalmente un espacio de funciones contiene todas las funciones con un nivel específico de suavidad. Un ejemplo sobresaliente es el espacio $\boldsymbol{C}[0,1]$ que contiene todas las **funciones continuas**:

f pertenece a $\boldsymbol{C}[0,1]$ y $\boldsymbol{||f||_C = \max|f(x)|}$ si $f(x)$ es continua para todo $0 \le x \le 1$.

La norma del máximo en el espacio funcional $\boldsymbol{C}$ es como la norma $\boldsymbol{\ell^\infty}$ para vectores. Podemos aumentar el nivel de suavidad a $\boldsymbol{C^1}[0,1]$ o $\boldsymbol{C^2}[0,1]$. Entonces la primera o la segunda derivada también deben ser continuas. Estos son espacios de Banach pero no espacios de Hilbert. Sus normas no provienen de productos internos: compare (9) y (10):

$$||\boldsymbol{f}||_{\boldsymbol{C^1}} = ||\boldsymbol{f}||_{\boldsymbol{C}} + \left|\left|\frac{\boldsymbol{df}}{\boldsymbol{dx}}\right|\right|_{\boldsymbol{C}} \qquad ||\boldsymbol{f}||_{\boldsymbol{C^2}} = ||\boldsymbol{f}||_{\boldsymbol{C}} + \left|\left|\frac{\boldsymbol{d^2f}}{\boldsymbol{dx^2}}\right|\right|_{\boldsymbol{C}} \tag{9}$$

Si queremos un espacio de Hilbert $\boldsymbol{H^1}$, entonces construimos sobre el espacio $\boldsymbol{L^2}$ habitual (que es $\boldsymbol{H^0}$):

$$||\boldsymbol{f}||^2_{\boldsymbol{H^1}} = ||\boldsymbol{f}||^2 + \left|\left|\frac{\boldsymbol{df}}{\boldsymbol{dx}}\right|\right|^2 \text{ y } (\boldsymbol{f},\boldsymbol{g})_{\boldsymbol{H^1}} = \int_0^1 f(x)\,g(x)\,dx + \int_0^1 \frac{df}{dx}\frac{dg}{dx}\,dx \tag{10}$$

Llevamos esta excursión salvaje sobre espacios funcionales a su final con tres ejemplos.

1. El vector infinito $\boldsymbol{v} = \left(1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots\right)$ está en $\boldsymbol{\ell^2}$ y $\boldsymbol{\ell^\infty}$. Pero no está en $\boldsymbol{\ell^1}$. La suma de sus componentes es infinita.
2. Una función escalón está en $\boldsymbol{L^1}$, $\boldsymbol{L^2}$ y $\boldsymbol{L^\infty}$, pero no en $\boldsymbol{C}$. La función tiene un salto.
3. La función rampa máx $(0, x)$ está en $\boldsymbol{C}$ y $\boldsymbol{H^1}$ pero no en $\boldsymbol{C^1}$. La pendiente tiene un salto.

Normas de matrices: la norma de Frobenius

Un espacio de matrices cumple todas las reglas de un espacio vectorial. Entonces una norma matricial $||\boldsymbol{A}||$ debe cumplir las tres reglas de una norma vectorial, y también una regla nueva cuando A multiplica a B:

$$||\boldsymbol{A}|| > \boldsymbol{0} \text{ si } A \text{ no es la matriz cero} \tag{11}$$

$$||\boldsymbol{cA}|| = |\boldsymbol{c}|\,||\boldsymbol{A}|| \text{ y } ||\boldsymbol{A}+\boldsymbol{B}|| \le ||\boldsymbol{A}|| + ||\boldsymbol{B}|| \tag{12}$$

- **Nueva regla para una norma matricial** $||\boldsymbol{AB}|| \le ||\boldsymbol{A}||\,||\boldsymbol{B}||$ (13)

Necesitamos verificar (13) para la norma de Frobenius, que trata a las matrices como vectores largos.

$$||\boldsymbol{A}||_{\boldsymbol{F}}^{\mathbf{2}} = (\textbf{Norma de Frobenius de } \boldsymbol{A})^{\mathbf{2}} = |a_{11}|^2 + \cdots + |a_{1n}|^2 + \cdots + |a_{mn}|^2. \tag{14}$$

La norma de Frobenius es la norma ℓ^2 (norma Euclídea) de un vector de mn componentes. Así (11) y (12) son ciertas. Y cuando AB es el vector columna $\boldsymbol{a}$ por el vector fila $\boldsymbol{b}^{\mathrm{T}}$, en otras palabras AB es la matriz de rango uno $\boldsymbol{ab}^{\mathrm{T}}$, la desigualdad en norma (13) es justamente una igualdad $||\boldsymbol{ab}^{\mathrm{T}}||_{\boldsymbol{F}} = ||\boldsymbol{a}||_{\boldsymbol{F}}||\boldsymbol{b}^{\mathrm{T}}||_{\boldsymbol{F}}$:

$$\boldsymbol{ab}^{\mathrm{T}} = \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} \begin{bmatrix} b_1 & \cdots & b_p \end{bmatrix} \quad \text{cumple} \quad ||\boldsymbol{ab}^{\mathrm{T}}||_{\boldsymbol{F}}^{\mathbf{2}} = \begin{matrix} |a_1|^2\left(|b_1|^2 + \cdots + |b_p|^2\right) \\ + \cdots + \\ |a_m|^2\left(|b_1|^2 + \cdots + |b_p|^2\right) \end{matrix} = ||\boldsymbol{a}||_{\boldsymbol{F}}^{\mathbf{2}}\, ||\boldsymbol{b}||_{\boldsymbol{F}}^{\mathbf{2}}. \tag{15}$$

Esto nos lleva a una primera prueba de $||\boldsymbol{AB}||_{\boldsymbol{F}} \leq ||\boldsymbol{A}||_{\boldsymbol{F}}\, ||\boldsymbol{B}||_{\boldsymbol{F}}$, usando que AB es una suma de matrices de rango 1.

$$\begin{aligned} ||\boldsymbol{AB}||_{\boldsymbol{F}} &= ||\boldsymbol{a}_1\boldsymbol{b}_1^{\mathrm{T}} + \cdots + \boldsymbol{a}_n\boldsymbol{b}_n^{\mathrm{T}}||_F \text{ por la multiplicacion columna-fila} \\ &\leq ||\boldsymbol{a_1}\boldsymbol{b_1^{\mathrm{T}}}||_F + \cdots + ||\boldsymbol{a_n}\boldsymbol{b_n^{\mathrm{T}}}||_F \text{ por la desigualdad triangular (12)} \\ &= ||\boldsymbol{a}_1||_F||\boldsymbol{b}_1||_F + \cdots + ||\boldsymbol{a}_n||_F||\boldsymbol{b}_n||_F \text{ por la ecuación (15)} \\ &\leq \left(||\boldsymbol{a}_1||_F^2 + \cdots + ||\boldsymbol{a}_n||_F^2\right)^{1/2}\left(||\boldsymbol{b}_1||_F^2 + \cdots + ||\boldsymbol{b}_n||_F^2\right)^{1/2} \text{ por Cauchy-Schwarz} \\ &= ||\boldsymbol{A}||_{\boldsymbol{F}}\, ||\boldsymbol{B}||_{\boldsymbol{F}} \text{ por la definición (14) de la norma de Frobenius} \end{aligned}$$

En los problemas de esta sección hay una prueba diferente y más rápida multiplicando AB vía filas por columnas.

Cuando Q es una matriz ortogonal, sabemos que $Q\boldsymbol{x}$ tiene la misma norma ℓ^2 que $\boldsymbol{x}$:

$$\boldsymbol{Q}\textbf{ Ortogonal} \quad ||\boldsymbol{Qx}||_{\mathbf{2}} = ||\boldsymbol{x}||_{\mathbf{2}} \qquad \boldsymbol{Q}\textbf{ multiplica las columnas de } \boldsymbol{B} \quad ||\boldsymbol{QB}||_{\boldsymbol{F}} = ||\boldsymbol{B}||_{\boldsymbol{F}}$$

Esto conecta la norma de Frobenius de $A = U\Sigma V^{\mathrm{T}}$ con sus valores singulares en Σ:

$$||\boldsymbol{A}||_{\boldsymbol{F}} = ||\boldsymbol{U\Sigma V^{\mathrm{T}}}||_{\boldsymbol{F}} = ||\boldsymbol{\Sigma V^{\mathrm{T}}}||_{\boldsymbol{F}} = ||\boldsymbol{\Sigma}||_{\boldsymbol{F}} = \sqrt{\boldsymbol{\sigma_1^2} + \cdots + \boldsymbol{\sigma_r^2}} \tag{16}$$

He aquí otra manera de llegar a esta buena fórmula para la norma de Frobenius

Multiplicar $A^{\mathrm{T}}A$ coloca todos los números $|a_{ij}|^2$ sobre la diagonal principal

$$||\boldsymbol{A}||_{\boldsymbol{F}}^{\mathbf{2}} = \textbf{traza de } \boldsymbol{A^{\mathrm{T}}A} = \textbf{suma de autovalores } = \boldsymbol{\sigma_1^2} + \cdots + \boldsymbol{\sigma_r^2} \tag{17}$$

La norma de Frobenius (al cuadrado) de A es fácil de calcular: solo hay que elevar los elementos al cuadrado y sumar.

La desigualdad de $||AB|| \leq ||A||\,||B||$ valdrá automáticamente para las normas matriciales que veremos a continuación.

Normas matriciales $||A||$ a partir de normas vectoriales $||v||$

Comencemos con una norma cualquiera $||v||$ para los vectores en $\mathbf{R}^n$. Cuando comparamos $||Av||$ con $||v||$, esto mide el factor de crecimiento: el aumento o la disminución de tamaño que se produce cuando multiplicamos por A. Si elegimos el vector v con el *mayor factor de crecimiento*, obtenemos una norma matricial importante $||A||$.

Norma vectorial lleva a una norma matricial

$$\boxed{||A|| = \max_{v \neq 0} \frac{||Av||}{||v||} = \text{mayor factor de crecimiento}} \tag{18}$$

El mayor cociente $||A||$ automáticamente satisface todas las condiciones de una norma matricial porque $||v||$ cumple con todas las condiciones de una norma vectorial. La matriz identidad tendrá $||I|| = 1$ porque su factor de crecimiento es siempre $||Iv||/||v|| = 1$. Lo importante de (18) es que $||Av|| \leq ||A||\,||v||$ porque $||A||$ es el mayor valor alcanzado por $||Av||/||v||$. Entonces $||ABv|| \leq ||A||\,||Bv|| \leq ||A||\,||B||\,||v||$. Por lo tanto $||AB|| \leq ||A||\,||B||$.

Pensamos a $||v||_2$, $||v||_1$ y $||v||_\infty$ como las normas vectoriales importantes. Entonces (19), (20), (21) producen las tres normas matriciales $||A||_2$ y $||A||_1$ y $||A||_\infty$. Para todas ellas se cumple que $||AB|| \leq ||A||\,||B||$. Para una matriz A dada, ¿ cómo calculamos esas tres normas ? es decir ¿ cómo maximizamos el cociente $||Av||/||v||$?

$$\boxed{\begin{aligned}
&\textbf{norma } \ell^2 \quad ||A||_2 = \textbf{mayor valor singular } \sigma_1 \text{ de } A && (19)\\
&\textbf{norma } \ell^1 \quad ||A||_1 = \textbf{mayor norma } \ell^1 \textbf{ de las columnas de } A && (20)\\
&\textbf{norma } \ell^\infty \quad ||A||_\infty = \textbf{mayor norma } \ell^1 \textbf{ de las filas de } A && (21)
\end{aligned}}$$

Este libro enfatiza $||A||_2$ = mayor cociente $||Av||_2/||v||_2 = \sigma_1$. Esto se sigue de $A = U\Sigma V^{\mathrm{T}}$, porque las matrices ortogonales U y V^{T} no tienen efecto sobre las normas ℓ^2. Esto nos lleva a la matriz diagonal Σ que tiene norma ℓ^2 igual a σ_1. Observe: $A^{\mathrm{T}}A$ tiene norma σ_1^2 (de nuevo U y V no tienen efecto).

Las tres normas matriciales tienen dos conexiones especialmente hermosas :

$$\boxed{||A||_\infty = ||A^{\mathrm{T}}||_1 \qquad ||A||_2^2 \leq ||A||_1\,||A||_\infty.} \tag{22}$$

Las filas de A son las columnas de A^{T}. Así $||A||_\infty = ||A^{\mathrm{T}}||_1$ proviene directamente de (20)-(21).

Para la desigualdad (22) que involucra las tres normas, miramos el primer vector singular v de A. Para ese vector $A^{\mathrm{T}}Av = \sigma_1^2 v$. Tomamos la norma ℓ_1 de este v particular y usamos $||A||_\infty = ||A^{\mathrm{T}}||_1$:

$$\sigma_1^2\,||v||_1 = ||A^{\mathrm{T}}Av||_1 \leq ||A^{\mathrm{T}}||_1\,||Av||_1 \leq ||A||_\infty\,||A||_1\,||v||_1.$$

Como $\sigma_1 = ||A||_2$, esto nos dice que $||A||_2^2 \leq ||A||_\infty\,||A||_1$.

La norma nuclear

$||A||_{\textbf{nuclear}}$ se obtiene directamente de los valores singulares de A. También se la llama la **norma de la traza**. Junto con $\boldsymbol{\ell^2}$ y Frobenius, $||A||_N$ comienza con $A = U\Sigma V^{\mathrm{T}}$ (la SVD). Esas tres normas no están afectadas por U y V ("invarianza unitaria"). Tomamos simplemente las normas $\boldsymbol{\ell^1}$ y $\boldsymbol{\ell^2}$ y $\boldsymbol{\ell^\infty}$ del vector $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_r)$ en la diagonal principal de Σ:

$$\boxed{||A||_{\text{nuclear}} = \boldsymbol{\sigma_1 + \cdots + \sigma_r} \qquad ||A||_F^2 = \boldsymbol{\sigma_1^2 + \cdots + \sigma_r^2} \qquad ||A||_2 = \boldsymbol{\sigma_1}}$$

En III.4, la norma nuclear es la clave para completar la matriz, cuando faltan datos. Un hecho destacable: $||A||_N$ *es el valor mínimo de* $||U||_F\,||V||_F$ *sujeto a* $UV = A$. Y un hecho relacionado pero mucho más fácil: $||A^{\mathrm{T}}A||_N = ||A||_F^2$.

Observe cómo $||A||_\infty = \text{máx}||A\boldsymbol{v}||_\infty/||\boldsymbol{v}||_\infty$ es completamente diferente de $|||A|||_\infty = \text{máx}|a_{ij}|$. Llamamos a $|||A|||_\infty$ *la norma del máximo elemento* porque $|||A - B|||_\infty$ es pequeña cuando todo elemento (cada pixel) a_{ij} en A está cerca de b_{ij} en B. Entonces $||A - B||_\infty$ también es pequeña.

Ejemplo 1 $A = \begin{bmatrix} 1 & 2 \\ 3 & 6 \end{bmatrix}$ tiene $||A||_2 = \boldsymbol{\sqrt{50}}$ porque $A^{\mathrm{T}}A = \begin{bmatrix} 10 & 20 \\ 20 & 40 \end{bmatrix}$ tiene $\lambda_1 = 50$.

Las normas $\boldsymbol{\ell^1}$ y $\boldsymbol{\ell^\infty}$ son $\boldsymbol{||A||_1 = 8}$ (suma de columnas) y $\boldsymbol{||A||_\infty = 9}$ (suma de filas).

$A = \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix}$ tiene $\boldsymbol{||A||_2 = \sqrt{10}}$, $\boldsymbol{||A||_1 = 4}$ (columna 2) y $\boldsymbol{||A||_\infty = 3}$ (filas).

La norma $\boldsymbol{\ell^2}$ es $\sqrt{10}$ porque $A^{\mathrm{T}}A = \begin{bmatrix} 2 & 4 \\ 4 & 8 \end{bmatrix}$ tiene autovalores 0 y 10. Y $10 < (4)(3)$.

Importante ¡El mayor autovalor $\boldsymbol{|\lambda|_{\text{máx}}}$ de $\boldsymbol{A}$ no está en nuestra lista de normas matriciales!

El radio espectral

Ese número $\boldsymbol{|\lambda|_{\text{max}} = \text{max}\,|\lambda_i|}$ falla los tres requerimientos principales para ser una norma. A y B pueden tener todos autovalores nulos aún siendo matrices *no* nulas (vea las A y B más abajo). Los tests en $A + B$ y AB (desigualdad triangular y $||AB|| \le ||A||\,||B||$) también fallan para el mayor autovalor. *Todas las normas satisfacen* $||A|| \ge |\lambda|_{\text{máx}}$.

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \qquad \begin{aligned} \lambda_{\text{max}}(A + B) &= 1 > \lambda_{\text{max}}(A) + \lambda_{\text{max}}(B) = (0) + (0) \\ \lambda_{\text{max}}(AB) &= 1 > \lambda_{\text{max}}(A) \times \lambda_{\text{max}}(B) = (0) \times (0) \end{aligned}$$

Este número $|\lambda|_{\text{máx}}$ es el "**radio espectral**". No es una norma pero es importante por esta razón: $\boldsymbol{||A^n|| \to 0}$ **exactamente cuando** $\boldsymbol{|\lambda|_{\text{max}} < 1}$.

Cuando multiplicamos una y otra vez por una matriz A (como ocurre con las cadenas de Markov en la Sección V.6) el mayor autovalor $|\lambda|_{\text{máx}}$ de A comienza a dominar. Esta es la base del "método de las potencias" para calcular $|\lambda|_{\text{máx}}$.

Problemas de la Sección I.11

1 Muestre directamente este hecho sobre las normas vectoriales ℓ^1 y ℓ^2 y ℓ^∞ : $||\boldsymbol{v}||_2^2 \leq ||\boldsymbol{v}||_1 \, ||\boldsymbol{v}||_\infty$.

2 Pruebe la desigualdad de Cauchy-Schwarz $|\boldsymbol{v}^{\mathrm{T}}\boldsymbol{w}| \leq ||\boldsymbol{v}||_2 \, ||\boldsymbol{w}||_2$ para vectores en $\mathbf{R}^n$. Puede verificar y usar esta identidad :

$$0 \leq \left(\boldsymbol{v} - \frac{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}}\boldsymbol{w}, \boldsymbol{v} - \frac{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}}\boldsymbol{w} \right) = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{v} - \frac{|\boldsymbol{v}^{\mathrm{T}}\boldsymbol{w}|^2}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}}.$$

3 Muestre que siempre $||\boldsymbol{v}||_2 \leq \sqrt{n}\, ||\boldsymbol{v}||_\infty$. También pruebe que $||\boldsymbol{v}||_1 \leq \sqrt{n}\, ||\boldsymbol{v}||_2$ eligiendo un vector apropiado $\boldsymbol{w}$ y aplicando la desigualdad Cauchy-Schwarz.

4 Las normas ℓ^p y ℓ^q son "*duales*" si $p^{-1} + q^{-1} = 1$. Las normas vectoriales ℓ^1 y ℓ^∞ son duales (y ℓ^2 es auto-dual). La desigualdad de Hölder extiende a la de Cauchy-Schwarz para todos esos pares duales. ¿ Qué dice para $p = 1$ y $q = \infty$?

Desigualdad de Hölder $\quad |\boldsymbol{v}^{\mathrm{T}}\boldsymbol{w}| \leq ||\boldsymbol{v}||_{\boldsymbol{p}} \, ||\boldsymbol{w}||_{\boldsymbol{q}}$ cuando $\boldsymbol{p^{-1} + q^{-1} = 1}$.

5 ¿ Qué reglas deben cumplir los "*productos internos*" de los vectores $\boldsymbol{v}$ y $\boldsymbol{w}$?

6 La primera página de I.11 muestra *bolas unitarias* para las normas ℓ^1 y ℓ^2 y ℓ^∞. Esos son los tres conjuntos de vectores $\boldsymbol{v} = (v_1, v_2)$ con $||\boldsymbol{v}||_1 \leq 1, ||\boldsymbol{v}||_2 \leq 1, ||\boldsymbol{v}||_\infty \leq 1$. *Las bolas unitarias son siempre convexas debido a la desigualdad triangular para normas vectoriales* :

Si $||\boldsymbol{v}|| \leq 1$ y $||\boldsymbol{w}|| \leq 1$ muestre que $||\frac{\boldsymbol{v}}{2} + \frac{\boldsymbol{w}}{2}|| \leq 1$.

7 Una prueba corta de $||AB||_F \leq ||A||_F \, ||B||_F$ comienza con la multiplicación desde el punto de vista filas por columnas :

$|(AB)_{ij}|^2 \leq ||$fila i de $A||^2 \, ||$columna j de $B||^2$ es la desiguladad de Cauchy-Schwarz

Sume ambos lados para todo i y j para mostrar que $||AB||_F^2 \leq ||A||_F^2 \, ||B||_F^2$.

8 Verifique que $||AB||_F \leq ||A||_F \, ||B||_F$ para $A = B = I$ y $A = B =$ "matriz de todos unos".

9 (Conjetura) Las únicas matrices con $||AB||_F = ||A||_F \, ||B||_F$ y sin elementos nulos tienen rango uno y son de la forma $A = \boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}$ y $B = \boldsymbol{v}\boldsymbol{w}^{\mathrm{T}}$ con un vector compartido $\boldsymbol{v}$.

10 El espacio de las matrices de m por n con la norma de Frobenius es de hecho un espacio de Hilbert: **tiene un producto interno $(\boldsymbol{A}, \boldsymbol{B}) = \mathbf{traza}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{B})$**. Muestre que $||\boldsymbol{A}||_{\boldsymbol{F}}^{\boldsymbol{2}} = (\boldsymbol{A}, \boldsymbol{A})$.

11 ¿ Por qué es (21) una fórmula cierta para $||A||_\infty$? ¿ Qué vector $\boldsymbol{v}$ con componentes ± 1 tiene $||A\boldsymbol{v}||_\infty = ||A||_\infty$?

12 Suponga que A, B, y AB son de m por n, n por p, y m por p. La "norma del máximo" de A es el mayor elemento : $|||\boldsymbol{A}|||_\infty = \mathbf{max}|\boldsymbol{a_{ij}}|$.

Muestre que $|||\boldsymbol{AB}|||_\infty \leq \boldsymbol{n}\, |||\boldsymbol{A}|||_\infty \, |||\boldsymbol{B}|||_\infty$ (esto es falso sin el factor n).

Reescríbalo en la forma $(\sqrt{mp}\, |||AB|||_\infty) \leq (\sqrt{mn}\, |||A|||_\infty)(\sqrt{np}\, |||B|||_\infty)$. Reescalar usando raíces cuadradas da una verdadera norma matricial.

I.12. Factorización de matrices y tensores : positivos y ralos

Esta sección amplía el campo de las factorizaciones de matrices de datos (y se extiende a tensores). Hasta ahora nos hemos enfocado completamente en la SVD. $\boldsymbol{A = U\Sigma V^{\mathrm{T}}}$ dio la factorización perfecta para el Análisis de Componentes Principales (perfecto hasta que surgen complicaciones en los problemas ralos o nonegatividad o datos tensoriales). Para muchas aplicaciones estos problemas son importantes. Entonces debemos considerar a la SVD como un primer paso pero *no el último*.

He aquí factorizaciones de A y T con nuevas propiedades importantes :

Matrices no negativas	mín $\|\|\boldsymbol{A - UV}\|\|_{\boldsymbol{F}}^2$ con $U \geq 0$ y $V \geq 0$
Ralas y no negativas	mín $\|\|\boldsymbol{A - UV}\|\|_{\boldsymbol{F}}^2 + \boldsymbol{\lambda}\|\|\boldsymbol{UV}\|\|_{\boldsymbol{N}}$ con $U \geq 0$ y $V \geq 0$
Descomposición tensorial CP	mín $\|\|\boldsymbol{T} - \sum_{i=1}^{R} \boldsymbol{a_i} \circ \boldsymbol{b_i} \circ \boldsymbol{c_i}\|\|$

Trabajaremos con matrices A y luego tensores T. Una matriz es solo un tensor de orden dos.

Para calcular una factorización $A = UV$, introducimos una **iteración alternada** simple. **Actualice $\boldsymbol{U}$ con $\boldsymbol{V}$ fijo, y luego actualice $\boldsymbol{V}$ con $\boldsymbol{U}$ fijo**. Cada medio paso es rápido porque es lineal (si el otro factor está fijo). Esta idea se aplica a la SVD ordinaria si incluimos la matriz diagonal Σ con U. El algoritmo es simple y a menudo efectivo. En la Sección III.4 se hará aún mejor.

Esta idea de UV también se adapta al famoso **algoritmo de $\boldsymbol{k}$-medias** sobre grafos presentado en la Sección IV.7. El problema es distribuir los n vectores $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ en r grupos (*clusters*). Si $\boldsymbol{a}_k$ está en el grupo alrededor de $\boldsymbol{u}_j$, *la afirmación* $\boldsymbol{a}_k \approx \boldsymbol{u}_j$ *se expresa en la columna* k *de* $\boldsymbol{A \approx UV}$. Entonces la columna k de V es la columna j de la matriz identidad de r por r.

Factorización de matrices no negativas (NMF)

El objetivo de NMF es aproximar a una matriz no negativa $A \geq 0$ por un producto de menor rango UV de dos matrices no negativas $U \geq 0$ y $V \geq 0$. El propósito de menor rango es la simplicidad. El propósito de la no negatividad (sin elementos negativos) es producir números que *tengan un significado*. Un peso, volumen, cuenta o probabilidad negativa es erróneo desde el principio.

Pero la no negatividad puede ser difícil. Cuando $A \geq 0$ es simétrica y definida positiva, quisiéramos una matriz $B \geq 0$ que cumpla $B^{\mathrm{T}}B = A$. Muy a menudo no existe tal matriz. (La matriz $A = A^{\mathrm{T}}$ con las diagonales constantes $1 + \sqrt{5}, 2, 0, 0, 2$ es un ejemplo de 5×5). Estamos obligados a aceptar la matriz $B^{\mathrm{T}}B$ (con $B \geq 0$) que esté más cercana a A (cuando $A \geq 0$). La pregunta es cómo encontrar B. El caso no simétrico, probablemente no cuadrado, busca U y V.

Lee y Seung se enfocaron en la NMF en una carta a *Nature* **401** (1999) 788–791.

El problema básico es claro. Raleza (*sparsity*) y no negatividad son propiedades muy valiosas. Para un vector o matriz ralos, los pocos valores no nulos tienen significado. 1 000 o 100 000 números individuales no pueden comprenderse por separado. Y generalmente ocurre que los números son naturalmente no negativos. Pero los vectores singulares en la SVD casi siempre tienen muchos componentes pequeños con signos mezclados. En problemas prácticos, debemos estar dispuestos a sacrificar la ortogonalidad de U y V. Estas son hermosas propiedades, pero el ensayo de Lee-Seung fomenta el valor de un PCA ralo y la ausencia de números negativos.

Estos nuevos objetivos requieren de nuevas factorizaciones de A.

NMF	Encontrar **matrices no negativas U y V** tales que $A \approx UV$	(1)
SPCA	Encontrar **matrices B y C** ralas y de rango bajo tal que $A \approx BC$.	(2)

Recordemos primero el significado y el propósito de una factorización. $A = BC$ expresa cada columna de A como una combinación de las columnas de B. Los coeficientes de esa combinación están en una columna de C. Así cada columna $\boldsymbol{a}_j$ de A es la aproximación $c_{1j}\boldsymbol{b}_1 + \cdots + c_{nj}\boldsymbol{b}_n$. Una buena elección de BC significa que esta suma es casi exacta.

Si B tiene menos columnas que A, esto es *reducción de dimensiones lineal*. Esto es fundamental para la compresión, para la selección de variables y para visualización. En muchos de esos problemas se puede suponer que el ruido es Gaussiano. Entonces la norma de Frobenius $||A - BC||_F$ es la medida natural del error de aproximación. He aquí un excelente ensayo que describe dos aplicaciones importantes, y un trabajo reciente con algoritmos y referencias.

N. Gillis, *The Why and How of Nonnegative Matrix Factorization*, arXiv : 1401.5226.

L. Xu, B. Yu, and Y. Zhang, *An alternating direction and projection algorithm for structure-enforced matrix factorization*, Computational Optimization Appl. **68** (2017) 333-362.

Extracción de rasgos faciales

Cada vector columna de la matriz de datos A representa un rostro. Sus componentes son las intensidades de los pixeles en la imagen, así $A \geq 0$. El objetivo es encontrar unos pocos "rostros básicos" en B, para que sus combinaciones estén cerca de los muchos rostros en A. Podemos esperar que algunas variaciones en la geometría de los ojos, nariz y boca permitan una reconstrucción cercana de la mayoría de los rostros. El desarrollo de los *autorostros* (*eigenfaces*) de Turk y Pentland encuentra un conjunto de rostros básicas y la factorización de matrices $A \approx BC$ es otra forma efectiva de hacerlo.

Minería de textos y clasificación de documentos

Ahora cada columna de A representa un documento. Cada fila de A representa una palabra. Una construcción simple (en general no es la mejor : ignora el orden de las palabras) es una matriz no negativa rala. Para clasificar los documentos en A, buscamos factores no negativos ralos :

$$\text{Documento} \quad \boldsymbol{a}_j \approx \sum(\text{importancia } c_{ij})\,(\text{tema } \boldsymbol{b}_i) \tag{3}$$

Como $B \geq 0$, cada vector de tema $\boldsymbol{b}_i$ puede interpretarse como un documento. Dado que $C \geq 0$, combinamos, pero *no restamos* esos documentos temáticos (*topic documents*). Así NMF identifica

los temas y clasifica todo el conjunto de documentos con respecto a esos temas. Otros métodos relacionados son el "analisis semántico latente" e indexación.

Note que NMF es un problema NP-difícil, a diferencia de la SVD. Incluso las soluciones exactas $A = BC$ no son siempre únicas. Es más, la cantidad de temas (columnas de B) es desconocida.

Condiciones óptimas para U y V no negativas

Dado $A \geq 0$, he aquí las condiciones para $U \geq 0$ y $V \geq 0$ que minimizan $||A - UV||_F^2$:

$$\begin{aligned} Y &= UVV^{\mathrm{T}} - AV^{\mathrm{T}} \geq 0 \quad \text{con } Y_{ij} \text{ o } U_{ij} = 0 \text{ para todo } i, j \\ Z &= U^{\mathrm{T}}UV - U^{\mathrm{T}}A \geq 0 \quad \text{con } Z_{ij} \text{ o } V_{ij} = 0 \text{ para todo } i, j \end{aligned} \tag{4}$$

Esas últimas condiciones ya sugieren que U y V pueden resultar ralas.

Cálculo de factores : métodos básicos

Muchos algoritmos han sido sugeridos para calcular U y V y B y C. Una idea central es la **factorización alternada** : mantenga un factor fijo y optimice el otro factor. Mantenga ese fijo, optimice el primer factor, y repita. Con la norma de Frobenius cada paso es una forma de cuadrados mínimos. Este es un enfoque natural y generalmente da buenos resultados. *Pero la convergencia a los factores óptimos no es un hecho seguro.* Podemos esperar que haya un mayor desarrollo de la teoría de optimización. Y hay una mejora ya establecida de este método que se desarrollará en la Sección III.4. : **método de multiplicadores de direcciones alternadas (ADMM, por sus sigla en inglés)**.

Este algoritmo ADMM usa un término de penalización y dualidad para acelerar la convergencia.

Componentes principales ralas

Muchas aplicaciones permiten tanto números negativos como positivos. No estamos contando o construyendo objetos reales. En finanzas podemos comprar o vender. En otras aplicaciones el punto cero no tiene valor intrínseco. La temperatura cero es subjetiva, dependiendo de si hablamos de grados centígrados o Fahrenheit.

La cantidad de componentes distintas de cero suele ser importante. Esa es la dificultad con los vectores singulares $\boldsymbol{u}$ y $\boldsymbol{v}$ en la SVD. Están llenos de valores no nulos como en mínimos cuadrados. No podemos comprar cantidades pequeñas de un recurso gigante debido a los costos de transacción. Si conocemos 500 genes que afectan la evolución de un paciente, no podemos abordarlos individualmente a cada uno. Para entenderlos y poder decidir como proceder, la cantidad de variables de decisión distintas de cero debe estar bajo control.

Una posibilidad es quitar las componentes muy pequeñas de los $\boldsymbol{u}$ y los $\boldsymbol{v}$. Pero si queremos verdadero control, es conveniente una construcción directa de vectores ralos. Una buena cantidad de algoritmos han sido propuestos.

H. Zou, T. Hastie, R. Tibshirani, *Sparse principal component analysis*, J. Computational and Graphical Statistics **15** (2006) 265–286. See https://en.wikipedia.org/wiki/Sparse_PCA

El PCA ralo parte de una matriz de datos A o una matriz de covarianza muestral S (semi)definida positiva. Dada S, una idea natural es incluir Card $(\boldsymbol{x})$ = número de componentes no nulas en un término de penalización o una restricción en $\boldsymbol{x}$:

$$\underset{||\boldsymbol{x}||=1}{\text{Maximizar}} \ \boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x} - \rho \,\text{Card}\,(\boldsymbol{x}) \quad \text{o} \quad \underset{||\boldsymbol{x}||=1}{\text{Maximizar}} \ \boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x} \ \text{ sujeto a } \ \text{Card}\,(\boldsymbol{x}) \le k. \tag{5}$$

Pero la cardinalidad de $\boldsymbol{x}$ no es la mejor cantidad para los algoritmos de optimización.

Otra posibilidad es la *programación semidefinida*, que se trata brevemente en la Sección VI.3. El vector desconocido $\boldsymbol{x}$ se vuelve una matriz simétrica X desconocida. Las desigualdades como $\boldsymbol{x} \ge 0$ (que significa que cada $x_i \ge 0$) se reemplaza por $X \ge 0$ (X debe ser semidefinida positiva). La raleza se logra al incluir una penalización $\boldsymbol{\ell^1}$ sobre la matriz X desconocida. Adelantándonos a IV.5, esa penalización usa la *norma nuclear* $||X||_N$ *: la suma de sus valores singulares* σ_i.

La conexión entre $\boldsymbol{\ell^1}$ y raleza apareció en las figuras al inicio de la Sección I.11. **La minimización $\boldsymbol{\ell^1}$ daba la solución rala** $\boldsymbol{x} = (0, \frac{1}{4})$. Ese cero podía haber parecido accidental o insignificante, para este vector de solo dos componentes. Pero al contrario, ese cero es un hecho importante. Para matrices, reemplazamos la norma $\boldsymbol{\ell^1}$ por la norma nuclear $||X||_N$.

Una penalización en $||\boldsymbol{x}||_1$ o $||\boldsymbol{X}||_N$ produce vectores $\boldsymbol{x}$ ralos y matrices $\boldsymbol{X}$ ralas.

Al final, para los vectores $\boldsymbol{x}$ ralos, nuestro algoritmo debe *seleccionar* las variables importantes. Esta es la gran propiedad de la optimización $\boldsymbol{\ell^1}$. Es la clave para LASSO:

$$\boxed{\textbf{LASSO} \qquad \text{Minimizar } \ ||A\boldsymbol{x} - \boldsymbol{b}||^2 + \lambda \sum_{1}^{n} |x_k|} \tag{6}$$

Encontrar ese mínimo eficientemente es un triunfo de la optimización no lineal. El algoritmo ADMM y el algoritmo de Bergman se desarrollarán en la Sección III.4.

Una observación sobre LASSO: el $\boldsymbol{x}^*$ óptimo no tendrá más componentes no nulas que la cantidad de muestras. Agregar una penalización $\boldsymbol{\ell^2}$ produce una "red elástica" sin esta desventaja. Esta regresión $\boldsymbol{\ell^1}$+*ridge* puede resolverse tan rápido como cuadrados mínimos.

$$\boxed{\textbf{Red elástica} \qquad \text{Minimizar } \ ||A\boldsymbol{x} - \boldsymbol{b}||_2^2 + \lambda||\boldsymbol{x}||_1 + \beta||\boldsymbol{x}||_2^2} \tag{7}$$

La Sección III.4 presentará el algoritmo **ADMM** que separa $\boldsymbol{\ell^1}$ de $\boldsymbol{\ell^2}$ y agrega una penalización usando multiplicadores de Lagrange y dualidad. Esa combinación es poderosa.

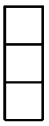
1. R. Tibshirani, Regression shrinkage and selection via the Lasso, *Journal of the Royal Statistical Society*, Series B **58** (1996) 267–288.

2. H. Zou and T. Hastie, Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society*, Series B **67** (2005) 301–320.

Tensores

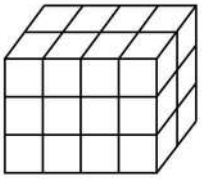
Un vector columna es un tensor de orden uno. Una matriz es un tensor de orden dos. Entonces, un tensor $\boldsymbol{T}$ de orden tres tiene **elementos $\boldsymbol{T_{ijk}}$ con tres índices** : número de fila, número de columna y "número de tubo". Las secciones (*slices*) de $\boldsymbol{T}$ son cortes bidimensionales, así el tensor de orden tres de abajo tiene tres secciones horizontales, cuatro secciones laterales y dos secciones frontales. Las filas y columnas y tubos son las **fibras de $\boldsymbol{T}$**, con solo un índice variando.

Podemos apilar matrices de m por n (p de ellas) en un tensor de orden 3. Y podemos apilar m por n por p tensores en un tensor de orden 4 = arreglo de orden 4.



vector
$\boldsymbol{x}$ **en** $\mathbf{R^3}$

matriz
$\boldsymbol{A}$ **en** $\mathbf{R^{3 \times 4}}$



tensor
$\boldsymbol{T}$ **en** $\mathbf{R^{3 \times 4 \times 2}}$

Ejemplo 1 : una imagen a color es un tensor con 3 secciones

Una imagen en blanco y negro es simplemente una matriz de píxeles. Los números de la matriz son los tonos de la escala de grises de cada píxel. Normalmente esos números están entre cero (negro) y 255 (blanco). Cada elemento de A tiene $2^8 = 256$ tonos de la escala de grises posibles.

Una imagen a color es un tensor. Tiene 3 secciones correspondientes a rojo, verde y azul (RGB). Cada sección muestra la densidad de uno de los colores RGB. Procesar este tensor T (por ejemplo en aprendizaje profundo : Sección VII.2) no es más difícil que una imagen en blanco y negro.

Ejemplo 2 : la derivada $\partial w/\partial A$ de $w = Av$

Este es un tensor que no esperábamos. La matriz A de $m \times n$ contiene "pesos" a optimizar en aprendizaje profundo. Esa matriz multiplica un vector $\boldsymbol{v}$ para producir $\boldsymbol{w} = A\boldsymbol{v}$. Así el algoritmo para optimizar A (Secciones VI.4 a VII.3) involucra la derivada de cada resultado w_i con respecto a cada peso A_{jk}. Así tenemos tres índices i, j, k.

En la multiplicación de matrices, sabemos que la fila j de A no afecta la fila i de $\boldsymbol{w} = A\boldsymbol{v}$. Entonces la fórmula de las derivadas incluye el símbolo δ_{ij}, que es 1 si $i = j$ y 0 en caso contrario. Si usamos notacion propia de tensores ese símbolo se convierte en δ^i_j (nuestra autoridad en materia de tensores es Pavel Grinfeld). Las derivadas de la función lineal $\boldsymbol{w} = A\boldsymbol{v}$ con respecto a los pesos A_{jk} están en T :

$$\boxed{\boldsymbol{T_{ijk}} = \frac{\boldsymbol{\partial w_i}}{\boldsymbol{\partial A_{jk}}} = \boldsymbol{v_k}\,\boldsymbol{\delta_{ij}} \qquad \text{así } \boldsymbol{T_{111}} = \boldsymbol{v_1} \text{ y } \boldsymbol{T_{122}} = \boldsymbol{0}} \tag{8}$$

La Sección VII.3 tiene un ejemplo de $2 \times 2 \times 2$. Este tensor $T_{ijk} = v_k \delta_{ij}$ es de interés especial:

1. Las secciones $k =$ constante son múltiplos v_k de la matriz identidad.
2. La función clave del aprendizaje profundo conecta cada capa de una red neuronal con la siguiente. Si una capa contiene un vector $\boldsymbol{v}$, la siguiente contiene el vector $\boldsymbol{w} = (A\boldsymbol{v} + \boldsymbol{b})_+$. A *es una matriz de "pesos". Optimizamos esos pesos para que "concuerden" con los datos de entrenamiento.* Así las derivadas de la función de pérdida L serán cero para los pesos óptimos.

Usando la regla de la cadena de cálculo, las derivadas de L se obtienen multiplicando las derivadas $\partial \boldsymbol{w}/\partial A$ de cada capa a la siguiente capa. Este es un paso lineal $A\boldsymbol{v} + \boldsymbol{b}$, seguido de una función ReLU no lineal que hace que todas las componentes negativas sean cero. La derivada del paso lineal es nuestro tensor de orden tres $v_k \delta_{ij}$.

Esto volverá a aparecer en la Sección VII.3, pero no usaremos explícitamente cálculo tensorial. La idea de la **retropropagación** es calcular todas las derivadas de L "automáticamente". Para cada paso en el cálculo de L, la derivada de ese paso entra en la derivada de L. Nuestra sencilla fórmula (8) para un tensor interesante quedará enterrada en la retropropagación.

Ejemplo 3 : el tensor de probabilidad conjunta

Supongamos que medimos la edad a en años, la altura h en pulgadas y el peso w en libras. Ponemos N niños en I grupos etarios, J grupos de altura y K grupos de peso. Entonces un niño típico estará en el grupo etario i y grupo de altura j y grupo de peso k: aquí los números i, j, k están entre $1, 1, 1$ e I, J, K.

Elijamos un niño al azar. Supongamos que los I grupos etarios contienen $a_1, a_2, \ldots, a_I$ niños (sumando en total N niños). Entonces un niño aleatorio está en el grupo etario i con probabilidad $\boldsymbol{a_i/N}$. De manera similar los J grupos de altura contienen $h_1, h_2, \ldots, h_J$ niños y los K grupos de peso contienen $w_1, w_2, \ldots, w_K$ niños. Para ese niño aleatorio,

$$\text{Probabilidad del grupo de altura } j \text{ es } \frac{h_j}{N} \qquad \text{Probabilidad del grupo de peso } k \text{ es } \frac{w_k}{N}$$

Ahora viene nuestro objetivo real: **las probabilidades conjuntas $\boldsymbol{p_{ijk}}$**. Para cada combinación i, j, k contamos solo los niños que están en el grupo etario i y también en el grupo de alutra j y también en el grupo de peso k. Cada niño tiene I *por* J *por* K *posibilidades*. (Posiblemente $\boldsymbol{p_{I11}}$ sea cero, el niño mayor no tendrá la menor altura y menor peso). Suponga que N_{ijk} niños son encontrados en la *intersección* del grupo etario i con el grupo de altura j y el grupo de peso k:

$$\textbf{La probabilidad conjunta de esta combinación de edad, altura y peso es } \boldsymbol{p_{ijk} = \frac{N_{ijk}}{N}}. \qquad (9)$$

Tenemos I por J por K números p_{ijk}. Todos esos números se encuentran entre 0 y 1. Encajan en un tensor de orden tres T de probabilidades conjuntas. Este tensor T tiene I filas y J columnas y K "tubos". La suma de todos los elementos N_{ijk}/N es 1.

Para apreciar este tensor de I por J por K, supongamos que sumamos todos los números $\boldsymbol{p_{2jk}}$. Estamos tomando en cuenta todos los niños en el grupo etario 2:

$$\sum_{j=1}^{J} \sum_{k=1}^{K} \boldsymbol{p_{2jk}} = \boldsymbol{p_2^a} = \textbf{ probabilidad de que un niño esté en el grupo etario 2}. \qquad (10)$$

Vemos una **sección** 2D del tensor T. Sin duda la suma $p_1^a + p_2^a + \cdots + p_I^a$ es igual a 1.

Análogamente podemos sumar todos los números $\boldsymbol{p_{2j5}}$. Ahora estamos teniendo en cuenta todos los niños en el grupo etario 2 y grupo de peso 5 :

$$\sum_{j=1}^{J} \boldsymbol{p_{2j5}} = \boldsymbol{p_{25}^{aw}} = \textbf{ probabilidad del grupo etario 2 y grupo de peso 5.} \tag{11}$$

Estos números están en una *columna* de T. Podemos combinar columnas para hacer una sección de T. Podemos combinar secciones para producir todo el tensor T :

$$\sum_{i=1}^{I} \sum_{k=1}^{K} \boldsymbol{p_{ik}^{aw}} = \sum_{i=1}^{I} \boldsymbol{p_i^a} = \mathbf{1}$$

Al medir tres propiedades llegamos a un tensor T de orden 3 con elementos T_{ijk}.

La norma y el rango de un tensor

En general un tensor es un arreglo de dimensión d. De la misma manera que un elemento de una matriz necesita números i, j para identificar su posición, un tensor de orden d necesita d números. Nos enfocaremos aquí en $d = 3$ y *tensores de orden* 3 (tensores 3-way). Después de los vectores y matrices, $d = 3$ es el más común y más fácil de entender. La norma de $\boldsymbol{T}$ es como la norma de Frobenius de una matriz : **sumar todos los $\boldsymbol{T_{ijk}^2}$ para encontrar $\boldsymbol{||T||^2}$.**

La teoría de tensores es también parte del álgebra lineal (o tal vez del álgebra multilineal). Al igual que una matriz, un tensor puede tener dos roles distintos en la ciencia y en la ingeniería :

1 Un tensor puede multiplicar vectores, matrices o tensores: es un **operador lineal**.

2 Un tensor puede **contener datos**. Sus elementos pueden ser el brillo de los píxeles de una imagen. Una imagen a color es de orden 3, apilando RGB. Un vídeo a color es un tensor de orden 4.

El tensor operador podría multiplicar el tensor de datos de la misma manera que una matriz de permutación o una de reflexión o una matriz ortogonal opera sobre una matriz de datos.

Las analogías son claras pero los tensores requieren de más índices y se ven más complicados. *Lo son.* Podríamos tener éxito con la multiplicación de tensores (como con las matrices, las operaciones pueden venir en distintos órdenes). *No tendremos tanto éxito con la factorización de tensores.* Esta ha sido y aún es un área de investigación intensa: poder extender a tensores tanto como sea posible las factorizaciones de matrices que son tan fundamentales para el álgebra lineal : $LU, QR, Q\Lambda Q^{\mathrm{T}}, U\Sigma V^{\mathrm{T}}$.

Ni siquiera la definición y cálculo del **"rango de un tensor"** son tan sencillos ni exitosos como los de rango de una matriz. Pero los tensores de rango uno = productos externos, son los más sencillos y los más claros : se crean a partir de tres vectores $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$.

$$\boxed{\textbf{tensor de orden 3} \qquad \boldsymbol{T = a \circ b \circ c} \textbf{ de rango uno} \qquad\qquad \boldsymbol{T_{ijk} = a_i b_j c_k}} \tag{12}$$

Este producto externo $\boldsymbol{a \circ b \circ c}$ está definido por los $m + n + p$ números en esos tres vectores. *El rango de un tensor es el menor número de tensores de rango* 1 *que suman* T.

Si sumamos varios de estos productos externos, tenemos un tensor de rango bajo (incluso si no sabemos exactamente su rango). He aquí un ejemplo para mostrar por qué.

$$\boldsymbol{T = u \circ u \circ v + u \circ v \circ u + v \circ u \circ u} \text{ (tres tensores de rango 1 con } ||u|| = ||v|| = 1)$$

T parece tener rango 3, pero es el límite de estos tensores T_n de rango **2** cuando $n \to \infty$:

$$\boldsymbol{T_n} = n\left(\boldsymbol{u} + \frac{1}{n}\boldsymbol{v}\right) \circ \left(\boldsymbol{u} + \frac{1}{n}\boldsymbol{v}\right) \circ \left(\boldsymbol{u} + \frac{1}{n}\boldsymbol{v}\right) - n\,\boldsymbol{u} \circ \boldsymbol{u} \circ \boldsymbol{u} \tag{13}$$

¿ Por qué nunca ocurre esto con las matrices ? Porque la aproximación más cercana a A por una matriz de rango k es fija debido al teorema de Eckart-Young. Esa mejor aproximación es A_k usando los k vectores singulares principales de la SVD. La distancia de una de rango 3 a una de rango 2 es fija.

Desafortunadamente parece que no hay SVD para un tensor de orden 3 general. Pero igualmente seguimos intentando, porque para los cálculos queremos una buena aproximación de T de rango bajo. Dos opciones son CP y Tucker.

La descomposición CP de un tensor

Un problema fundamental en análisis tensorial y en cálculo tensorial rápido es aproximar un tensor dado T por una **suma de tensores de rango uno** : una factorización aproximada.

$$\boxed{\textbf{Descomposición CP} \qquad \boldsymbol{T \approx a_1 \circ b_1 \circ c_1 + \cdots + a_R \circ b_R \circ c_R}} \tag{14}$$

Esta descomposición tuvo varios descubridores : Hitchcock, Carroll, Chang, y Harshman. También tiene nombres desafortunados como CANDECOMP y PARAFAC. Con el tiempo se la llamó una **Descomposición CP de $\boldsymbol{T}$**.

Esto parece una extensión a tensores de la SVD. Pero hay diferencias importantes. *Los vectores $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_R$ no son ortogonales* (lo mismo para los $\boldsymbol{b}$ y los $\boldsymbol{c}$). No tenemos invarianza ortogonal (que hace que $Q_1 A Q_2^{\mathrm{T}}$ tenga los mismos valores singulares que A). Y el teorema de Eckart-Young no es cierto: generalmente no conocemos el tensor de rango R más cercano a T. Hay otros enfoques para la descomposición de tensores pero por ahora **CP** ha sido la más útil. Kruskal probó que los tensores de rango uno más cercanos son únicos (si existen). Si cambiamos R, el mejor $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ cambia.

Nos encontramos ante un problema completamente nuevo. Desde la perspectiva de la computabilidad, el problema es NP-difícil (no se puede resolver en tiempo polinomial a menos que $P = NP$, lo que soprendería a casi todos). Lim y Hillar probaron que muchos problemas aparentemente más sencillos de tensores también son NP-difíciles. Los cálculos exactos no son podibles.

C. Hillar and L.-H. Lim, *Most tensor problems are NP-hard*, J. ACM **60** (2013) Article 45.

Buscamos un algoritmo que calcule los vectores $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ de manera razonablemente eficiente. Un paso importante en el cálculo de tensores es acercarse a la mejor descomposición CP. Una idea simple, (*cuadrados mínimos alternados*) funciona razonablemente bien por ahora. El problema general no es convexo, pero el ciclo de subproblemas al mejorar A luego B luego C, y cada subproblema (para A, B y C) es de cuadrados mínimos convexo.

Alternar $\boldsymbol{A}, \boldsymbol{B}$, y $\boldsymbol{C}$			(15)
	(a) Fijar B, C y variar $\boldsymbol{A}$	**Minimizar** $\|\|\boldsymbol{T_1} - \boldsymbol{A}(\boldsymbol{C} \odot \boldsymbol{B})^{\mathbf{T}}\|\|_{\boldsymbol{F}}^{\mathbf{2}}$	
	(b) Fijar A, C y variar $\boldsymbol{B}$	(c) Fijar A, B y variar $\boldsymbol{C}$	

Este algoritmo alternante usa las tres formas matriciales T_1, T_2, T_3, descriptas a continuación. $C \circ B$ es el "producto de Khatri-Rao" que veremos en la ecuación (17). **Así en (15) son todas matrices.**

Forma matricial de un tensor T

Sean A, B, C las matrices cuyas columnas son los $\boldsymbol{a}$, los $\boldsymbol{b}$ y los $\boldsymbol{c}$ en (14). Cada matriz tiene R columnas. Si el tensor T de tercer orden tiene dimensiones I, J, K entonces las tres matrices son de I por R, J por R, y K por R. Es bueno "matriciar" al tensor T para calcular la descomposición CP.

Comencemos separando los $\boldsymbol{a}$ en la matriz A de I por R. Luego buscamos una matriz M_1 de R por JK de manera que AM_1 exprese nuestra suma (14) de tensores de rango uno. M_1 debe obtenerse de los $\boldsymbol{b}$ y los $\boldsymbol{c}$, pero ¿ de qué manera un producto matricial AM_1 expresará un tensor de orden 3 ? La respuesta es que debemos **desdoblar el tensor $\boldsymbol{T}$ en una matriz $\boldsymbol{T_1}$**. Luego podemos comparar $\boldsymbol{T_1}$ con AM_1.

Un ejemplo de Kolda y Bader muestra cómo los tensores se desdoblan en matrices. Tenemos $I \times J \times K = 3 \times 4 \times 2 = 24$ números en T. El desdoblamiento matricial de T puede ser 3×8 o 4×6 o 2×12. Tenemos tres desdoblamientos T_1, T_2, T_3, cortando T de tres maneras :

Primera manera
secciones frontal y trasera
$I \times JK = 3 \times 8$

$$\boldsymbol{T_1} = \left[\begin{array}{cccc|cccc} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{array}\right] \tag{16}$$

Segunda manera
$J \times IK = 4 \times 6$
Los mismos 24 números

$$\boldsymbol{T_2} = \left[\begin{array}{cccccc} 1 & 2 & 3 & 13 & 14 & 15 \\ 4 & 5 & 6 & 16 & 17 & 18 \\ 7 & 8 & 9 & 19 & 20 & 21 \\ 10 & 11 & 12 & 22 & 23 & 24 \end{array}\right]$$

Tercera manera
$K \times IJ = 2 \times 12$
Los mismos 24 números

$$\boldsymbol{T_3} = \left[\begin{array}{cccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 \end{array}\right]$$

El producto $A \odot B$ de Khatri-Rao

La Sección IV.3 presentará el **Producto de Kronecker $\boldsymbol{K = A \otimes B}$** de dos matrices A y B. Contiene todos los productos a_{ij} por b_{kl} de los elementos en A y B (así que puede ser una matriz grande). Si A y B son solo *vectores columna* (matrices de J por 1 y de K por 1) entonces $\boldsymbol{A \otimes B}$ **es una columna larga** : JK por 1. Primero viene a_{11} por cada elemento de B, luego a_{21} por esos mismos K elementos, y finalmente a_{J1} por esos K elementos. $\boldsymbol{A \odot B}$ **tiene $\boldsymbol{R}$ de esas columnas largas.**

Khatri-Rao multiplica todos los $a_{ij}b_{Ij}$ para encontrar la columna j. A y B y $A \odot B$ tienen R columnas :

$$\boxed{\textbf{Khatri-Rao} \qquad \textbf{Columna } j \textbf{ de } \boldsymbol{A \odot B} = \textbf{(columna } j \textbf{ de } \boldsymbol{A}) \otimes \textbf{(columna } j \textbf{ de } \boldsymbol{B})} \tag{17}$$

Así $\boldsymbol{C}$ y $\boldsymbol{B}$ ($\boldsymbol{K}$ por $\boldsymbol{R}$ y $\boldsymbol{J}$ por $\boldsymbol{R}$) producen $\boldsymbol{C \odot B}$ con $\boldsymbol{R}$ columnas largas ($\boldsymbol{JK}$ por $\boldsymbol{R}$).

Resumen T se aproxima por $\sum \boldsymbol{a}_i \circ \boldsymbol{b}_i \circ \boldsymbol{c}_i$. Cortamos T en tres direcciones y colocamos los cortes uno al lado del otro en tres matrices T_1, T_2, T_3. Luego buscamos tres matrices M_1, M_2, M_3 que nos den ecuaciones casi correctas mediante la multiplicación matricial ordinaria :

$$T_1 \approx AM_1 \quad \text{y} \quad T_2 \approx BM_2 \quad \text{y} \quad T_3 \approx CM_3. \tag{18}$$

Recordemos que A es de I por R, con columnas $\boldsymbol{a}_1$ a $\boldsymbol{a}_R$. T_1 es de I por JK. Entonces la M_1 correcta debe ser de R por JK. Esta M_1 viene de las matrices B y C, con columnas $\boldsymbol{b}_j$ y $\boldsymbol{c}_k$.

$\boldsymbol{M_1}$ es la traspuesta del producto de Khatri-Rao $\boldsymbol{C \odot B}$ que es de $\boldsymbol{JK}$ por $\boldsymbol{R}$.

La i-ésima columna de $C \odot B$ viene de las i-ésimas columnas de C y B (para $i = 1$ a R). Esa i-ésima columna de $C \odot B$ contiene todos los JK productos de los números c_{ki} y b_{ji}, para $1 \le k \le K$ y $1 \le j \le J$. $C \odot B$ contiene todos los JKR productos de c_{ki} y b_{ji}, que vienen de las i-ésimas columnas de C y B para $i = 1$ a R.

Las tres formas matriciales T_1, T_2, T_3 de T ahora son productos matriciales aproximados. La definición de Khatri-Rao se inventó para que la ecuación (19) sea cierta.

$$\boxed{T_1 \approx A(C \odot B)^{\mathrm{T}} \qquad T_2 \approx B(C \odot A)^{\mathrm{T}} \qquad T_3 \approx C(B \odot A)^{\mathrm{T}}} \tag{19}$$

Calculando la descomposición CP de T

Queremos calcular los $\boldsymbol{a}$, los $\boldsymbol{b}$ y los $\boldsymbol{c}$ en la aproximación (14) al tensor T. El plan es usar **minimizaciones alternantes**. Con los $\boldsymbol{b}$ y los $\boldsymbol{c}$ en las matrices B y C fijas, resolvemos un problema de mínimos cuadrados para los $\boldsymbol{a}$ en A.

Coloquemos T en su forma matricial T_1. Esa matriz es de I por JK. Por la ecuación (19), buscamos $T_1 \approx A\,(C \odot B)^{\mathrm{T}} = (I \times R)(R \times JK)$. Por ahora fijemos B y C.

$$\textbf{Elijamos el mejor } \boldsymbol{A} \text{ en } \; \boldsymbol{||T_1 - A\,(C \odot B)^{\mathrm{T}}||_F^2 = ||T_1^{\mathrm{T}} - (C \odot B)\,A^{\mathrm{T}}||_F^2}\,. \tag{20}$$

Aquí $C \odot B$ es la matriz de coeficientes de $JK \times R$. Multiplica cada una de las I columnas de A^{T}. Con la norma de Frobenius, tenemos que resolver I problemas de mínimos cuadrados ordinarios para A : uno para cada columna de A^{T} (= fila de A).

A tener en cuenta : A no está en su posición usual $A\boldsymbol{x} = \boldsymbol{b}$ para mínimos cuadrados. Las filas de A son las incógnitas. **La matriz de coeficiente es $\boldsymbol{C \odot B}$** (no A). Esperamos que la matriz sea alta y delgada con $JK \ge R$. De manera similar queremos $IK \ge R$ e $IJ \ge R$ cuando las incógnitas están alternadas para convertirse en B y C.

La solución al problema de mínimos cuadrados $A\boldsymbol{x} = \boldsymbol{b}$ está dada por la pseudoinversa $\widehat{\boldsymbol{x}} = A^{+}\boldsymbol{b}$.

Si esa matriz A tiene columnas independientes (como se suele suponer en mínimos cuadrados), A^{+} es una inversa a izquierda $(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}$ de la matriz A. Allí podemos ver la matriz de coeficientes $A^{\mathrm{T}}A$ en las ecuaciones normales usuales para el mejor $\widehat{\boldsymbol{x}}$. En nuestro caso esa matriz de coeficientes no es A sino el producto de Khatri-Rao $C \odot B$. Por suerte la pseudoinversa de nuestra matriz de coeficientes $C \odot B$ puede expresarse como

$$(C \odot B)^{+} = [(C^{\mathrm{T}}C) \mathbin{.*} (B^{\mathrm{T}}B)]^{+}(C \odot B)^{\mathrm{T}}. \tag{21}$$

Esta fórmula fue tomada prestada de la ecuación (2.2) de Kolda y Bader. Nos permite formar $C^{\mathrm{T}}C$ y $B^{\mathrm{T}}B$ por adelantado (matrices de R por R). El símbolo $\mathbf{.*}$ (o frecuentemente $\circ$) representa el producto elemento por elemento (el **producto de Hadamard**). Trasponiendo, el problema de mínimos cuadrados (20) con B y C fijos se resuelve utilizando la matriz

$$\boxed{A = T_1\,(C \odot B)\,(C^{\mathrm{T}}C \mathbin{.*} B^{\mathrm{T}}B)^{+}} \tag{22}$$

Luego usamos esta A junto con C para encontrar la mejor B. Un ciclo del algoritmo alternante termina al encontrar la mejor C, con A y B fijas en sus nuevos valores.

La descomposición de Tucker

La SVD separa una matriz (un tensor de orden 2) en $U\Sigma V^{\mathrm{T}}$. Las columnas de U y V son ortonormales. Esa descomposición es generalmente imposible para tensores de orden superior. Esta es la razón de la aproximación CP y también de la aproximación Tucker.

Tucker permite P vectores columna $\boldsymbol{a}_p$, Q vectores columna $\boldsymbol{b}_q$ y R vectores columna $\boldsymbol{c}_r$. *Entonces todas las combinaciones de rango uno $\boldsymbol{a}_p \circ \boldsymbol{b}_q \circ \boldsymbol{c}_r$ están permitidas*. Un tensor central G con dimensiones P, Q, R determina los coeficientes en esas combinaciones :

$$\boxed{\textbf{Descomposición de Tucker de } \boldsymbol{T} \qquad T \approx \sum_{1}^{P}\sum_{1}^{Q}\sum_{1}^{R} g_{pqr}\ \boldsymbol{a}_p \circ \boldsymbol{b}_q \circ \boldsymbol{c}_r} \tag{23}$$

Con esta libertad adicional en G (que era tan solo un tensor diagonal en la descomposición CP) podemos pedir que los $\boldsymbol{a}$, los $\boldsymbol{b}$ y los $\boldsymbol{c}$ sean *tres conjuntos de columnas ortonormales*. Entonces Tucker es una combinación de PQR tensores de rango uno, en lugar de solo R.

Recordemos que (23) es una aproximación y no una igualdad. Se generaliza a tensores de orden d. El caso de orden 3 tiene una forma matricial, donde T_1, T_2, T_3 y análogamente G_1, G_2, G_3 son las matrices desdobladas de la ecuación (16). Las matrices CP en (14) cambian a matrices Tucker, y ahora tenemos productos de Kronecker en lugar de Khatri-Rao :

$$\boxed{\textbf{Tucker} \quad T_1 \approx A\,G_1\,(C \otimes B)^{\mathrm{T}} \quad T_2 \approx B\,G_2\,(C \otimes A)^{\mathrm{T}} \quad T_3 \approx C\,G_3\,(B \otimes A)^{\mathrm{T}}} \tag{24}$$

La *SVD de orden superior (HOSVD, por sus siglas en inglés)* es una descomposición particular de Tucker. Sus propiedades y su cálculo están mejor desarrolladas en el trabajo de De Lathauwer.

Descomposición y aleatorización para tensores grandes

En esta sección se han desarrollado los pasos básicos del cálculo con tensores. *Los datos llegan en forma de tensores*. Terminamos con dos construcciones nuevas (desde la CP) y con referencias.

1. Descomposición en tren de tensores (Oseledets y Tyrtyshnikov) El problema es manejar tensores de orden d. La descomposición CP completa aproximaría a T por una suma de tensores de rango uno. Para dimensiones d grandes, la CP se vuelve imposible. Una mejor manera es reducir T a un tren de tensores de orden 3. Así el álgebra lineal y la CP pueden operar en este formato de tren de tensores.

2. Descomposiciones CURT (Song, Woodruff y Zhong) Esta es una aproximación de rango bajo para tensores. Este artículo busca calcular un tensor de rango k a distancia ϵ del más cercano a T. Para matrices, esto se logra (con $\epsilon = 0$) con la SVD y el teorema de Eckart-Young. Para tensores no tenemos SVD, sino que los cálculos están basados en un aproximación CUR columna-fila (Sección III.3) con un *tensor de mezcla* U.

El algortimo casi cumple el objetivo de los nnz pasos : igual al número de valores distintos de cero en T. Usa **factorizaciones aleatorias**, la poderosa herramienta en la Sección II.4 para cálculos muy grandes. Los tensores están en la frontera del álgebra lineal numérica.

1. T. Kolda and B. Bader, *Tensor decompositions and applications*, SIAM Review **52** (2009) 455–500.
2. M. Mahoney, M. Maggioni, and P. Drineas, *Tensor-CUR decompositions for tensor-based data*, SIAM J. Matrix Analysis Appl. **30** (2008) 957–987.
3. B. Bader and T. Kolda, MATLAB Tensor Toolbox, version 2.2 (2007).
4. C. Andersson and R. Bro, The N-Way Toolbox for MATLAB (2000).
5. R. A. Harshman, http://www.psychology.uwo.ca/faculty/harshman
6. P. Paatero and U. Tapper, *Positive matrix factorization*, Environmetrics **5** (1994) 111–126.
7. D. D. Lee and H. S. Seung, *Learning the parts of objects by non-negative matrix factorization*, Nature **401** (1999) 788–791.
8. L. De Lathauwer, B. de Moor, and J. Vandewalle, SIAM J. Matrix Anal. Appl. **21** (2000) 1253–1278 and 1324–1342 (*and more recent papers on tensors*).
9. S. Ragnarsson and C. Van Loan, *Block tensor unfoldings*, SIAM J. Matrix Anal. Appl. **33** (2013) 149–169, arXiv : 1101.2005, 2 Oct 2011.
10. C. Van Loan, **www.alm.unibo.it/∼simoncin/CIME/vanloan1.pdf—vanloan4.pdf**
11. I. Oseledets, *Tensor-train decomposition*, SIAM J. Sci. Comp. **33** (2011) 2295-2317.
12. Z. Song, D. P. Woodruff, and P. Zhong, *Relative error tensor low rank approximation*, arXiv : 1704.08246, 29 Mar 2018.
13. P. Grinfeld, *Introduction to Tensor Calculus and the Calculus of Moving Surfaces*, Springer (2013).

Problemas de la Sección I.12

Las primeras 5 preguntas son sobre minimizar $||A - UV||_F^2$ cuando UV tiene rango 1.

Problema Minimice $\left\| \begin{bmatrix} a & c \\ b & d \end{bmatrix} - \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \begin{bmatrix} v_1 & v_2 \end{bmatrix} \right\|_F^2$

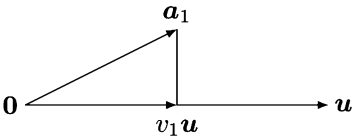
1 Mire la primera columna de $A - UV$ con A y U fijas :

$$\text{Minimice} \quad \left\| \begin{bmatrix} a - v_1 u_1 \\ b - v_1 u_2 \end{bmatrix} \right\|^2 = (a - v_1 u_1)^2 + (b - v_1 u_2)^2$$

Muestre usando cálculo que el número v_1 donde se alcanza el mínimo verifica $(u_1^2 + u_2^2)\, v_1 = u_1 a + u_2 b$.

En notación vectorial $\boldsymbol{u}^{\mathrm{T}} \boldsymbol{u}\, v_1 = \boldsymbol{u}^{\mathrm{T}} \boldsymbol{a}_1$ donde $\boldsymbol{a}_1$ es la columna 1 de A.

2 ¿ Qué punto $v_1 \boldsymbol{u}$ en esta imagen minimiza $||\boldsymbol{a}_1 - v_1 \boldsymbol{u}||^2$?



El vector error $\boldsymbol{a}_1 - v_1 \boldsymbol{u}$ es _____ a $\boldsymbol{u}$.

A partir de ese hecho, encuentre de nuevo el número v_1.

3 La segunda columna de $A - UV$ es $\begin{bmatrix} c \\ d \end{bmatrix} - \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} v_2 = \boldsymbol{a}_2 - v_2 \boldsymbol{u}$. ¿ Qué número v_2 minimiza $||\boldsymbol{a}_2 - v_2 \boldsymbol{u}||^2$?

Forma vectorial	El mejor $\boldsymbol{v} = \begin{bmatrix} v_1 & v_2 \end{bmatrix}$ resuelve $(\boldsymbol{u}^{\mathrm{T}} \boldsymbol{u})\, \boldsymbol{v} = \boldsymbol{u}^{\mathrm{T}} A$

4 Los problemas 1 a 3 minimizan $||A - UV||_F^2$ con U fija, cuando UV tiene rango 1.

Con $V = \begin{bmatrix} v_1 & v_2 \end{bmatrix}$ fijo, ¿ cuál $U = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$ da el mínimo de $||A - UV||_F^2$?

5 (Computadora) A partir de cualquier U_0, ¿ converge esta minimización alternada a la matriz de rango uno más cercana $A_1 = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^{\mathrm{T}}$ de la SVD ?

$$\begin{array}{lll} \underset{V}{\text{Minimice}} & ||A - U_n V||_F^2 \quad \text{en } V = V_n & A \text{ es de } 3 \times 3 \\ & & U \text{ es de } 3 \times 1 \\ \underset{U}{\text{Minimice}} & ||A - U V_n||_F^2 \quad \text{en } U = U_{n+1} & V \text{ es de } 1 \times 3 \end{array}$$

Observación : estas preguntas también son una introducción a mínimos cuadrados (Sección II.2). Para V fija o U fija, cada minimización es un problema de mínimos cuadrados, aún cuando el rango de UV es mayor a 1. Pero requerir la no negatividad o raleza de U y V hace que cada minimización sea más difícil y se necesiten nuevos métodos.

Los problemas 6 a 11 son sobre tensores. No estamos haciendo cálculo (con derivadas) o álgebra de tensores (en espacios de tensores). Nuestro enfasis está en un solo tensor $\boldsymbol{T}$ que contiene datos multidimensionales. Si tenemos las muestras 1 a n y cada muestra es una imagen (una matriz) entonces tenemos un tensor de orden 3. *¿ Cómo podemos aproximar este tensor T con una combinación de tensores simples* ? Esta pregunta corresponde a ciencia de datos.

Primero debemos decidir : el tensor $\boldsymbol{a} \otimes \boldsymbol{b} \otimes \boldsymbol{c}$ de rango 1 en la ecuación (12) es simple. Su elemento i, j, k es el número a_i por b_j por c_k, igual que una matriz de rango 1 $\boldsymbol{a}\boldsymbol{b}^{\mathrm{T}}$ con sus elementos $a_i b_j$. Una suma de tensores simples aproxima a $\boldsymbol{T}$.

6 Dada una matriz A de m por n, ¿ cómo sabemos si A tiene rango 1 ?

7 Dado un tensor T de m por n por p, ¿ cómo sabemos si T tiene rango 1 ?

8 El mayor rango posible de un tensor de 2 por 2 por 2 es 3. ¿ Puede encontrar un ejemplo ?

9 (a) Suponga que conoce las sumas de las filas r_1 a r_m y las sumas de las columnas c_1 a c_n de una matriz A de m por n. ¿ Qué condición deben cumplir esos números ?

(b) Para un tensor de m por n por p, las secciones son matrices de n por p, matrices de m por p y matrices de m por n. Suponga que suma los elementos en cada una de esas m secciones, n secciones y p secciones. ¿ Qué condiciones que conectan esos m números, n números y p números estarían garantizadas ?

10 Suponga que todos los elementos de un tensor T de $2 \times 2 \times 2$ son 1, excepto el primero que es $T_{111} = 0$. Escriba T como una suma de dos tensores de rango 1. ¿ Cuál es el tensor de rango 1 más cercano a T (en la norma habitual de Frobenius) ?

11 Un tensor $\boldsymbol{T}$ de 2 por 2 por 2 multiplicado por un vector $\boldsymbol{v}$ en $\mathbf{R}^2$ debería producir una matriz A en $\mathbf{R}^{2\times 2}$. ¿ Cómo definiría ese resultado $A = \boldsymbol{T}\boldsymbol{v}$?

Parte II

Cálculos con matrices grandes

II.1 Álgebra lineal numérica

II.2 Cuadrados mínimos: cuatro maneras

II.3 Tres bases para el espacio columna

II.4 Álgebra lineal aleatorizada

Parte II : Cálculos con matrices grandes

En esta parte del libro se describen diversos procedimientos que permiten resolver el problema lineal $A\boldsymbol{x} = \boldsymbol{b}$. La eliminación ordinaria puede calcular un $\boldsymbol{x}$ exacto, o no. Puede haber demasiadas ecuaciones ($m > n$) y ninguna solución. Una matriz cuadrada puede ser singular. La solución puede ser imposible de calcular (A es sumamente mal condicionada, o simplemente es demasiado grande). En aprendizaje profundo tenemos *demasiadas soluciones* y queremos una que generalice bien a los datos de prueba nunca vistos.

Estas dos páginas intentarán distinguir esas fuentes de dificultad. Somos como médicos realizando una evaluación: identificamos los problemas y sugerimos un curso de acción para cada uno. La "pseudoinversa" de A propone una inversa para cada matriz, pero esta podría no ser útil.

0. Toda matriz $A = U\Sigma V^{\mathrm{T}}$ tiene una **pseudoinversa** $\boldsymbol{A^+ = V\Sigma^+U^{\mathrm{T}}}$. Para la matriz diagonal Σ la pseudoinversa Σ^+ contiene $1/\sigma_k$ para cada valor singular distinto de cero. *Pero la pseudoinversa de* 0 *es* 0. Saber cuándo un número es exactamente cero es un requisito sumamente estricto y en muchos cálculos es imposible.

La pseudoinversa de la Sección II.2 es una manera de resolver $A\boldsymbol{x} = \boldsymbol{b}$. He aquí otras maneras :

1. Supongamos que A es cuadrada e invertible, de tamaño razonable, y su número de condición σ_1/σ_n no es grande. Aquí la **eliminación** tendrá éxito (probablemente con intercambios de fila). Tenemos $PA = LU$ o $A = LU$ (con intercambios de fila o sin) como en la Sección I.4.

El comando barra invertida (*backslash* en inglés) $A\backslash\boldsymbol{b}$ esta diseñado para diagonalizar por bloques, cuando sea posible, la matriz A.

2. Supongamos que $m > n = r$: hay demasiadas ecuaciones en $A\boldsymbol{x} = \boldsymbol{b}$ como para esperar una solución. Si las columnas de A son independientes y no demasiado mal condicionadas, entonces resolvemos las **ecuaciones normales** $\boldsymbol{A^{\mathrm{T}}A\widehat{x} = A^{\mathrm{T}}b}$ para encontrar la solución por cuadrados mínimos $\widehat{\boldsymbol{x}}$.

Probablemente el vector $\boldsymbol{b}$ no esté en el espacio columna de A, y probablemente $A\boldsymbol{x} = \boldsymbol{b}$ sea imposible. En la Sección II.2 $A\widehat{\boldsymbol{x}}$ es la **proyección de $\boldsymbol{b}$** sobre ese espacio columna.

Esos son buenos problemas para resolver: una A invertible o una $A^{\mathrm{T}}A$ invertible, bien condicionadas y no demasiado grandes. La próxima página describe cuatro cálculos (todavía ecuaciones lineales) que son más difíciles.

3. Supongamos que $m < n$. Ahora la ecuación $A\boldsymbol{x} = \boldsymbol{b}$ puede tener muchas soluciones. El espacio nulo de A es distinto del espacio trivial $\{\mathbf{0}\}$. **La solución $\boldsymbol{x}$ está indeterminada**. Queremos elegir el mejor $\boldsymbol{x}$ para nuestro objetivo. Dos opciones posibles son $\boldsymbol{x}^+$ y $\boldsymbol{x}_1$:

$\boldsymbol{x} = \boldsymbol{x}^+ = A^+\boldsymbol{b}$. La pseudoinversa A^+ da la **solución que minimiza la norma $\boldsymbol{\ell^2}$** entre las que tienen componente del espacio nulo = cero.
$\boldsymbol{x} = \boldsymbol{x}_1 =$ **solución que minimiza la norma $\boldsymbol{\ell^1}$**. Esta solución suele ser rala (muchas componentes nulas) y muy deseable. Aparece al realizar "búsqueda de bases" ("*basis pursuit*" en inglés) en la Sección III.4.

4. Las columnas de A pueden estar **mal condicionadas**. Ahora el cociente σ_1/σ_r es demasiado grande. Entonces $\boldsymbol{x}$ no está bien determinado (como en la interpolación de orden superior : Sección III.3). La solución típica es **ortogonalizar las columnas** vía el algoritmo de Gram-Schmidt o de Householder. Crear vectores ortonormales $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n$ a partir de las columnas $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$.

La Sección II.2 explica dos maneras importantes de Gram-Schmidt. El modo estándar es ortogonalizar cada columna $\boldsymbol{a}_{k+1}$ contra las direcciones conocidas $\boldsymbol{q}_1$ a $\boldsymbol{q}_k$. La manera más segura ortogonaliza todas las $n - k$ columnas restantes contra $\boldsymbol{q}_k$ apenas se calcula este vector. Así una columna pequeña $k + 1$ puede intercambiarse con una columna posterior cuando sea necesario. La manera más segura de todas elige la columna más grande disponible en cada paso.

5. A puede ser casi singular (como en **4**). En este caso $A^{\mathrm{T}}A$ tendrá una inversa muy grande. Gram-Schmidt puede fallar. Un enfoque diferente es agregar un **término de penalización** :

$$\text{Minimizar } ||A\boldsymbol{x} - \boldsymbol{b}||^2 + \delta^2||\boldsymbol{x}||^2 \qquad \text{Resolver } (A^{\mathrm{T}}A + \delta^2 I)\boldsymbol{x}_{\boldsymbol{\delta}} = A^{\mathrm{T}}\boldsymbol{b}$$

Cuando la penalización δ^2 tiende a cero, $(A^{\mathrm{T}}A + \delta^2 I)^{-1}A^{\mathrm{T}}$ tiende a la pseudoinversa A^+ (Sección II.2). Logramos la invertibilidad al agregar $\delta^2 I$ para hacer que $A^{\mathrm{T}}A$ sea más positiva. Esto está conectado a regresión contraída (*ridge regression*) en estadística.

Términos de penalización como $\delta^2||\boldsymbol{x}||^2$ son comunes en **problemas inversos**, que buscan reconstruir un sistema a partir de conocimiento de sus salidas (outputs). Solemos conocer el sistema (como una red eléctrica) y encontramos sus salidas (corrientes y voltajes). El problema inverso comienza con salidas (como tomografías computarizadas o resonancias magnéticas). Reconstruir el sistema es un problema mal condicionado.

6. Supongamos que A es **demasiado grande**. Rebalsa la memoria rápida. Podemos pedir ver algunas columnas pero no podemos hacer eliminación. Multiplicar A^{T} por A sería imposible, incluso con una computadora *petascale* que llegará a Oak Ridge y la máquina *exascale* planeada por Argonne (*New York Times*, 28 de febrero 2018). ¿ Qué se hace con una matriz tan grande ?

La mejor solución es **tomar muestras aleatorias** de las columnas (Sección II.4). Si A es demasiado grande pero bastante coherente, cada $A\boldsymbol{x}$ será una muestra útil del espacio columna. Los resultados del muestreo aleatorio nunca son certeros, pero las probabilidades de equivocarse son bajas. El *álgebra lineal numérica aleatoria* ha generado algoritmos con una base estadística segura.

Esta es una evolución necesaria, o revolución, basada en resultados profundos de probabilidad.

II.1. Algebra lineal numérica

Esta sección resume las ideas centrales del álgebra lineal numérica (clásica). No se explicarán en detalle ya que muchos libros lo hacen bien. Su objetivo es resolver $A\boldsymbol{x} = \boldsymbol{b}$ o $A\boldsymbol{x} = \lambda\boldsymbol{x}$ o $A\boldsymbol{v} = \sigma\boldsymbol{u}$. Son la base sobre la que se construyen nuevos métodos computacionales.

Los verdaderos objetivos de esta parte del libro son esos nuevos métodos, que buscan extraer información de los datos que están en una matriz o un tensor. Cuando una matriz o tensor es muy grande ("big data") a menudo necesitamos **muestrear la matriz**. Puede parecer imposible que las muestras aleatorias den una respuesta confiable, pero en realidad esto es altamente probable.

La biblia para el álgebra lineal numérica es *Matrix Computations*. Sus autores son Gene Golub y Charles Van Loan. La cuarta edición la publicó Johns Hopkins University Press en 2013. Gene dio clases en Johns Hopkins por más de 30 años y eso llevó a la primera edición (creo que la editorial jamás se hubiera imaginado un libro en 2013 con 750 páginas).

Puede encontrar muchos otros libros en **math.mit.edu/learningfromdata**. Aquí destacamos uno : *Numerical Linear Algebra* de Trefethen y Bau. Los títulos de sus capítulos brindan un buen panorama de las ideas y algoritmos centrales :

I. **Fundamentos** (incluyendo hasta la SVD y Eckart-Young)

II. **Factorización $\boldsymbol{QR}$ y mínimos cuadrados** (los 3 métodos : A^+ y $(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}$ y QR)

III. **Condicionamiento y estabilidad** (número de condición, estabilidad hacia atrás, perturbaciones)

IV. **Sistemas de ecuaciones** (eliminación directa : $PA = LU$ y $S = A^{\mathrm{T}}A$ de Cholesky)

V. **Autovalores** (reducción a tridiagonal-Hessenberg-bidiagonal; QR con desplazamientos)

VI. **Métodos iterativos** (Arnoldi, Lanczos, GMRES, gradiente conjugado, Krylov).

Nuestro plan para esta sección es presentar los algoritmos iterativos importantes de las partes **VI** y **V**. Todos se encuentran en los principales códigos o programas computacionales que resuelven $A\boldsymbol{x} = \boldsymbol{b}$, $S\boldsymbol{q} = \lambda\boldsymbol{q}$ y $A\boldsymbol{x} = \lambda\boldsymbol{x}$ y $A\boldsymbol{v} = \sigma\boldsymbol{u}$. La palabra "iterativo" indica que repetimos un paso simple y rápido con la idea de aproximar una solución de un problema más grande y más difícil.

Un modelo para fabricar iteraciones (¡ pero no necesariamente un algoritmo rápido !) se obtiene separando A en $A = S - T$:

$$\textbf{Preparación para la iteración} \qquad \textbf{Reescribimos } A\boldsymbol{x} = \boldsymbol{b} \textbf{ como } S\boldsymbol{x} = T\boldsymbol{x} + \boldsymbol{b}. \tag{1}$$

Comencemos con un $\boldsymbol{x}_0$ cualquiera y resolvamos $S\boldsymbol{x}_1 = T\boldsymbol{x}_0 + \boldsymbol{b}$. Continuemos con $S\boldsymbol{x}_2 = T\boldsymbol{x}_1 + \boldsymbol{b}$. Cien iteraciones es muy común. Si S fue bien elegida, cada paso $\boldsymbol{S\boldsymbol{x}_{k+1} = T\boldsymbol{x}_k + \boldsymbol{b}}$ será rápido.

Restando la iteración $S\boldsymbol{x}_{k+1} = T\boldsymbol{x}_k + \boldsymbol{b}$ de la exacta $S\boldsymbol{x} = T\boldsymbol{x} + \boldsymbol{b}$, el error $\boldsymbol{x} - \boldsymbol{x}_k$ obedece la ecuación de error (y $\boldsymbol{b}$ se cancela):

$$\textbf{Ecuación de error} \qquad S(\boldsymbol{x} - \boldsymbol{x_{k+1}}) = T(\boldsymbol{x} - \boldsymbol{x_k}) \tag{2}$$

Cada paso multiplica el error por $\boldsymbol{S^{-1}T}$. Cuando $||S^{-1}T|| \ll 1$ la convergencia es rápida. Pero en la práctica $S^{-1}T$ suele tener un autovalor cercano a 1. Entonces se necesita una idea mejor: como el método del gradiente conjugado.

Pero antes hago un comentario. Según los libros de texto se pueden encontrar autovalores resolviendo $\det(A - \lambda I) = 0$. Se pueden encontrar valores singulares trabajando con la matriz $A^{\mathrm{T}}A$. En la realidad, esos determinantes son impensables y una $A^{\mathrm{T}}A$ grande puede ser desaconsejable desde el punto de vista numérico. $A\boldsymbol{x} = \lambda\boldsymbol{x}$ y $A\boldsymbol{v} = \sigma\boldsymbol{u}$ son problemas serios. Los resolveremos en esta sección para matrices de tamaño mayor o igual a 100. Para $n = 10^4$, lea la Sección II.4.

Subespacios de Krylov e iteración de Arnoldi

Idea clave: **la multiplicación matriz-vector $\boldsymbol{Ab}$ es rápida, especialmente si $\boldsymbol{A}$ es rala**. Si comenzamos con A y $\boldsymbol{b}$, podemos calcular rápidamente cada uno de los vectores $\boldsymbol{b}, A\boldsymbol{b}, \ldots, A^{n-1}\boldsymbol{b}$ (nunca calcule A^2 o A^3. Solo calcule vectores). Las combinaciones de esos n vectores componen el n-ésimo **subespacio de Krylov**. Buscamos dentro de este subespacio K_n una aproximación cercana a la solución deseada $\boldsymbol{x}$.

El primer problema es encontrar una base mucho mejor que esos vectores $\boldsymbol{b}, A\boldsymbol{b}, \ldots, A^{n-1}\boldsymbol{b}$. ¡Una base ortogonal $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n$ suele ser lo mejor! La idea de Gram-Schmidt de restarle las proyecciones de $\boldsymbol{v} = A\boldsymbol{q}_k$ sobre todos los vectores previos $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_k$ es muy natural. **Este es el método de Arnoldi para encontrar $\boldsymbol{q}_{k+1}$.**

Iteración de Arnoldi	$\boldsymbol{q}_1 = \boldsymbol{b}/\|\|\boldsymbol{b}\|\|$, $\boldsymbol{q}_2, \ldots, \boldsymbol{q}_k$ son conocidas
$\boldsymbol{v} = A\boldsymbol{q}_k$	**Comience con nuevos $\boldsymbol{v}$**
para $j = 1$ a k	Para cada $\boldsymbol{q}$ conocido
$h_{jk} = \boldsymbol{q}_j^{\mathrm{T}}\boldsymbol{v}$	Calcule el producto interno
$\boldsymbol{v} = \boldsymbol{v} - h_{jk}\boldsymbol{q}_j$	Reste la proyección
$h_{k+1,k} = \|\|\boldsymbol{v}\|\|$	Calcule la norma
$\boldsymbol{q}_{k+1} = \boldsymbol{v}/h_{k+1,k}$	**Nuevo vector base con norma 1**

Veámoslo en lenguaje matricial. La última columna es $A\boldsymbol{q}_k$ = combinación de $\boldsymbol{q}_1$ a $\boldsymbol{q}_{k+1}$:

$$\begin{bmatrix} & & \\ & \boldsymbol{A} & \\ & & \end{bmatrix} \begin{bmatrix} & & \\ \boldsymbol{q}_1 & \cdots & \boldsymbol{q}_k \\ & & \end{bmatrix} = \begin{bmatrix} & & \\ \boldsymbol{q}_1 & \cdots & \boldsymbol{q}_{k+1} \\ & & \end{bmatrix} \begin{bmatrix} \boldsymbol{h_{11}} & \cdots & \boldsymbol{h_{1k}} \\ \boldsymbol{h_{21}} & \cdots & \cdot \\ & \ddots & \vdots \\ & & \boldsymbol{h_{k+1,k}} \end{bmatrix} \tag{3}$$

Esto es $\boldsymbol{AQ_k = Q_{k+1}H_{k+1,k}}$. Multipliquemos ambos lados por $\boldsymbol{Q_k^{\mathrm{T}}}$.

$$\boxed{\boldsymbol{Q_k^{\mathrm{T}} A Q_k} = Q_k^{\mathrm{T}} Q_{k+1} H_{k+1,k} = \begin{bmatrix} \boldsymbol{I_{k\times k}} & \mathbf{0}_{k\times 1} \end{bmatrix} \begin{bmatrix} H_k \\ \text{fila } k+1 \end{bmatrix} = \boldsymbol{H_k}.} \tag{4}$$

La matriz cuadrada H_k ha perdido la última fila que estaba en la ecuación (3). Esto nos deja una matriz triangular superior y *una subdiagonal que contiene* de h_{21} *a* $h_{k,k-1}$. Las matrices con solo una subdiagonal distinta de cero se llaman **matrices de Hessenberg**. Esta H_k tiene una linda interpretación:

$\boldsymbol{H_k = Q_k^{\mathrm{T}} A Q_k}$ es la proyección de $\boldsymbol{A}$ en el espacio de Krylov, usando la base de los $\boldsymbol{q}$.

El proceso de Arnoldi para encontrar H_k es uno de los grandes algoritmos del álgebra lineal numérica. Es estable numéricamente y los $\boldsymbol{q}$ son ortonormales.

Autovalores usando Arnoldi

Los números en $H_k = Q_k^{\mathrm{T}} A Q_k$ son calculados durante el proceso de Arnoldi. Si repetimos el proceso $k =$ (tamaño de A) veces, tenemos una matriz de Hessenberg $H = Q^{-1} A Q$ que es **similar a $\boldsymbol{A}$**: mismos autovalores. Calculamos esos autovalores aplicando el algoritmo QR con desplazamientos a H.

En realidad no utilizamos el procedimiento de Arnoldi hasta el final. Nos detenemos en un valor decente de k. Así los k autovalores de H son (a menudo) buenas aproximaciones de los k autovalores extremos de A. Trefethen y Bau hacen hincapié en que para A *no simétrica* quizás ¡no queremos sus autovalores! Los casos mal condicionados llevaron a Trefethen y Embree a la teoría de pseudoespectros.

Sistemas lineales vía Arnoldi y GMRES

Arnoldi nos dio una excelente base ($\boldsymbol{q}$ ortonormales) para los subespacios crecientes de Krylov generados por $\boldsymbol{b}, A\boldsymbol{b}, \ldots, A^{k-1}\boldsymbol{b}$. Así, Arnoldi es el primer paso. En ese subespacio, la idea de GMRES para $A\boldsymbol{x} = \boldsymbol{b}$ es **encontrar el vector $\boldsymbol{x_k}$ que minimiza $||\boldsymbol{b} - \boldsymbol{A x_k}||$**: el RESiduo Mínimo Generalizado (GMRES, por su sigla en inglés). Con una base ortonormal, podemos calcular con precisión y seguridad:

GMRES con la base de Arnoldi $\boldsymbol{q_1, \ldots, q_k}$
Encontrar $\boldsymbol{y_k}$ que minimice la longitud de $\boldsymbol{H_{k+1,k}\, y - (||b||, 0, \ldots, 0)^{\mathrm{T}}}$.
Luego $\boldsymbol{x_k = Q_k y_k}$

Encontrar $\boldsymbol{y_k}$ es un problema de mínimos cuadrados con una matriz de Hessenberg de $k+1$ por k. Los ceros debajo de la primera subdiagonal de $H_{k+1,k}$ hacen que GMRES sea particularmente rápido.

Matrices simétricas : Arnoldi se convierte en Lanczos

Supongamos que nuestra matriz $A = S$ es *simétrica*. En este caso importante, valen dos propiedades adicionales.

1. $\boldsymbol{H_k = Q_k^{\mathrm{T}} S Q_k}$ **también es simétrica**. Su traspuesta es claramente H_k.
2. $\boldsymbol{H_k}$ **es tridiagonal** : solo una diagonal arriba porque hay solo una diagonal debajo.

Una matriz H tridiagonal nos permite ahorrar mucho en costo, la iteración de Arnoldi necesita solo un paso de ortogonalización. Las otras ortogonalidades se dan automáticamente porque H es de **Hessenberg simétrica** (por lo tanto es tridiagonal).

Aquí tenemos Lanczos con letras más simples a_1 a a_k (sobre la diagonal principal) y b_1 a b_{k-1} (sobre las diagonales superior e inferior). Los a y los b en T reemplazan a los h de la matriz de Arnoldi H.

Iteración de Lanczos para $\boldsymbol{Sx = \lambda x}$ (Arnoldi simétrica)

$\boldsymbol{q}_0 = \boldsymbol{0},\ \boldsymbol{q}_1 = \boldsymbol{b}/\|\|\boldsymbol{b}\|\|$	Ortogonalizar $\boldsymbol{b}, S\boldsymbol{b}, S\boldsymbol{b}^2, \ldots$
Para $k = 1, 2, 3, \ldots$	
$\boldsymbol{v} = S\boldsymbol{q}_k$	Comenzar con un nuevo $\boldsymbol{v}$
$a_k = \boldsymbol{q}_k^{\mathrm{T}} \boldsymbol{v}$	El elemento diagonal en T es a_k
$\boldsymbol{v} = \boldsymbol{v} - b_{k-1}\boldsymbol{q}_{k-1} - a_k\,\boldsymbol{q}_k$	Ortogonal a los $\boldsymbol{q}$ anteriores
$b_k = \|\|\boldsymbol{v}\|\|$	Elemento por fuera de la diagonal en T es b_k
$\boldsymbol{q}_{k+1} = \boldsymbol{v}/b_k$	Nuevo vector base

Escribimos T por tridiagonal en lugar de H por Hessenberg, he aquí la información clave para Lanczos. Fueron simplemente copiadas de Arnoldi :

Ecuaciones (3) y (4) $$\boxed{\boldsymbol{T_k = Q_k^{\mathrm{T}} S Q_k} \quad \text{y} \quad \boldsymbol{S Q_k = Q_{k+1}\, T_{k+1,k}}} \tag{5}$$

"Los autovalores de T_k (rápidos de calcular) aproximan a los autovalores de S." ¡ Si tan solo fueran exactos siempre ! Trefethen y Bau crean una matriz diagonal S con 201 autovalores equiespaciados de 0 a 2, y también dos autovalores más grandes 2.5 y 3.0. Comenzando por un vector aleatorio $\boldsymbol{b}$, Lanczos en el paso $k = 9$ aproxima a $\lambda = 2.5$ y a $\lambda = 3.0$ exponencialmente bien. Los otros 7 autovalores de $T_9 = Q_9^{\mathrm{T}} S Q_9$ se amontonan cerca de 0 y 2, pero no se capturan los autovalores individuales en ese grupo de 201 autovalores.

A pesar que las iteraciones de Lanczos exactas garantizarían $\boldsymbol{q}$ ortogonales, el problema es que no lo son. **Lanczos es valioso** pero hay que tener un cuidado especial en mantener, en la práctica, todos los $\boldsymbol{q}$ ortogonales. Lo mismo ocurre con Gram-Schmidt.

Autovalores de una tridiagonal T usando la iteración QR

¿ Cómo se calculan los autovalores de una matriz tridiagonal simétrica T ? Esta es la pregunta clave en el problema simétrico de autovalores. La matriz original S se simplificó a una tridiagonal $T = Q^{\mathrm{T}}SQ = Q^{-1}SQ$ vía Lanczos (no hay cambios en los autovalores porque T es similar a S). Esos ceros podrían también obtenerse usando "rotaciones de Givens" de 2 por 2 .

Ahora tenemos una matriz simétrica tridiagonal $T = T_0$. Casi de la nada apareció una idea fantástica para encontrar sus autovalores :

1. **Con Gram-Schmidt o Householder, factorizar T_0 en QR. Observe que $R = Q^{-1}T_0$.**
2. **Revertir esos factores Q y R para producir $T_1 = RQ = Q^{-1}T_0Q$.**
3. Repetir... Repetir... Repetir...

La nueva $T_1 = Q^{-1}T_0Q$ es similar a T : *mismos autovalores*. Además, la nueva T_1 *sigue siendo tridiagonal* (Problema 6). Así el próximo paso y todos los posteriores siguen siendo rápidos. Lo mejor de todo: **las matrices similares $T, T_1, T_2, \ldots$ se acercan a una matriz diagonal Λ.** Esa matriz diagonal revela los autovalores (intactos) de la matriz original T. El primer autovalor que se va revelando lo hace en la última posición T_{nn}.

Esto se conoce como el "**algoritmo QR**" para calcular autovalores. A medida que ganaba popularidad, causó sensación en el álgebra lineal numérica. Pero los analistas numéricos son personas serias, si les das un buen algoritmo, inmediatamente comienzan a mejorarlo. En este caso tuvieron un éxito rotundo porque las mejoras se obtuvieron casi sin costo (y realmente funcionaban).

El algoritmo mejorado es **QR desplazado**, o **QR con desplazamientos**. El "desplazamiento" resta un múltiplo s_kI de la matriz identidad antes del paso QR, y lo vuelve a sumar luego del paso RQ :

Algoritmo QR con desplazamientos para hallar autovalores	**Elegir un desplazamiento s_k en el paso k** **Factorizar $T_k - s_kI = Q_kR_k$** **Revertir factores y desplazar : $T_{k+1} = R_kQ_k + s_kI$**

Todas las T tienen los mismos autovalores porque son matrices similares. Cada T_{k+1} nueva es $Q_k^{-1}T_kQ_k$. Todavía es simétrica porque $Q_k^{-1} = Q_k^{\mathrm{T}}$:

$$R_k = Q_k^{-1}(T_k - s_kI) \text{ y así } T_{k+1} = Q_k^{-1}(T_k - s_kI)Q_k + s_kI = Q_k^{-1}T_kQ_k. \tag{6}$$

Desplazamientos bien elegidos s_k acelerarán en gran medida el acercamiento de las T a una matriz diagonal Λ. Un buen desplazamiento es $s = T_{nn}$. Wilkinson sugirió un desplazamiento basado en la última submatriz de 2 por 2 de T_k :

Desplazamiento de Wilkinson $\quad s_k =$ el autovalor de $\begin{bmatrix} a_{n-1} & b_{n-1} \\ b_{n-1} & a_n \end{bmatrix}$ más cercano a a_n.

QR desplazada logra convergencia cúbica (muy poco frecuente). En el ejemplo a continuación, los elementos por fuera de la diagonal van de $\operatorname{sen}\theta$ a $-(\operatorname{sen}\theta)^3$. Los n autovalores de una tridiagonal T típica necesitan solo $O(n^3/\epsilon)$ operaciones de punto flotante (*flops*) para una precisión ϵ.

$$\boldsymbol{T_0} = \begin{bmatrix} \cos\theta & \operatorname{sen}\theta \\ \mathbf{sen}\,\boldsymbol{\theta} & 0 \end{bmatrix} \quad \text{Desplazamiento} = 0 \quad \boldsymbol{Q_0} = \begin{bmatrix} \cos\theta & -\operatorname{sen}\theta \\ \operatorname{sen}\theta & \cos\theta \end{bmatrix} \quad \boldsymbol{R_0} = \begin{bmatrix} 1 & \operatorname{sen}\theta\cos\theta \\ 0 & -\operatorname{sen}^2\theta \end{bmatrix}$$

$$\boldsymbol{T_1} = \boldsymbol{R_0}\,\boldsymbol{Q_0} = \begin{bmatrix} \cos\theta(1+\operatorname{sen}^2\theta) & -\operatorname{sen}^3\theta \\ \mathbf{-\,sen^3\,\boldsymbol{\theta}} & -\operatorname{sen}^2\theta\cos\theta \end{bmatrix} \textbf{ ha hecho cúbico el error en un paso.}$$

Cálculo de la SVD

¿ Cuál es la principal diferencia entre el problema simétrico de autovalores $Sx = \lambda x$ y la SVD $A = U\Sigma V^{\mathrm{T}}$? ¿ Cuánto podemos simplificar S y A antes de calcular los λ y los σ ?

Los autovalores son los mismos para S y $Q^{-1}SQ = Q^{\mathrm{T}}SQ$ porque Q es ortogonal.

Tenemos libertad limitada para crear ceros en $Q^{-1}SQ$ (que se mantiene simétrica). Si intentamos poner demasiados ceros en $Q^{-1}S$, la Q final puede destruirlos. La buena $Q^{-1}SQ$ será **tridiagonal** : solo tres diagonales.

Los valores singulares son los mismos para A y $Q_1AQ_2^{\mathrm{T}}$ aún si Q_1 es diferente de Q_2.

Tenemos más libertad para crear ceros en $Q_1AQ_2^{\mathrm{T}}$. Con las Q correctas, será **bidiagonal** (dos diagonales). Rápidamente podemos encontrar Q, Q_1 y Q_2 de tal manera que

$$\boldsymbol{Q^{-1}SQ} = \begin{bmatrix} \boldsymbol{a_1} & \boldsymbol{b_1} & & \\ \boldsymbol{b_1} & \boldsymbol{a_2} & \boldsymbol{b_2} & \\ & \boldsymbol{b_2} & \cdot & \cdot \\ & & \cdot & \boldsymbol{a_n} \end{bmatrix} \begin{matrix} \leftarrow \text{ para los } \lambda \\ \boldsymbol{Q_1AQ_2^{\mathrm{T}}} = \\ \text{para los } \sigma \rightarrow \end{matrix} \begin{bmatrix} \boldsymbol{c_1} & \boldsymbol{d_1} & & \\ 0 & \boldsymbol{c_2} & \boldsymbol{d_2} & \\ & 0 & \cdot & \cdot \\ & & 0 & \boldsymbol{c_n} \end{bmatrix} \tag{7}$$

Sabemos que los valores singulares de A son las raíces cuadradas de los autovalores de $S = A^{\mathrm{T}}A$ y que los valores singulares intactos de $Q_1AQ_2^{\mathrm{T}}$ son las raíces cuadradas de los autovalores intactos de $(Q_1AQ_2^{\mathrm{T}})^{\mathrm{T}}(Q_1AQ_2^{\mathrm{T}}) = Q_2A^{\mathrm{T}}AQ_2^{\mathrm{T}}$. **Multiplicamos (bidiagonal)$^{\mathrm{T}}$(bidiagonal) para obtener una tridiagonal**.

Esto nos ofrece un camino que no debemos tomar. No multiplique $A^{\mathrm{T}}A$ para buscar sus autovalores, es trabajo innecesario y el condicionamiento del problema será elevado al cuadrado. El algoritmo de Golub-Kahan para la SVD funciona directamente sobre A, en dos pasos:

1. Encontrar Q_1 y Q_2 para que $Q_1AQ_2^{\mathrm{T}}$ sea bidiagonal como en (7).
2. Ajustar el algoritmo QR desplazado para preservar los valores singulares de esta matriz bidiagonal.

El paso **1** requiere de $O(mn^2)$ multiplicaciones para llevar una matriz A de m por n a una matriz bidiagonal. Los pasos posteriores solo funcionarán con matrices bidiagonales. Normalmente llevará $O(n^2)$ multiplicaciones encontrar los valores singulares (correctamente hasta casi precisión de máquina). El algoritmo completo se describe en las páginas 489 a 492 de Golub-Van Loan (4ª edición).

Esos números de operaciones son muy aceptables para muchas aplicaciones: la SVD es calculable. Se han propuesto otros algoritmos y son exitosos, pero el costo no es trivial (no se pueden hacer las SVD de a miles). Las próximas secciones de este libro desarrollarán métodos que incluyen el "*muestreo aleatorio*" de la matriz A original. Este enfoque puede aplicarse a matrices realmente grandes. La probabilidad de que los resultados sean precisos es muy alta. La mayoría de los apostadores diría que es innegable un buen resultado a partir de un muestreo aleatorio cuidadoso.

Gradiente conjugado para $Sx = b$

El algoritmo del gradiente conjugado (CG, por su sigla en inglés) se aplica a matrices S simétricas definidas positivas: resuelve $S\boldsymbol{x} = \boldsymbol{b}$. En teoría, da la solución exacta en n pasos (pero esos pasos son más lentos que la eliminación). En la práctica, para matrices grandes, da resultados excelentes mucho antes del n-ésimo paso (este descubrimiento revivió la idea y ahora **CG** es uno de los mejores algoritmos). Tal es la historia del método de Krylov más celebrado.

Punto clave: como S es simétrica, la matriz de Hessenberg H en Arnoldi se convierte en la matriz tridiagonal T en Lanczos. Tres elementos no nulos en las filas y columnas de T hacen al caso simétrico especialmente rápido.

Ahora S no es solo simétrica, también es definida positiva. En ese caso $||\boldsymbol{x}||_S^2 = \boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}$ da una norma muy apropiada (la norma S) para medir el error luego de n pasos. De hecho, la iteración k-ésima $\boldsymbol{x}_k$ del gradiente conjugado tiene una propiedad para destacar:

$\boldsymbol{x_k}$ minimiza el error $||\boldsymbol{x} - \boldsymbol{x_k}||_S$ sobre el k-ésimo subespacio de Krylov
$\boldsymbol{x_k}$ es la mejor combinación de $\boldsymbol{b}, S\boldsymbol{b}, \ldots, S^{\boldsymbol{k-1}}\boldsymbol{b}$

He aquí los pasos de la iteración del gradiente conjugado para resolver $S\boldsymbol{x} = \boldsymbol{b}$:

Iteración del gradiente conjugado para S definida positiva
$\boldsymbol{x}_0 = \boldsymbol{0}, \boldsymbol{r}_0 = \boldsymbol{b}, \boldsymbol{d}_0 = \boldsymbol{r}_0$
para $k = 1$ **a** N

$\alpha_k = (\boldsymbol{r}_{k-1}^{\mathrm{T}}\boldsymbol{r}_{k-1})/(\boldsymbol{d}_{k-1}^{\mathrm{T}} S\boldsymbol{d}_{k-1})$	longitud del paso $\boldsymbol{x}_{k-1}$ a $\boldsymbol{x}_k$
$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \alpha_k \boldsymbol{d}_{k-1}$	solución aproximada
$\boldsymbol{r}_k = \boldsymbol{r}_{k-1} - \alpha_k S\boldsymbol{d}_{k-1}$	residuo nuevo $\boldsymbol{b} - S\boldsymbol{x}_k$
$\beta_k = (\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{r}_k)/(\boldsymbol{r}_{k-1}^{\mathrm{T}}\boldsymbol{r}_{k-1})$	paso
$\boldsymbol{d}_k = \boldsymbol{r}_k + \beta_k \boldsymbol{d}_{k-1}$	siguiente dirección de búsqueda

% *Observe: solo* 1 *multiplicación matriz-vector* $S\boldsymbol{d}$ *en cada paso*

He aquí las dos grandes consecuencias que pueden probarse (con paciencia) de estos pasos. ¡Los zigzags desaparecieron!

1. **Los residuos $\boldsymbol{r_k} = \boldsymbol{b} - S\boldsymbol{x_k}$ son ortogonales: $\boldsymbol{r_k^{\mathrm{T}} r_j} = 0$**
2. **Las direcciones de búsqueda $\boldsymbol{d_k}$ son S-ortogonales: $\boldsymbol{d_k^{\mathrm{T}} S d_j} = 0$**

Observe *Resolver* $S\boldsymbol{x} - \boldsymbol{b} = \boldsymbol{0}$ *es lo mismo que minimizar la cuadrática* $\frac{1}{2}\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x} - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{b}$. Una es el gradiente de la otra. Así el método del gradiente conjugado también es un algoritmo de minimización. Puede generalizarse a ecuaciones no lineales y a funciones de costo no cuadráticas. Podría considerarse en aprendizaje profundo en la Parte VII de este libro, pero las matrices allí son demasiado grandes para usar gradiente conjugado.

Terminamos presentando la acotación más limpia del error luego de k pasos del gradiente conjugado.

$$\textbf{Método CG} \qquad ||\boldsymbol{x} - \boldsymbol{x}_k||_S \leq 2||\boldsymbol{x} - \boldsymbol{x}_0||_S \left(\frac{\sqrt{\lambda_{\text{máx}}} - \sqrt{\lambda_{\text{mín}}}}{\sqrt{\lambda_{\text{máx}}} + \sqrt{\lambda_{\text{mín}}}} \right)^k . \tag{8}$$

Precondicionamiento para $Ax = b$

La idea de precondicionar es encontrar un problema "cercano" que pueda resolverse rápido. Esto es sencillo de decir. Pero *elegir un buen precondicionador es un problema difícil.* Para una matriz A dada, la idea es elegir una matriz P más sencilla que esté cerca de A. Esas matrices pueden estar cerca en el sentido de que $A - P$ tiene una norma o rango pequeños. Trabajar con $P^{-1}A$ es más rápido :

$$\textbf{Precondicionamiento} \qquad P^{-1}A\boldsymbol{x} = P^{-1}\boldsymbol{b} \text{ en lugar de } A\boldsymbol{x} = \boldsymbol{b}. \tag{9}$$

La prueba de convergencia (para cualquier algoritmo que se use) se aplica a $P^{-1}A$ en lugar de A.

Si el algortimo es gradiente conjugado (que funciona con matrices simétricas definidas positivas), es probable que cambiemos de A a $P^{-1/2}AP^{-1/2}$.

He aquí opciones frecuentes para el precondicionador P :

1 $P =$ matriz diagonal (copiando la diagonal principal de A) : *iteración de Jacobi*

2 $P =$ matriz triangular (copiando esa parte de A) : *método de Gauss-Seidel*

3 $P = L_0U_0$ omite rellenado de $A = LU$ (eliminación) para preservar la raleza : *LU incompleta*

4 $P =$ misma matriz de diferencias que A pero en una grilla más gruesa : *método multigrilla*

El método multigrilla es poderoso y está muy desarrollado. Usa una secuencia entera de grillas o mallas. El problema dado en la malla más fina tiene muchos nodos (matriz A grande). Problemas sucesivos en grillas más gruesas tienen menos nodos (matrices más pequeñas). Esos pueden resolverse rápido y los resultados pueden ser interpolados de nuevo a la malla fina. Altamente eficiente con una rápida convergencia.

Iteración de Kaczmarz

La ecuación (10) se ejecuta rápido pero no es fácil de analizar. Ahora sabemos que la convergencia es exponencialmente rápida con alta probabilidad cuando cada paso resuelve una ecuación al azar de $A\boldsymbol{x} = \boldsymbol{b}$. El paso k de Kaczmarz hace que se cumpla la i-ésima ecuación de forma exacta :

$$\boldsymbol{x}_{k+1} \textbf{ satisface } \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} = b_i \qquad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \frac{b_i - \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}_k}{||\boldsymbol{a}_i||^2}\,\boldsymbol{a}_i \tag{10}$$

Cada paso projecta el $\boldsymbol{x}_k$ previo en el plano $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} = b_i$. Kaczmarz clásico consiste en pasar por las m ecuaciones en orden. El algoritmo aleatorizado elije la fila i con probabilidad proporcional a $||\boldsymbol{a}_i||^2$ (**muestreo ponderado por norma al cuadrado en II.4**).

La iteración de Kaczmarz es un ejemplo importante de *descenso por gradiente estocástico* (estocástico porque la ecuación i es una decisión aleatoria en el paso k). Volveremos a este algoritmo en la Sección VI.5 que trata sobre optimización de pesos en aprendizaje profundo.

T. Strohmer y R. Vershynin, *A randomized Kaczmarz algorithm with exponential convergence*, J. Fourier Anal. Appl. **15** (2009) 262-278; arXiv : math/0702226.

Problemas de la Sección II.1

Estos problemas comienzan con una matriz bidiagonal de n por n de diferencias atrasadas $D = I - S$. Dos matrices tridiagonales de diferencias segundas son DD^{T} y $A = -S + 2I - S^{\mathrm{T}}$. El desplazamiento S tiene una subdiagonal no nula $S_{i,i-1} = 1$ para $i = 2, \ldots, n$. A tiene diagonales $-1, 2, -1$.

1 Muestre que DD^{T} es igual a A excepto que $1 \neq 2$ en la posición $(1,1)$. Del mismo modo $D^{\mathrm{T}}D = A$ excepto que $1 \neq 2$ en sus posiciones (n,n).

Observe $A\boldsymbol{u}$ corresponde a $-d^2u/dx^2$ para $0 \leq x \leq 1$ con fronteras fijas $u(0) = 0$ y $u(1) = 0$. DD^{T} cambia la primera condición a $du/dx(0) = 0$ (*libre*). $D^{\mathrm{T}}D$ cambia la segunda condición a $du/dx(1) = 0$ (*libre*). Matrices muy útiles.

2 Muestre que la inversa de $D = I - S$ es D^{-1} = matriz triangular inferior "*suma de unos*". $DD^{-1} = I$ es como el teorema fundamental de cálculo: *la derivada de la integral de f es igual a f*. Multiplique $(D^{-1})^{\mathrm{T}}$ por D^{-1} para hallar $(DD^{\mathrm{T}})^{-1}$ para $n = 4$.

3 El problema 1 dice que $A = DD^{\mathrm{T}} + \boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}$ donde $\boldsymbol{e} = (1, 0, \ldots, 0)$. La Sección III.1 mostrará que $A^{-1} = (DD^{\mathrm{T}})^{-1} - \boldsymbol{z}\boldsymbol{z}^{\mathrm{T}}$. Para $n = 3$, ¿puede descubrir el vector $\boldsymbol{z}$? Un cambio de rango uno en DD^{T} produce un cambio de rango uno en su inversa.

4 Suponga que A se separa en $-S + 2I$ (triangular inferior) y $-S^{\mathrm{T}}$ (triangular superior). La iteración de Gauss-Seidel para resolver $A\boldsymbol{x} = \boldsymbol{b}$ será $(-S + 2I)\,\boldsymbol{x}_{k+1} = S^{\mathrm{T}}\boldsymbol{x}_k + \boldsymbol{b}$.

Esta iteración converge siempre y cuando todos los autovalores de $(-S + 2I)^{-1}\,S^{\mathrm{T}}$ cumplan $|\lambda| < 1$. Encuentre esos autovalores para los tamaños $n = 2$ y $n = 3$.

5 Para $\boldsymbol{b} = (1, 0, 0)$ y $n = 3$, los vectores $\boldsymbol{b}$, $A\boldsymbol{b}$, $A^2\boldsymbol{b}$ son una base no ortogonal de $\mathbf{R}^3$. Aplique la iteración de Arnoldi a la matriz A para producir una base ortonormal $\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3$. Encuentre la matriz H que dé $AQ_2 = Q_3H$ como en la ecuación (3).

6 En el Problema 5, verifique si $Q_2^{\mathrm{T}}AQ_2$ es una matriz tridiagonal.

7 Aplique un paso del *algoritmo QR* a la matriz A de 3 por 3 de diferencias segundas. Los autovalores verdaderos de A son $\lambda = 2 - \sqrt{2}, 2, 2 + \sqrt{2}$.

8 Realice un paso del algoritmo QR con el desplazamiento recomendado $s = A_{33} = 2$.

9 Resuelva $A\boldsymbol{x} = (1, 0, 0)$ a mano. Luego con computadora usando el método del gradiente conjugado.

II.2. Cuadrados mínimos : cuatro enfoques

Muchas aplicaciones llevan a **ecuaciones lineales $Ax = b$ que no tienen solución**. Es irónico que esto sea un problema tan importante en álgebra lineal. No podemos descartar ecuaciones, necesitamos producir una mejor solución $\widehat{x}$. **El método de los cuadrados mínimos elije el $\widehat{x}$ que hace $||b - A\widehat{x}||^2$ tan pequeño como es posible**. Minimizar ese error significa que sus derivadas son cero : esas son las *ecuaciones normales* $A^{\mathrm{T}}A\widehat{x} = A^{\mathrm{T}}b$. Su geometría está en la Figura II.2.

Esta sección explica cuatro maneras de resolver esas ecuaciones importantes (¡ que tienen solución !) :

1 La SVD de A lleva a su **pseudoinversa** A^+. Así $\widehat{x} = A^+b$: una fórmula corta.

2 $A^{\mathrm{T}}A\widehat{x} = A^{\mathrm{T}}b$ puede resolverse directamente cuando A tiene **columnas independientes**.

3 La idea de Gram-Schmidt produce **columnas ortogonales** en Q. Así $A = QR$.

4 **Minimizamos** $||b - Ax||^2 + \delta^2||x||^2$. Esa penalización cambia las ecuaciones normales a $(A^{\mathrm{T}}A + \delta^2 I)x_\delta = A^{\mathrm{T}}b$. Ahora la matriz es invertible y x_δ tiende a $\widehat{x}$ cuando $\delta \to 0$.

$A^{\mathrm{T}}A$ tiene una simetría atractiva, pero su tamaño puede ser un problema. Y su número de condición (que mide el peligro de un error de redondeo inaceptable) es el *cuadrado* del número de condición de A. En problemas bien planteados de tamaño moderado resolvemos la ecuación de mínimos cuadrados vía $A^{\mathrm{T}}A\widehat{x} = A^{\mathrm{T}}b$, pero en problemas grandes o mal planteados encontramos otra manera.

Podríamos ortogonalizar las columnas de A. Podríamos usar su SVD. Para problemas muy grandes hacemos un muestreo del espacio columna de A simplemente multiplicando Av por **vectores aleatorios** v. Este parece ser el futuro de cálculos muy grandes : altas probabilidades de éxito.

En primer lugar, permítanos enfatizar la importancia de $A^{\mathrm{T}}A$ y $A^{\mathrm{T}}CA$. Esa matriz C es a menudo una matriz diagonal positiva. Da las rigideces o conductancias o capacidades de bordes o varianzas inversas $1/\sigma^2$. Son las constantes de la ciencia o la ingeniería o las estadísticas que definen nuestro problema particular: los "pesos" en mínimos cuadrados ponderados.

He aquí una muestra de donde aparece $A^{\mathrm{T}}A$ y $A^{\mathrm{T}}CA$ en matemática aplicada :

En ingeniería mecánica $A^{\mathrm{T}}A$ (o $A^{\mathrm{T}}CA$) es la **matriz de rigidez**

En teoría de circuitos, $A^{\mathrm{T}}A$ (o $A^{\mathrm{T}}CA$) es la **matriz de conductancia**

En teoría de grafos, $A^{\mathrm{T}}A$ (o $A^{\mathrm{T}}CA$) es el **laplaciano del grafo** (ponderado)

En matemática, $A^{\mathrm{T}}A$ es la **matriz de Gram** : productos internos de las columnas de A

En problemas grandes, $A^{\mathrm{T}}A$ es costosa y a menudo peligrosa de calcular. ¡ Lo evitamos siempre que podamos ! Gram-Smith remplaza A por QR (Q ortogonal, R triangular). Así $A^{\mathrm{T}}A$ es la misma que $R^{\mathrm{T}}Q^{\mathrm{T}}QR = R^{\mathrm{T}}R$. Y la ecuación fundamental $A^{\mathrm{T}}A\widehat{x} = A^{\mathrm{T}}b$ se transforma en $R^{\mathrm{T}}R\widehat{x} = R^{\mathrm{T}}Q^{\mathrm{T}}b$. Por último, esto es $R\widehat{x} = Q^{\mathrm{T}}b$, seguro y rápido de resolver.

Por lo tanto $A^{\mathrm{T}}A$ y $A^{\mathrm{T}}CA$ son matrices cruciales, pero paradójicamente tratamos de no calcularlas. Las matrices ortogonales y las matrices triangulares: esas son las buenas.

A^+ es la pseudoinversa de A

Primero describiré la pseudoinversa A^+ con palabras. Si A es invertible entonces A^+ es A^{-1}. Si A es de m por n entonces A^+ es de n por m. Cuando A multiplica un vector $\boldsymbol{x}$ en su espacio fila, este produce $A\boldsymbol{x}$ en el espacio columna. Esos dos espacios tienen igual dimensión r (el rango de A). Restringida a estos espacios, A siempre es invertible, y A^+ invierte A. **Así $\boldsymbol{A^+Ax = x}$ exactamente cuando $\boldsymbol{x}$ está en el espacio fila.** Y $\boldsymbol{AA^+b = b}$ cuando $\boldsymbol{b}$ está en el espacio columna.

El espacio nulo de A^+ es igual al espacio nulo de A^{T}. Contiene los vectores $\boldsymbol{y}$ en $\mathbf{R}^m$ tales que $A^{\mathrm{T}}\boldsymbol{y} = \mathbf{0}$. Esos vectores $\boldsymbol{y}$ son perpendiculares a todos los $A\boldsymbol{x}$ en el espacio columna. Para estos $\boldsymbol{y}$, aceptamos $\boldsymbol{x}^+ = A^+\boldsymbol{y} = \mathbf{0}$ como la mejor solución para la ecuación sin solución $A\boldsymbol{x} = \boldsymbol{y}$. Considerando todo lo dicho, **A^+ invierte A donde sea posible :**

$$\textbf{La pseudoinversa de } \boldsymbol{A} = \begin{bmatrix} \mathbf{2} & 0 \\ 0 & \mathbf{0} \end{bmatrix} \textbf{ es } \boldsymbol{A^+} = \begin{bmatrix} \mathbf{1/2} & 0 \\ 0 & \mathbf{0} \end{bmatrix}.$$

El objetivo es producir una "pseudoinversa" apropiada cuando A no tiene inversa.

Regla 1 Si A tiene columnas independientes, entonces $\boldsymbol{A^+ = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}}$ y así $A^+A = I$.

Regla 2 Si A tiene filas independientes, entonces $\boldsymbol{A^+ = A^{\mathrm{T}}(AA^{\mathrm{T}})^{-1}}$ y así $AA^+ = I$.

Regla 3 Una matriz diagonal Σ se invierte donde puede ser invertida, donde no, Σ^+ tiene ceros :

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\sigma_1} & 0 & 0 & 0 \\ 0 & \boldsymbol{\sigma_2} & 0 & 0 \\ 0 & 0 & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \boldsymbol{\Sigma^+} = \begin{bmatrix} \mathbf{1/\sigma_1} & 0 & 0 \\ 0 & \mathbf{1/\sigma_2} & 0 \\ 0 & 0 & \mathbf{0} \\ 0 & 0 & 0 \end{bmatrix}$$

En los cuatro subespacios $\boldsymbol{\Sigma^+\Sigma = I}$ $\boldsymbol{\Sigma\Sigma^+ = I}$ $\boldsymbol{\Sigma^+\Sigma = 0}$ $\boldsymbol{\Sigma\Sigma^+ = 0}$

Todas las matrices $\boxed{\textbf{La pseudoinversa de } \boldsymbol{A = U\Sigma V^{\mathrm{T}}} \textbf{ es } \boldsymbol{A^+ = V\Sigma^+U^{\mathrm{T}}}.}$ (1)

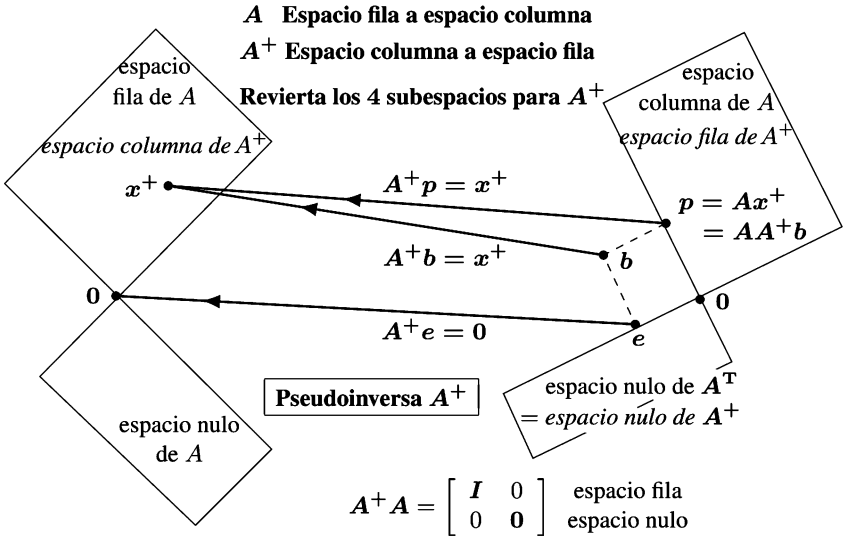


Figura II.1: los vectores $\boldsymbol{p} = A\boldsymbol{x}^+$ en el espacio columna de A son la imagen $\boldsymbol{x}^+$ del espacio fila.

Esta pseudoinversa A^+ (*a veces escrita* $A^\dagger$ con una daga en lugar de un signo más) resuelve la ecuación de mínimos cuadrados $A^{\mathrm{T}}A\widehat{\boldsymbol{x}} = A^{\mathrm{T}}\boldsymbol{b}$ en un paso. Esta página verifica que $\boldsymbol{x}^+ = A^+\boldsymbol{b} = V\Sigma^+U^{\mathrm{T}}\boldsymbol{b}$ es la mejor posible. Al final de esta sección, miraremos más detalladamente a A^+.

Pregunta: La fórmula $A^+ = V\Sigma^+U^{\mathrm{T}}$ usa la SVD. ¿Es la SVD esencial para encontrar A^+? *Respuesta*: No, A^+ también podría calcularse directamente a partir de A modificando los pasos usuales de eliminación para producir A^{-1}. Sin embargo, cada paso debería realizarse con aritmética exacta. Hay que distinguir entre ceros exactos y valores distintos de cero pequeños. Esta es la parte difícil de A^+.

La solución de mínimos cuadrados de $A\boldsymbol{x} = \boldsymbol{b}$ es $\boldsymbol{x}^+ = A^+\boldsymbol{b}$

He escrito $\boldsymbol{x}^+$ en lugar de $\widehat{\boldsymbol{x}}$ porque el vector $\boldsymbol{x}^+$ tiene dos propiedades:

1 $\boldsymbol{x} = \boldsymbol{x}^+ = \boldsymbol{A^+b}$ hace $\|\|\boldsymbol{b} - A\boldsymbol{x}\|\|^2$ tan pequeño como es posible.	**Solución de mínimos cuadrados**
2 Si otro $\widehat{\boldsymbol{x}}$ alcanza ese mínimo entonces $\|\|\boldsymbol{x}^+\\| < \|\|\widehat{\boldsymbol{x}}\|\|$.	**Solución de norma mínima**

$\boldsymbol{x}^+ = \boldsymbol{A^+b}$ es la solución de mínimos cuadrados de norma mínima. Cuando A tiene columnas independientes y rango $r = n$, esta es la única solución de mínimos cuadrados. Pero si hay vectores no nulos $\boldsymbol{x}$ en el espacio nulo de A (es decir si $r < n$), pueden sumarse a $\boldsymbol{x}^+$. El error $\boldsymbol{b} - A(\boldsymbol{x}^+ + \boldsymbol{x})$ no se ve afectado cuando $A\boldsymbol{x} = \boldsymbol{0}$. Pero la longitud $\|\|\boldsymbol{x}^+ + \boldsymbol{x}\|\|^2$ crecerá a $\|\|\boldsymbol{x}^+\|\|^2 + \|\|\boldsymbol{x}\|\|^2$. Esos vectores son ortogonales: espacio fila $\perp$ espacio nulo.

Así la solución de norma mínima (la de menor longitud o la más corta) de $A^{\mathrm{T}}A\widehat{\boldsymbol{x}} = A^{\mathrm{T}}\boldsymbol{b}$ es $\boldsymbol{x}^+ = A^+\boldsymbol{b}$, **$\boldsymbol{x}^+$ tiene componente cero en el espacio nulo de $\boldsymbol{A}$.**

Ejemplo 1 $\boldsymbol{x}^+ = A^+\boldsymbol{b} = \begin{bmatrix} 1/3 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 6 \\ 8 \end{bmatrix} = \begin{bmatrix} \boldsymbol{2} \\ \boldsymbol{0} \end{bmatrix}$ es la solución de mínimos cuadrados más corta de $\begin{bmatrix} 3 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 8 \end{bmatrix}$. Todos los vectores $\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{x_2} \end{bmatrix}$ están en el espacio nulo de A. Todos los vectores $\widehat{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{2} \\ \boldsymbol{x_2} \end{bmatrix}$ minimizan $\|\|\boldsymbol{b} - A\widehat{\boldsymbol{x}}\|\|^2 = 64$. Pero $\boldsymbol{x}^+ = \begin{bmatrix} \boldsymbol{2} \\ \boldsymbol{0} \end{bmatrix}$ es el más corto.

Ese ejemplo muestra las soluciones de mínimos cuadrados cuando A es una matriz diagonal como Σ. Para considerar cualquier matriz $U\Sigma V^{\mathrm{T}}$, hay que tener en cuenta las matrices ortogonales U y V. Podemos multiplicar con libertad por U^{T} sin cambiar ninguna longitud porque $U^{\mathrm{T}}U = I$:

$$\textbf{Error cuadrático} \qquad \|\|\boldsymbol{b} - A\boldsymbol{x}\|\|^2 = \|\|\boldsymbol{b} - U\Sigma V^{\mathrm{T}}\boldsymbol{x}\|\|^2 = \|\|U^{\mathrm{T}}\boldsymbol{b} - \Sigma V^{\mathrm{T}}\boldsymbol{x}\|\|^2. \tag{2}$$

Tomemos $\boldsymbol{w} = V^{\mathrm{T}}\boldsymbol{x}$ para conseguir $\|\|U^{\mathrm{T}}\boldsymbol{b} - \Sigma\boldsymbol{w}\|\|^2$. **El mejor $\boldsymbol{w}$ es $\Sigma^+U^{\mathrm{T}}\boldsymbol{b}$.** Y por último, $\boldsymbol{x}^+$ es $\boldsymbol{A^+b}$:

$$\boldsymbol{w} = V^{\mathrm{T}}\boldsymbol{x}^+ = \Sigma^+U^{\mathrm{T}}\boldsymbol{b} \text{ y } V^{\mathrm{T}} = V^{-1} \text{ hacen que } \boldsymbol{x}^+ = \boldsymbol{V\Sigma^+U^{\mathrm{T}}b = A^+b.} \tag{3}$$

La SVD resolvió el problema de mínimos cuadrados en un paso $\boldsymbol{A^+b}$. La única pregunta es el costo de cálculo. Calcular valores singulares y vectores singulares cuesta más que la eliminación. Las siguientes dos soluciones propuestas funcionan directamente sobre las ecuaciones lineales $A^{\mathrm{T}}A\widehat{\boldsymbol{x}} = A^{\mathrm{T}}\boldsymbol{b}$. Esto funciona cuando $A^{\mathrm{T}}A$ es invertible, y en este caso $\widehat{\boldsymbol{x}}$ es lo mismo que $\boldsymbol{x}^+$.

¿ Cuándo es $A^{\mathrm{T}}A$ invertible ?

La invertibilidad (o no) de la matriz $A^{\mathrm{T}}A$ es una pregunta importante con una linda respuesta :

$A^{\mathrm{T}}A$ es invertible exactamente cuando **A tiene columnas independientes.**
Si $Ax = 0$ entonces $x = 0$

¡ A y $A^{\mathrm{T}}A$ siempre tienen el mismo espacio nulo ! Esto ocurre porque $A^{\mathrm{T}}Ax = 0$ siempre lleva a $x^{\mathrm{T}}A^{\mathrm{T}}Ax = 0$. Esto es $||Ax||^2 = 0$. Entonces $Ax = 0$ y x está en $\mathbf{N}(A)$. Para todas las matrices :

$$\mathbf{N}(A^{\mathrm{T}}A) = \mathbf{N}(A) \text{ y } \mathbf{C}(AA^{\mathrm{T}}) = \mathbf{C}(A) \text{ y } \textbf{rango } (A^{\mathrm{T}}A) = \textbf{rango } (AA^{\mathrm{T}}) = \textbf{rango } (A)$$

Cuando $A^{\mathrm{T}}A$ es invertible podemos directamente resolver las ecuaciones normales $A^{\mathrm{T}}A\widehat{x} = A^{\mathrm{T}}b$.

Las ecuaciones normales $A^{\mathrm{T}}A\widehat{x} = A^{\mathrm{T}}b$

La Figura II.2 muestra una imagen del problema de mínimos cuadrados y su solución. El problema es que b no está en el espacio columna de A, entonces $Ax = b$ no tiene solución. El mejor vector $p = A\widehat{x}$ es una proyección. **Proyectamos b en el espacio columna de A.** Los vectores $\widehat{x}$ y $p = A\widehat{x}$ provienen de resolver un sistema famoso de ecuaciones lineales : $A^{\mathrm{T}}A\widehat{x} = A^{\mathrm{T}}b$. Para invertir $A^{\mathrm{T}}A$, necesitamos saber que A tiene columnas independientes.

La imagen muestra el triángulo rectángulo con lados b, p, y e de vital importancia.

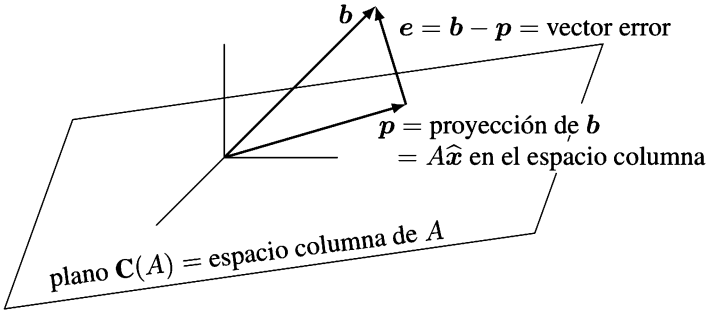


Figura II.2: la proyección $p = A\widehat{x}$ es el punto en el espacio columna de A más cercano a b.

Todos entendemos que e es perpendicular al plano (espacio columna de A). Esto nos dice que $b - p = b - A\widehat{x}$ es perpendicular a todos los vectores Ax en el espacio columna :

$$(Ax)^{\mathrm{T}}\,(b - A\widehat{x}) = x^{\mathrm{T}}A^{\mathrm{T}}(b - A\widehat{x}) = 0 \text{ para todos los } x \text{ implica que } A^{\mathrm{T}}(b - A\widehat{x}) = \mathbf{0}. \quad (4)$$

Cuando reescribimos esta última ecuación como $\boldsymbol{A^{\mathrm{T}}A\,\widehat{x} = A^{\mathrm{T}}b}$, tenemos

Ecuación normal para $\widehat{x}$	$\boldsymbol{A^{\mathrm{T}}A\,\widehat{x} = A^{\mathrm{T}}b}$	(5)
Solución de mínimos cuadrados de $Ax = b$	$\boldsymbol{\widehat{x} = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b}$	(6)
Proyección de b en el espacio columna de A	$\boldsymbol{p = A\,\widehat{x} = A(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b}$	(7)
Matriz de proyección que multiplica a b para obtener p	$\boldsymbol{P = A(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}}$	(8)

Ahora A tiene columnas independientes: $r = n$. Eso hace que $A^{\mathrm{T}}A$ sea definida positiva e invertible. Podríamos verificar que nuestro $\widehat{\boldsymbol{x}}$ sea el mismo vector $\boldsymbol{x}^+ = A^+\boldsymbol{b}$ que obtuvimos de la pseudoinversa. No hay otros $\widehat{\boldsymbol{x}}$ porque se supone que el rango es $r = n$. El espacio nulo de A solo contiene el vector cero.

Las matrices de proyección tienen la propiedad especial $\boldsymbol{P^2 = P}$. Cuando proyectamos una segunda vez, la proyección $\boldsymbol{P}$ se mantiene igual. Usemos la ecuación (8) para P:

$$\boldsymbol{P^2} = A\,(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}A\,(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}} = A\,(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}} = \boldsymbol{P}. \tag{9}$$

La tercera manera de calcular $\widehat{x}$: Gram-Schmidt

Todavía suponemos que las columnas de A son independientes: $r = n$. Pero no suponemos que sean ortogonales. Entonces $A^{\mathrm{T}}A$ no es una matriz diagonal y necesitamos trabajar para resolver $A^{\mathrm{T}}A\widehat{\boldsymbol{x}} = A^{\mathrm{T}}\boldsymbol{b}$. Nuestro tercer enfoque **ortogonaliza las columnas de $\boldsymbol{A}$**, y así $\widehat{\boldsymbol{x}}$ es fácil de encontrar.

Se podría decir que ahora el trabajo está en producir columnas ortogonales (incluso ortonormales). Exactamente. Aunque el número de operaciones es el doble comparado con $A^{\mathrm{T}}A\widehat{\boldsymbol{x}} = A^{\mathrm{T}}\boldsymbol{b}$, los vectores ortogonales dan estabilidad numérica. La estabilidad se vuelve importante cuando $A^{\mathrm{T}}A$ es casi singular. El **número de condición** de $A^{\mathrm{T}}A$ es su norma $||A^{\mathrm{T}}A||$ por $||(A^{\mathrm{T}}A)^{-1}||$. Cuando este número σ_1^2/σ_n^2 es grande, es buena idea ortogonalizar las columnas de A al comienzo. Luego trabajamos con una matriz Q ortogonal.

El número de condición de Q es $||Q||$ por $||Q^{-1}||$. *Esas normas son iguales a* 1*: inmejorable.*

He aquí la famosa idea de Gram-Schmidt, comenzando con $\boldsymbol{A}$ y terminando con $\boldsymbol{Q}$. Las columnas independientes $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ se convierten en $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n$ ortonormales. Este es un cálculo crucial en álgebra lineal. El primer paso es $\boldsymbol{q}_1 = \boldsymbol{a}_1/||\boldsymbol{a}_1||$, que es un vector unitario: $||\boldsymbol{q}_1|| = 1$. Luego restamos de $\boldsymbol{a}_2$ su componente en la dirección $\boldsymbol{q}_1$:

Paso Gram-Schmidt	**Ortogonalizar**	$\boldsymbol{A}_2 = \boldsymbol{a}_2 - (\boldsymbol{a}_2^{\mathrm{T}}\,\boldsymbol{q}_1)\,\boldsymbol{q}_1$	(10)
	Normalizar	$\boldsymbol{q}_2 = \boldsymbol{A}_2/||\boldsymbol{A}_2||$	(11)

Restar esa componente $(\boldsymbol{a}_2^{\mathrm{T}}\,\boldsymbol{q}_1)\,\boldsymbol{q}_1$ produce el vector $\boldsymbol{A}_2$ ortogonal a $\boldsymbol{q}_1$:

$$(\boldsymbol{a}_2 - (\boldsymbol{a}_2^{\mathrm{T}}\,\boldsymbol{q}_1)\,\boldsymbol{q}_1)^{\mathrm{T}}\boldsymbol{q}_1 = \boldsymbol{a}_2^{\mathrm{T}}\,\boldsymbol{q}_1 - \boldsymbol{a}_2^{\mathrm{T}}\,\boldsymbol{q}_1 = \boldsymbol{0} \text{ ya que } \boldsymbol{q}_1^{\mathrm{T}}\,\boldsymbol{q}_1 = 1.$$

El algoritmo sigue con $\boldsymbol{a_3}$, $\boldsymbol{A}_3$ y $\boldsymbol{q_3}$, normalizando cada vez para tener $||\boldsymbol{q}|| = 1$. Restar las componentes de $\boldsymbol{a}_3$ en las direcciones $\boldsymbol{q}_1$ y $\boldsymbol{q}_2$ nos da $\boldsymbol{A}_3$:

$$\textbf{Ortogonalizar}\quad \boldsymbol{A}_3 = \boldsymbol{a}_3 - (\boldsymbol{a}_3^{\mathrm{T}}\,\boldsymbol{q}_1)\,\boldsymbol{q}_1 - (\boldsymbol{a}_3^{\mathrm{T}}\,\boldsymbol{q}_2)\,\boldsymbol{q}_2 \qquad \textbf{Normalizar}\quad \boldsymbol{q_3} = \frac{\boldsymbol{A}_3}{||\boldsymbol{A}_3||} \tag{12}$$

$$\boldsymbol{A_3^{\mathrm{T}}q_1} = \boldsymbol{A_3^{\mathrm{T}}q_2} = \boldsymbol{0} \quad \text{y} \quad ||\boldsymbol{q_3}|| = \boldsymbol{1}$$

Cada $\boldsymbol{q_k}$ es una combinación de $\boldsymbol{a_1}$ a $\boldsymbol{a_k}$. Así, cada $\boldsymbol{a_k}$ es una combinación de $\boldsymbol{q_1}$ a $\boldsymbol{q_k}$.

$$\textbf{Los } \boldsymbol{a} \textbf{ a partir de los } \boldsymbol{q} \qquad \begin{aligned} \boldsymbol{a}_1 &= ||\boldsymbol{a}_1||\,\boldsymbol{q}_1 \\ \boldsymbol{a}_2 &= (\boldsymbol{a}_2^{\mathrm{T}}\,\boldsymbol{q}_1)\,\boldsymbol{q}_1 + ||\boldsymbol{A}_2||\,\boldsymbol{q}_2 \\ \boldsymbol{a}_3 &= (\boldsymbol{a}_3^{\mathrm{T}}\,\boldsymbol{q}_1)\,\boldsymbol{q}_1 + (\boldsymbol{a}_3^{\mathrm{T}}\,\boldsymbol{q}_2)\,\boldsymbol{q}_2 + ||\boldsymbol{A}_3||\,\boldsymbol{q}_3 \end{aligned} \tag{13}$$

Esas ecuaciones nos dicen que **la matriz $R = Q^{\mathrm{T}}A$ con $r_{ij} = q_i^{\mathrm{T}}a_j$ es triangular superior :**

$$\begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix} \quad \text{es } A = QR. \tag{14}$$

Gram-Schmidt produce q ortonormales a partir de a independientes. Entoces $A = QR$.

Si $A = QR$ entonces $R = Q^{\mathrm{T}}A =$ productos internos de los q con los a. R es triangular, dado que las a siguientes no están involucrados en q anteriores. Y sin duda $A^{\mathrm{T}}A = R^{\mathrm{T}}Q^{\mathrm{T}}QR = R^{\mathrm{T}}R$:

La solución de mínimos cuadrados de $Ax = b$ es $\widehat{x} = R^{-1}Q^{\mathrm{T}}b$.

El comando de MATLAB es $[Q,\ R] = qr(A)$. Debido a que $R = Q^{\mathrm{T}}A$, se tiene que $r_{ij} = q_i^{\mathrm{T}}a_j$. El vector $\widehat{x} = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b$ es $(R^{\mathrm{T}}R)^{-1}R^{\mathrm{T}}Q^{\mathrm{T}}b$. Esto es exactamente $\widehat{x} = R^{-1}Q^{\mathrm{T}}b$.

Gram-Schmidt con pivoteo por columnas

Esa descripción sin rodeos de Gram-Schmidt funcionó con las columnas de A en su orden original $a_1, a_2, a_3, \ldots$ Esto podría ser peligroso. No podríamos sobrevivir con un código de eliminación que no permita intercambio de filas. El error de redondeo podría aniquilarnos.

De modo similar, cada paso de Gram-Schmidt debería comenzar con una nueva columna que sea lo más independiente posible de las columnas ya procesadas. **Necesitamos intercambiar columnas para elegir la columna más grande entre las restantes. Cambiar el orden de las columnas a medida que avanzamos.**

Para elegir correctamente entre las columnas restantes de A, hacemos un cambio sencillo en Gram-Schmidt :

Antes Aceptar la columna a_j siguiente. Restar sus componentes en las direcciones q_1 a q_{j-1}

Ahora Cuando se encuentra q_{j-1}, restar la componente q_{j-1} de **todas las columnas restantes**

Parece más trabajo, pero no lo es. Tarde o temprano tenemos que restar $(a_i^{\mathrm{T}}q_{j-1})q_{j-1}$ de cada columna restante a_i. Ahora lo hacemos antes: tan pronto conocemos q_{j-1}. Así podemos elegir con libertad con qué columna trabajar a continuación, y elegimos la más grande.

Eliminación Intercambios de fila en A nos dieron $PA = LU$ (matriz de permutación P)
Gram-Schmidt Intercambios de columna nos dan $AP = QR$ (matriz de permutación P)

Así es la situación luego de $j-1$ pasos de Gram-Schmidt con pivoteo por columas. Tenemos $j-1$ vectores unitarios ortogonales q_1 a q_{j-1} en las columnas de una matriz Q_{j-1}. Tenemos la matriz cuadrada R_{j-1} que combina esas columnas de Q_{j-1} para producir $j-1$ columnas de A. *Pueden no ser las* primeras $j-1$ *columnas de* A (estamos optimizando el orden de las columnas). Todas las columnas restantes de A han sido *ortogonalizadas* con respecto a los vectores q_1 a q_{j-1}.
Paso j. De las columnas restantes de A elegir la de mayor norma. Normalizarla a longitud 1.

Esto es q_j. Luego, a cada uno de los $n-j$ vectores que todavía esperan ser elegidos, se le resta la componente en dirección a este último q_j. Ahora estamos listos para el paso $j+1$.

Seguiremos las notas del curso de Gunnar Martinsson de 2016 para APPM 5720, para expresar el paso j en pseudocódigo. La A original es A_0 y las matrices Q_0 y R_0 están vacías.

El paso j es el siguiente bucle, que comienza con A_{j-1} y termina en A_j. El código termina luego de que j alcance el $\min(m, n)$. **He aquí el pivoteo por columnas en Gram-Schmidt** :

$\boldsymbol{i}$ = argmax $||A_{j-1}(:, \ell)||$ encuentra la columna más grande que aún no haya sido elegida para la base

$\boldsymbol{q}_j = A_{j-1}(:, i)/||A_{j-1}(:, i)||$ normaliza esa columna para obtener el nuevo vector unitario $\boldsymbol{q}_j$

$Q_j = \begin{bmatrix} Q_{j-1} & \boldsymbol{q}_j \end{bmatrix}$ actualiza Q_{j-1} con el nuevo vector unitario ortogonal $\boldsymbol{q}_j$

$\boldsymbol{r}_j = \boldsymbol{q}_j^{\mathrm{T}} A_{j-1}$ calcula la fila de productos internos de $\boldsymbol{q}_j$ con las columnas restantes de $\boldsymbol{A}$

$R_j = \begin{bmatrix} R_{j-1} \\ \boldsymbol{r}_j \end{bmatrix}$ actualiza R_{j-1} con la nueva fila de productos internos

$A_j = A_{j-1} - \boldsymbol{q}_j \boldsymbol{r}_j$ resta la nueva matriz de rango uno de cada columna para obtener A_j

Cuando termina este bucle o lazo, tenemos Q, R y $A = QR$. Esta R es una *permutación* de una matriz triangular superior. (Será triangular superior si las columnas más grandes en el Paso 1 están primeras, así cada $i = j$.) El resultado puede ser una matriz triangular superior y un vector con los números $1, \ldots, n$ con el orden de permutación que necesitamos conocer para construir R.

En la práctica, este algoritmo QR con pivoteo es más seguro si se lo *reortonormaliza* :

$$\boldsymbol{q}_j = \boldsymbol{q}_j - Q_{j-1}(Q_{j-1}^{\mathrm{T}} \boldsymbol{q}_j)$$
$$\boldsymbol{q}_j = \boldsymbol{q}_j \,/\, ||\boldsymbol{q}_j|| \quad (\textbf{¡para asegurarnos!})$$

Existe un reordenamiento similar para "QR con matrices de Householder" para reducir el error de redondeo. Hemos identificado el punto esencial de pivoteo : **las columnas buenas tienen prioridad.**

Pregunta : ambas matrices, la Q de Gram-Schmidt y la U de la SVD contienen una base ortonormal para el espacio columna **C**(A). *¿ Serán la misma base* ?

Respuesta : No, no son la misma base. Las columnas de U son autovectores de AA^{T}. No se pueden encontrar autovectores (o autovalores) en un número finito de pasos "aritméticos" exactos para matrices de tamaño $n > 4$. La ecuación $\det(A - \lambda I) = 0$ es de grado mayor o igual a 5 : *no existe una fórmula para encontrar las raíces* λ *de una ecuación de grado* 5 *(Abel)*. Gram-Schmidt solo requiere de productos internos y raíces cuadradas, así que Q debe ser distinta de U.

En el pasado, calcular un autovalor casi exacto requería de un número de operaciones de punto flotante igual a un múltiplo de n^3 mucho mayor que el múltiplo de la eliminación o de Gram-Schmidt. Esto ya no es así en nuestros días.

Otro camino a Q : reflexiones de Householder

Sin intercambios de columnas, Gram-Schmidt resta de cada vector $\boldsymbol{a}_j$ sus componentes en las direcciones $\boldsymbol{q}_1$ a $\boldsymbol{q}_{j-1}$ que ya están fijos. Para la estabilidad numérica esas restas deben hacerse *de a una*. He aquí $\boldsymbol{q}_3$ con *dos restas separadas a partir de* $\boldsymbol{a}_3$:

$$\text{Calcular} \quad \boldsymbol{a_3} - (\boldsymbol{a_3^{\mathrm{T}} q_1})\, \boldsymbol{q}_1 = \boldsymbol{a_3'} \quad \text{y} \quad \boldsymbol{a_3'} - (\boldsymbol{a_3'^{\mathrm{T}} q_2})\, \boldsymbol{q}_2 = \boldsymbol{A}_3 \quad \text{y} \quad \boldsymbol{q_3} = \boldsymbol{A_3}/||\boldsymbol{A_3}||.$$

Pero todavía el $\boldsymbol{q}_3$ calculado no es *exactamente* ortogonal a $\boldsymbol{q}_1$ y $\boldsymbol{q}_2$. Es difícil. La manera de crear una Q exactamente ortogonal es construirla a la manera de Householder :

$$\boxed{\textbf{Matriz de reflexión de Householder} \qquad \boldsymbol{H} = I - 2\,\frac{\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}}{||\boldsymbol{v}||^2} = I - 2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}.} \tag{15}$$

$\boldsymbol{u}$ es el vector unitario $\boldsymbol{v}/||\boldsymbol{v}||$. Luego, $\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} = \boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}/||\boldsymbol{v}||^2$. $\boldsymbol{H}$ **es simétrica y ortogonal.**

$$H^{\mathrm{T}}H = (I - 2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}})^2 = I - 4\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} + 4\boldsymbol{u}(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u})\,\boldsymbol{u}^{\mathrm{T}} = I. \tag{16}$$

Punto clave: **si** $\boldsymbol{v} = \boldsymbol{a} - \boldsymbol{r}$ **y** $||\boldsymbol{a}|| = ||\boldsymbol{r}||$ **entonces** $\boldsymbol{H}\boldsymbol{a} = \boldsymbol{r}$ (vea Problema 6). Para producir ceros en la columna k debajo de la diagonal principal, usamos esta H_k con $\boldsymbol{v} = (\boldsymbol{a}_{\text{inferior}} - \boldsymbol{r}_{\text{inferior}})$ y $\boldsymbol{u} = \boldsymbol{v}/||\boldsymbol{v}||$. **Estamos creando ceros en** $\boldsymbol{HA}$. Y tenemos opciones de signos en $\boldsymbol{r}$:

$$\boldsymbol{H_k}[\,\textbf{columna } \boldsymbol{k}\,] = \begin{bmatrix} I & \\ & I - 2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{a}_{\text{superior}} \\ \boldsymbol{a}_{\textbf{inferior}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}_{\text{superior}} \\ \pm\,||\boldsymbol{a}_{\textbf{inferior}}|| \\ \boldsymbol{n-k}\ \textbf{ceros} \end{bmatrix} = \boldsymbol{r_k} \tag{17}$$

Comenzamos con la matriz A. El primer paso produce H_1A y el siguiente paso calcula H_2H_1A. Recorriendo las columnas, las reflexiones H_1 a H_{n-1} se multiplican para obtener Q. Y en orden inverso, esas reflexiones crean ceros en A y producen una matriz triangular R:

$$\boxed{H_{n-1}\ldots H_2H_1A = \begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \ldots & \boldsymbol{r}_n \end{bmatrix} \quad \text{se convierte en} \quad Q^{\mathrm{T}}A = R.} \tag{18}$$

La clave es mantener un registro de la H_j almacenando solo el vector $\boldsymbol{v}_j = \boldsymbol{a}_j - \boldsymbol{r}_j$, no la matriz. Así cada $H_j = I - 2\boldsymbol{v}_j\boldsymbol{v}_j^{\mathrm{T}}/||\boldsymbol{v}_j||^2$ es exactamente ortogonal. Para resolver $A\boldsymbol{x} = \boldsymbol{b}$ por mínimos cuadrados, podemos comenzar con la matriz $[A \quad \boldsymbol{b}]$ como en la eliminación. Multiplicamos por todas las H para llegar a $[R \quad Q^{\mathrm{T}}\boldsymbol{b}]$. Después resolvemos el sistema triangular $R\widehat{\boldsymbol{x}} = Q^{\mathrm{T}}\boldsymbol{b}$ simplemente por sustitución hacia atrás. **Hemos encontrado la solución de mínimos cuadrados** $\widehat{\boldsymbol{x}} = \boldsymbol{R^{-1}Q^{\mathrm{T}}b}$.

Ejemplo $A = \begin{bmatrix} 4 & * \\ 3 & * \end{bmatrix}$ tiene $\boldsymbol{a} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$ y $\boldsymbol{r} = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$ y $||\boldsymbol{a}|| = ||\boldsymbol{r}||$

$$\text{Elegimos} \quad \boldsymbol{v} = \boldsymbol{a} - \boldsymbol{r} = \begin{bmatrix} -1 \\ 3 \end{bmatrix} \quad \text{y} \quad \boldsymbol{u} = \frac{\boldsymbol{v}}{||\boldsymbol{v}||} = \frac{1}{\sqrt{10}} \begin{bmatrix} -1 \\ 3 \end{bmatrix}$$

$$\text{Luego} \quad \boldsymbol{H} = I - 2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} = \frac{1}{5} \begin{bmatrix} 4 & 3 \\ 3 & -4 \end{bmatrix} = \boldsymbol{Q}^{\mathrm{T}} \quad \text{y} \quad \boldsymbol{HA} = \begin{bmatrix} \boldsymbol{5} & * \\ \boldsymbol{0} & * \end{bmatrix} = \boldsymbol{R}$$

Mínimos cuadrados con un término de penalización

Si A tiene columnas dependientes y $A\boldsymbol{x} = \mathbf{0}$ tiene soluciones no triviales, entonces $A^{\mathrm{T}}A$ no puede ser invertible. Aquí necesitamos A^{+}. Un enfoque moderado "regularizará" mínimos cuadrados:

$$\boxed{\textbf{Penalización} \quad \text{Minimizar } ||A\boldsymbol{x}-\boldsymbol{b}||^2+\delta^2\,||\boldsymbol{x}||^2 \quad \text{Resolver } (\boldsymbol{A^{\mathrm{T}}A+\delta^2 I})\,\widehat{\boldsymbol{x}} = A^{\mathrm{T}}\boldsymbol{b}} \tag{19}$$

Este cuarto enfoque de mínimos cuadrados se llama *regresión ridge*. Mostraremos que $\widehat{\boldsymbol{x}}$ se acerca a la solución de menor norma $\boldsymbol{x}^{+} = A^{+}\boldsymbol{b}$ a medida que desaparece la penalización (δ se va a cero).

La Sección III.4 describe una penalización diferente: **sumar la norma $\boldsymbol{\ell^1}$, $\boldsymbol{\lambda||x||_1}$.** El resultado es hermoso: en lugar de obtener el vector $\boldsymbol{x}^{+}$ de norma mínima, *la norma ℓ^1 da soluciones ralas.*

La pseudoinversa A^{+} es el límite de $(A^{\mathrm{T}}A+\delta^2 I)^{-1}A^{\mathrm{T}}$

La ecuación (19) crea la pseudoinversa A^{+} a partir de las matrices definidas positivas $A^{\mathrm{T}}A+\delta^2 I$. Estas son matrices invertibles para $\delta > 0$ excepto en el último minuto, cuando $\delta = 0$. En ese momento tenemos un salto que nos lleva a A^{+}. Es más fácil de ver si A es una matriz de 1 por 1 (*solo un número* σ):

$$\textbf{Para } \boldsymbol{\delta > 0} \quad (A^{\mathrm{T}}A+\delta^2 I)^{-1}A^{\mathrm{T}} = \left[\frac{\boldsymbol{\sigma}}{\boldsymbol{\sigma^2+\delta^2}}\right] \text{ es de 1 por 1} \quad \textbf{Ahora hagamos } \boldsymbol{\delta \to 0}$$

El límite es cero si $\boldsymbol{\sigma = 0}$. El límite es $\boldsymbol{\frac{1}{\sigma}}$ si $\boldsymbol{\sigma \neq 0}$. Esto es exactamente $\boldsymbol{A^{+}}$ **= cero o** $\boldsymbol{\frac{1}{\sigma}}$.

Ahora permitimos cualquier matriz diagonal Σ. Esto es sencillo porque todas las matrices son diagonales. Vemos el caso de 1 por 1 en cada posición de la diagonal principal. Σ tiene elementos positivos σ_1 a σ_r y el resto todos ceros. La penalización hace que toda la diagonal sea positiva:

$$(\Sigma^{\mathrm{T}}\Sigma+\delta^2 I)^{-1}\Sigma^{\mathrm{T}} \text{ tiene elementos diagonales positivos } \frac{\boldsymbol{\sigma_i}}{\boldsymbol{\sigma_i^2+\delta^2}} \text{ y el resto todos ceros.}$$

Los números positivos se acercan a $\boldsymbol{\frac{1}{\sigma_i}}$. Los ceros siguen siendo ceros. Cuando $\boldsymbol{\delta \to 0}$, el límite es nuevamente $\boldsymbol{\Sigma^{+}}$.

Para demostrar que para *toda matriz* A el límite es A^{+}, usamos la SVD: $A = U\Sigma V^{\mathrm{T}}$. Reemplazamos esta matriz A por $(A^{\mathrm{T}}A+\delta^2 I)^{-1}A^{\mathrm{T}}$. Las matrices ortogonales U y V se cancelan porque $U^{\mathrm{T}} = U^{-1}$ y $V^{\mathrm{T}} = V^{-1}$:

$$A^{\mathrm{T}}A+\delta^2 I = V\Sigma^{\mathrm{T}}U^{\mathrm{T}}U\Sigma V^{\mathrm{T}}+\delta^2 I = V(\Sigma^{\mathrm{T}}\Sigma+\delta^2 I)V^{\mathrm{T}}$$

$$(A^{\mathrm{T}}A+\delta^2 I)^{-1}A^{\mathrm{T}} = V(\Sigma^{\mathrm{T}}\Sigma+\delta^2 I)^{-1}V^{\mathrm{T}}V\Sigma^{\mathrm{T}}U^{\mathrm{T}} = \boldsymbol{V}\left[(\Sigma^{\mathrm{T}}\Sigma+\delta^2 I)^{-1}\Sigma^{\mathrm{T}}\right]\boldsymbol{U^{\mathrm{T}}}$$

Ahora es el momento de tomar $\delta \to 0$. Las matrices V y U^{T} se mantienen en sus lugares. **La matriz diagonal entre corchetes se acerca a $\boldsymbol{\Sigma^{+}}$** (este es exactamente el caso diagonal que se estableció antes). **El límite de $\boldsymbol{(A^{\mathrm{T}}A+\delta^2 I)^{-1}A^{\mathrm{T}}}$ es nuestra pseudoinversa $\boldsymbol{A^{+}}$.**

$$\boxed{\lim_{\delta \to 0} \; \boldsymbol{V}\left[(\boldsymbol{\Sigma^{\mathrm{T}}\Sigma+\delta^2 I})^{-1}\boldsymbol{\Sigma^{\mathrm{T}}}\right]\boldsymbol{U^{\mathrm{T}}} = \boldsymbol{V\Sigma^{+}U^{\mathrm{T}}} = \boldsymbol{A^{+}}.} \tag{20}$$

La dificultad al calcular A^+ es distinguir si un valor singular es **cero o muy pequeño**. El elemento diagonal correspondiente en Σ^+ es cero o extremadamente grande. Σ^+ y A^+ están lejos de ser funciones continuas de Σ y A. Lo valioso de A^+ es que nos advierte cuando λ o σ está muy cerca de cero: a menudo lo tratamos como cero sin tener la certeza.

He aquí matrices pequeñas y sus pseudoinversas, para que quede claro que A^+ no sigue todas las reglas de A^{-1}. Es *discontinua* donde algún valor singular vale cero, es *discontinua*.

$$\textbf{De 0 a } \mathbf{2^{10}} \quad \begin{bmatrix} 2 & 0 \\ 0 & \mathbf{0} \end{bmatrix}^+ = \begin{bmatrix} 1/2 & 0 \\ 0 & \mathbf{0} \end{bmatrix} \quad \text{pero} \quad \begin{bmatrix} 2 & 0 \\ 0 & \mathbf{2^{-10}} \end{bmatrix}^+ = \begin{bmatrix} 1/2 & 0 \\ 0 & \mathbf{2^{10}} \end{bmatrix}$$

No es cierto que $(AB)^+ = B^+A^+$. Las pseudoinversas no obedecen todas las reglas de las matrices inversas. Aunque sí obedecen $(A^{\mathrm{T}})^+ = (A^+)^{\mathrm{T}}$ y $(A^{\mathrm{T}}A)^+ = A^+(A^{\mathrm{T}})^+$.

Si $A = \begin{bmatrix} 1 & 0 \end{bmatrix}$ y $B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, entonces $(AB)^+$ no es igual a B^+A^+ :

$$AB = \begin{bmatrix} 1 \end{bmatrix} \text{ y } (\boldsymbol{AB})^+ = \begin{bmatrix} \mathbf{1} \end{bmatrix} \text{ pero } B^+ = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix} \text{ y } A^+ = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ y } \boldsymbol{B^+A^+} = \begin{bmatrix} \frac{\mathbf{1}}{\mathbf{2}} \end{bmatrix}.$$

Si C tiene rango completo e igual al número de columnas y R tiene rango completo igual al número de filas entonces sí es cierto que $(CR)^+ = R^+C^+$.

Este es un hecho asombrosamente útil. Significa que la pseudoinversa de cualquier matriz puede calcularse sin conocer su SVD (y sin calcular ningún autovalor). He aquí el razonamiento. El primer paso apareció en la Sección I.1, página 4 del libro.

Toda matriz A de m por n de rango r puede factorizarse como $A = CR = (m \times r)(r \times n)$.

La matriz C da una base del espacio columna. R da una base del espacio fila.

$\boldsymbol{C^+} = (C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}} =$ inversa a izquierda de C.
$\boldsymbol{R^+} = R^{\mathrm{T}}(RR^{\mathrm{T}})^{-1} =$ inversa a derecha de R.

Entonces para $A = CR$ vale $\boldsymbol{A^+ = R^+C^+}$ sin calcular autovalores o valores singulares.

La trampa es que necesitamos cálculos exactos (sin redondeo) para conocer el rango r exacto. La pseudoinversa es discontinua cuando el rango disminuye repentinamente. El gran valor $1/\sigma$ de repente se vuelve cero. $\mathbf{0^+ = 0}$ siempre.

En el ejemplo de arriba con $(AB)^+ \neq B^+A^+$, se puede verificar que si ponemos esas matrices en orden inverso entonces es cierto que $(BA)^+ = A^+B^+$.

La pseudoinversa también es conocida como la inversa Moore-Penrose. En MATLAB es pinv ($\boldsymbol{A}$).

Cuadrados mínimos ponderados

Al elegir minimizar el error $||\boldsymbol{b} - A\boldsymbol{x}||^2$, estamos suponiendo de manera implícita que todas las observaciones $b_1, \ldots, b_m$ son *igualmente confiables*. Los errores en b_i tienen media = valor promedio $= 0$, y suponemos además que las varianzas son iguales. Esa suposición podría ser falsa. Algunos b_i pueden tener menos ruido y más precisión. Las varianzas $\sigma_1^2, \ldots, \sigma_m^2$ de esas m mediciones pueden no ser iguales. En este caso deberíamos asignar mayores pesos o ponderaciones a los datos más confiables (los b con menor varianza).

La opción natural es *dividir la k-ésima ecuación por* σ_k. Así b_k/σ_k tiene varianza 1. Todas las observaciones quedan normalizadas a varianza unitaria. Observe que σ es la notación aceptada para la raíz cuadrada de la varianza (vea la Sección V.1 para varianzas y V.4 para covarianzas). Aquí σ^2 es una varianza y *no* un valor singular de A.

Cuando las observaciones b_k son independientes, todas las covarianzas son cero. Los únicos valores distintos de cero en la matriz de varianza-covarianza C son $\sigma_1^2, \ldots \sigma_m^2$ en la diagonal. Así nuestros pesos $1/\sigma_k$ han multiplicado $A\boldsymbol{x} = \boldsymbol{b}$ por la matriz $C^{-1/2}$.

Multiplicar por $\boldsymbol{C^{-1/2}}$ sigue siendo la opción correcta cuando $\boldsymbol{C}$ tiene valores no nulos por fuera de la diagonal.

Estamos "blanqueando" los datos. La cantidad para minimizar no es $||\boldsymbol{b} - A\boldsymbol{x}||^2$. Ese error debería pesarse o ponderarse por C^{-1}.

Mínimos cuadrados ponderados minimiza $\boldsymbol{||C^{-1/2}(b - Ax)||^2 = (b - Ax)^{\mathrm{T}} C^{-1}(b - Ax).}$

Ahora la ecuación normal $A^{\mathrm{T}}A\widehat{\boldsymbol{x}} = A^{\mathrm{T}}\boldsymbol{b}$ para el mejor $\widehat{\boldsymbol{x}}$ incluye C^{-1} = covarianzas inversas:

$$\boxed{\textbf{Ecuación normal ponderada} \qquad \boldsymbol{A^{\mathrm{T}} C^{-1} A\widehat{x} = A^{\mathrm{T}} C^{-1} b.}} \tag{21}$$

Ejemplo 2 Sean $x = b_1$ y $x = b_2$ mediciones independientes con ruido del número x. Multiplicamos esas ecuaciones por sus pesos $1/\sigma$.

Resolvemos $Ax = \begin{bmatrix} 1 \\ 1 \end{bmatrix} x = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$ por mínimos cuadrados ponderados. Los pesos son $1/\sigma_1$ y $1/\sigma_2$.

Las ecuaciones se transforman en $x/\sigma_1 = b_1/\sigma_1$ y $x/\sigma_2 = b_2/\sigma_2$: $\begin{bmatrix} 1/\sigma_1 \\ 1/\sigma_2 \end{bmatrix} x = \begin{bmatrix} b_1/\sigma_1 \\ b_2/\sigma_2 \end{bmatrix}$

$$\textbf{Ecuación normal ponderada} \qquad \left[\ \frac{\boldsymbol{1}}{\boldsymbol{\sigma_1^2}} + \frac{\boldsymbol{1}}{\boldsymbol{\sigma_2^2}}\ \right] \widehat{\boldsymbol{x}} = \frac{\boldsymbol{b_1}}{\boldsymbol{\sigma_1^2}} + \frac{\boldsymbol{b_2}}{\boldsymbol{\sigma_2^2}}. \tag{22}$$

La mejor estimación estadística de x es un promedio ponderado de b_1 y b_2:

$$\widehat{\boldsymbol{x}} = \left(\frac{\sigma_1^2\,\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right)\left(\frac{b_1}{\sigma_1^2} + \frac{b_2}{\sigma_2^2}\right) = \frac{\boldsymbol{\sigma_2^2\, b_1 + \sigma_1^2\, b_2}}{\boldsymbol{\sigma_1^2 + \sigma_2^2}}$$

Problemas de la Sección II.2

1 (Una nueva prueba de que $\mathbf{N}(A^{\mathrm{T}}A) = \mathbf{N}(A)$.) Supongamos que $A^{\mathrm{T}}A\boldsymbol{x} = \mathbf{0}$. Entonces $A\boldsymbol{x}$ está en el espacio nulo de A^{T}. Pero $A\boldsymbol{x}$ siempre está en el espacio columna de A. Esos dos subespacios son ortogonales, por lo tanto *si* $A^{\mathrm{T}}A\boldsymbol{x} = \mathbf{0}$ *entonces* $A\boldsymbol{x} = \mathbf{0}$.

Demuestre la afirmación opuesta para obtener $\mathbf{N}(A^{\mathrm{T}}A) = \mathbf{N}(A)$.

2 ¿Por qué A y A^+ tienen el mismo rango? Si A es cuadrada, ¿tienen A y A^+ los mismos autovectores? ¿Cuáles son los autovalores de A^+?

3 Usando A y A^+ muestre que tiene sentido el producto A^+A y que $(A^+A)^2 = A^+A =$ proyección.

$$A = \sum \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^{\mathrm{T}} \qquad A^+ = \sum \frac{\boldsymbol{v}_i \boldsymbol{u}_i^{\mathrm{T}}}{\sigma_i} \qquad A^+A = \sum \boldsymbol{v}_i \boldsymbol{v}_i^{\mathrm{T}} \qquad AA^+ = \sum \boldsymbol{u}_i \boldsymbol{u}_i^{\mathrm{T}}$$

4 ¿Qué matrices tienen $A^+ = A$? ¿Por qué son cuadradas? Mire A^+A.

5 Suponga que A tiene columnas independientes (rango $r = n$; espacio nulo = vector cero).

(a) Describa la matriz Σ de m por n en $A = U\Sigma V^{\mathrm{T}}$. ¿Cuántos valores no nulos hay en Σ?

(b) Muestre que $\Sigma^{\mathrm{T}}\Sigma$ es invertible encontrando su inversa.

(c) Escriba la matriz $(\Sigma^{\mathrm{T}}\Sigma)^{-1}\Sigma^{\mathrm{T}}$ de n por m e identifíquela como $\boldsymbol{\Sigma^+}$.

(d) Reemplace $A = U\Sigma V^{\mathrm{T}}$ en $(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}$ e identifique esa matriz como $\boldsymbol{A^+}$.

$\boldsymbol{A^{\mathrm{T}}A\widehat{x} = A^{\mathrm{T}}b}$ **conduce a** $\boldsymbol{A^+ = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}}$**, pero solo si** $\boldsymbol{A}$ **tiene rango** $\boldsymbol{n}$**.**

6 La matriz de Householder H en la ecuación (17) elige $\boldsymbol{v} = \boldsymbol{a} - \boldsymbol{r}$ con $||\boldsymbol{a}||^2 = ||\boldsymbol{r}||^2$. Verifique que esta elección del vector $\boldsymbol{v}$ siempre da $H\boldsymbol{a} = \boldsymbol{r}$:

$$\text{Verifique que } \quad H\boldsymbol{a} = \boldsymbol{a} - 2\,\frac{(\boldsymbol{a}-\boldsymbol{r})\,(\boldsymbol{a}-\boldsymbol{r})^{\mathrm{T}}}{(\boldsymbol{a}-\boldsymbol{r})^{\mathrm{T}}(\boldsymbol{a}-\boldsymbol{r})}\boldsymbol{a} \quad \text{ es igual a } \quad \boldsymbol{r}.$$

7 Según el Problema 6, ¿qué matriz de Householder H de n por n da $H\boldsymbol{a} = \begin{bmatrix} ||\boldsymbol{a}|| \\ \textbf{ceros} \end{bmatrix}$?

8 ¿Qué múltiplo de $\boldsymbol{a} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ debería restarse de $\boldsymbol{b} = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$ para que el resultado $\boldsymbol{A_2}$ sea ortogonal a $\boldsymbol{a}$? Esboce una figura para mostrar $\boldsymbol{a}, \boldsymbol{b}$, y $\boldsymbol{A_2}$.

9 Complete el proceso de Gram-Schmidt en el Problema 8 calculando $\boldsymbol{q}_1 = \boldsymbol{a}/\|\boldsymbol{a}\|$ y $\boldsymbol{A}_2 = \boldsymbol{b} - (\boldsymbol{b}^{\mathrm{T}}\boldsymbol{q}_1)\boldsymbol{q}_1$ y $\boldsymbol{q}_2 = \boldsymbol{A_2}/\|\boldsymbol{A_2}\|$ y factorizando en QR:

$$\begin{bmatrix} 1 & 4 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_1 & \boldsymbol{q}_2 \end{bmatrix} \begin{bmatrix} \|\boldsymbol{a}\| & ? \\ 0 & \|\boldsymbol{A_2}\| \end{bmatrix}.$$

10 Si $A = QR$ entonces $A^{\mathrm{T}}A = R^{\mathrm{T}}R =$ triangular _____ por triangular _____. *Gram-Schmidt en A corresponde a eliminación en $A^{\mathrm{T}}A$.*

11 *Si $Q^{\mathrm{T}}Q = I$ muestre que $Q^{\mathrm{T}} = Q^+$. Si $A = QR$ con R invertible, muestre que $QQ^{\mathrm{T}} = AA^+$. En la última página* 155 *de la Parte* II, *esto será la clave para calcular una SVD.*

Esta página está dedicada a la aplicación más sencilla y más importante de mínimos cuadrados: **ajustar una recta a datos**. Una recta $b = C + Dt$ tiene $n = 2$ parámetros C y D. Tenemos $m > 2$ mediciones b_i en m instantes de tiempo diferentes t_i. Las ecuaciones $A\boldsymbol{x} = b$ (sin solución) y $A^{\mathrm{T}}A\widehat{\boldsymbol{x}} = A^{\mathrm{T}}\boldsymbol{b}$ (con solución) son

$$A\boldsymbol{x} = \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_m \end{bmatrix} \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{D} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \qquad A^{\mathrm{T}}A\widehat{\boldsymbol{x}} = \begin{bmatrix} m & \sum t_i \\ \sum t_i & \sum t_i^2 \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{C}} \\ \widehat{\boldsymbol{D}} \end{bmatrix} = \begin{bmatrix} \sum b_i \\ \sum b_i t_i \end{bmatrix}.$$

El espacio columna $\mathbf{C}(A)$ es un plano bidimensional en $\mathbf{R}^m$. El vector $\boldsymbol{b}$ está en este espacio columna si y solo si los m puntos (t_i, b_i) yacen efectivamente sobre una línea recta. Solo en ese caso, $A\boldsymbol{x} = \boldsymbol{b}$ tiene solución: la recta sobre la que yacen los puntos es $C + Dt$. **El vector $\boldsymbol{b}$ siempre se proyecta al $\boldsymbol{p}$ más cercano que pertenece a $\mathbf{C}(A)$.**

La mejor recta (el ajuste por mínimos cuadrados) pasa por los puntos (t_i, p_i). El vector error $e = A\widehat{x} - b$ tiene componentes $p_i - b_i$. Y e es perpendicular a p.

Existen dos maneras importantes de dibujar este problema de regresión por mínimos cuadrados. Una manera muestra la mejor recta $b = \widehat{C} + \widehat{D}t$ y los errores e_i (distancias verticales a la recta). La segunda manera es verlo en el espacio $\mathbf{R}^m = m$-dimensional. Allí podemos ver el vector de datos $\boldsymbol{b}$, su proyección $\boldsymbol{p}$ en $\mathbf{C}(A)$, y el vector de error $\boldsymbol{e}$. Este es un triángulo rectángulo con $\|\boldsymbol{p}\|^2 + \|\boldsymbol{e}\|^2 = \|\boldsymbol{b}\|^2$.

Para destacar las ideas clave, los problemas 12 a 22 usan datos que corresponden a cuatro puntos con $\boldsymbol{b} = (0, 8, 8, 20)$.

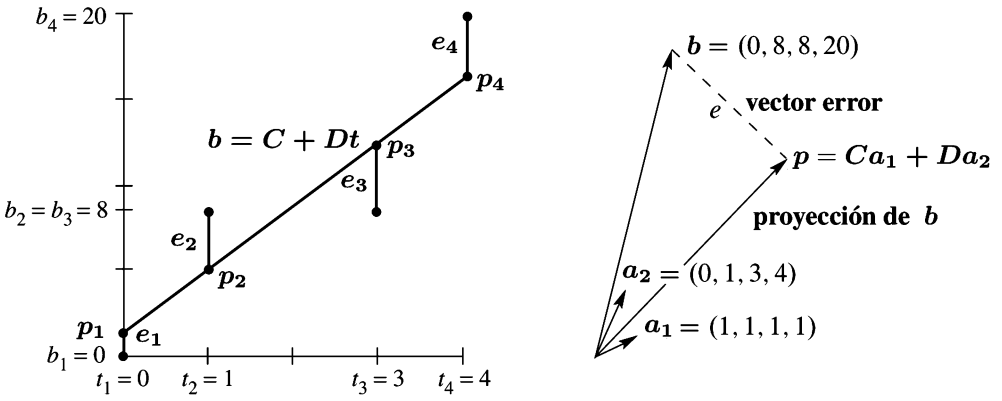


Figura II.3: la recta más cercana $C + Dt$ en el plano $t - b$ coincide con $C\boldsymbol{a}_1 + D\boldsymbol{a}_2$ en $\mathbf{R}^4$.

12 Con $b = 0, 8, 8, 20$ en $t = 0, 1, 3, 4$, plantee y resuelva las ecuaciones normales $A^{\mathrm{T}}A\widehat{\boldsymbol{x}} = A^{\mathrm{T}}\boldsymbol{b}$. Para la mejor recta en la Figura II.3 (izquierda), encuentre sus cuatro alturas p_i y cuatro errores e_i. ¿Cuál es el mínimo error cuadrático $E = e_1^2 + e_2^2 + e_3^2 + e_4^2$?

13 (La recta $C + Dt$ pasa por los p.) Con $b = 0, 8, 8, 20$ en los instantes de tiempo $t = 0, 1, 3, 4$, escriba las cuatro ecuaciones $A\boldsymbol{x} = \boldsymbol{b}$ (sin solución). Cambie las mediciones a $p = 1, 5, 13, 17$ y encuentre una solución exacta para $A\widehat{\boldsymbol{x}} = \boldsymbol{p}$.

14 Verifique que $\boldsymbol{e} = \boldsymbol{b} - \boldsymbol{p} = (-1, 3, -5, 3)$ es perpendicular a ambas columnas de la misma matriz A. ¿ Cuál es la menor distancia $\|\boldsymbol{e}\|$ de $\boldsymbol{b}$ al espacio columna de A?

15 (Usando cálculo) Escriba $E = \|A\boldsymbol{x} - \boldsymbol{b}\|^2$ como una suma de cuatro cuadrados, el último es $(C + 4D - 20)^2$. Encuentre las ecuaciones para las derivadas $\partial E/\partial C = 0$ y $\partial E/\partial D = 0$. Divida por 2 para obtener la ecuación normal $A^{\mathrm{T}}A\widehat{\boldsymbol{x}} = A^{\mathrm{T}}\boldsymbol{b}$.

16 Encuentre la altura C de la mejor *recta horizontal* para ajustar a $\boldsymbol{b} = (0, 8, 8, 20)$. Un ajuste exacto resolvería las ecuaciones sin solución $C = 0$, $C = 8$, $C = 8$, $C = 20$. Encuentre la matriz A de 4 por 1 en estas ecuaciones y resuelva $A^{\mathrm{T}}A\widehat{x} = A^{\mathrm{T}}\boldsymbol{b}$. Dibuje la recta horizontal de altura $\widehat{x} = C$ y los cuatro errores en $\boldsymbol{e}$.

17 Proyecte $\boldsymbol{b} = (0, 8, 8, 20)$ en la recta que pasa por $\boldsymbol{a} = (1, 1, 1, 1)$. Encuentre $\widehat{x} = \boldsymbol{a}^{\mathrm{T}}\boldsymbol{b}/\boldsymbol{a}^{\mathrm{T}}\boldsymbol{a}$ y la proyección $\boldsymbol{p} = \widehat{x}\boldsymbol{a}$. Verifique que $\boldsymbol{e} = \boldsymbol{b} - \boldsymbol{p}$ sea perpendicular a $\boldsymbol{a}$, y encuentre la menor distancia $\|\boldsymbol{e}\|$ de $\boldsymbol{b}$ a la recta que pasa por $\boldsymbol{a}$.

18 Encuentre la recta $b = Dt$ más cercana a los mismos cuatro puntos, *que pasa por el origen*. Un ajuste exacto resovería $D \cdot 0 = 0$, $D \cdot 1 = 8$, $D \cdot 3 = 8$, $D \cdot 4 = 20$. Encuentre la matriz de 4 por 1 y resuelva $A^{\mathrm{T}}A\widehat{x} = A^{\mathrm{T}}\boldsymbol{b}$. Vuelva a dibujar la Figura II.3a mostrando la mejor recta $b = Dt$.

19 Proyecte $\boldsymbol{b} = (0, 8, 8, 20)$ sobre la recta que pasa por $\boldsymbol{a} = (0, 1, 3, 4)$. Encuentre $\widehat{x} = D$ y $\boldsymbol{p} = \widehat{x}\boldsymbol{a}$. La mejor C del Problema 16 y la mejor D del Problema 18 *no coinciden* con el mejor $(\widehat{C}, \widehat{D})$ de los Problemas 11–14. Esto es porque las dos columnas $(1, 1, 1, 1)$ y $(0, 1, 3, 4)$ _____ son perpendiculares.

20 Para la parábola $b = C + Dt + Et^2$ más cercana a los mismos cuatro puntos, escriba las ecuaciones sin solución $A\boldsymbol{x} = \boldsymbol{b}$ con las tres incógnitas $\boldsymbol{x} = (C, D, E)$. Plantee las tres ecuaciones normales $A^{\mathrm{T}}A\widehat{\boldsymbol{x}} = A^{\mathrm{T}}\boldsymbol{b}$ (no hace falta calcular la solución). En la Figura II.3a ahora se está ajustando una parábola a 4 puntos, ¿ Qué está pasando en la Figura II.3b?

21 Para la ecuación cúbica $b = C + Dt + Et^2 + Ft^3$ más cercana a los mismos cuatro puntos, escriba las cuatro ecuaciones $A\boldsymbol{x} = \boldsymbol{b}$. Resuélvalas por eliminación. En la Figura II.3a esta cúbica ahora pasa exactamente por los puntos. ¿ Quiénes son $\boldsymbol{p}$ y $\boldsymbol{e}$?

22 Los promedios de los t_i y de los b_i son $\overline{t} = 2$ y $\overline{b} = 9$. Verifique que $C + D\overline{t} = \overline{b}$. Explique.

(a) Verifique que la mejor recta pasa por el punto central $(\overline{t}, \overline{b}) = (2, 9)$.

(b) Explique por qué $C + D\overline{t} = \overline{b}$ se deduce de la primera ecuación en $A^{\mathrm{T}}A\widehat{\boldsymbol{x}} = A^{\mathrm{T}}\boldsymbol{b}$.

II.3. Tres bases para el espacio columna

Esta sección aborda cuestiones computacionales serias. Las matrices son grandes. Si hay ruido aleatorio, su rango también es grande. Pero el rango efectivo, cuando se quita el ruido, puede ser mucho menor que m y n. El álgebra lineal moderna desarrolló algoritmos rápidos para resolver $A\boldsymbol{x} = \boldsymbol{b}$, $A\boldsymbol{x} = \lambda\boldsymbol{x}$ y $A\boldsymbol{v} = \sigma\boldsymbol{u}$ para matrices grandes.

En general dejaremos para los profesionales los algoritmos especiales. El álgebra lineal numérica se ha desarrollado rápidamente y bien: estamos a salvo con los expertos. Pero podemos reconocer algunas reglas básicas computacionales de sentido común, incluyendo :

1. No usar $A^{\mathrm{T}}A$ y AA^{T} cuando se puede operar directamente con A.

2. No suponer que el orden original de las filas y columnas es el mejor.

La primera advertencia se aplicaría a mínimos cuadrados y a la SVD (y a la estadística computacional). Al formar $A^{\mathrm{T}}A$ estamos elevando al cuadrado el número de condición σ_1/σ_n. Esto mide la sensibilidad y la vulnerabilidad de A. Y para matrices muy grandes, es impensable el costo de calcular y almacenar $A^{\mathrm{T}}A$. Es verdad que la matriz de rigidez de mecánica y la matriz de conductancia de electrónica y la matriz laplaciana de grafos para redes tienen la forma $A^{\mathrm{T}}A$ o $A^{\mathrm{T}}CA$ (con constantes físicas en C). Pero para matrices de datos queremos "algoritmos de raíz cuadrada" que funcionen directamente con la matriz A.

¿ Qué es lo que verdaderamente necesitamos de A ? A menudo la respuesta está en el corazón del álgebra lineal pura y aplicada : **necesitamos una buena base para el espacio columna**. ¡ A partir de ese punto inicial podemos hacer lo que sea ! Para $A\boldsymbol{x} = \boldsymbol{b}$, podemos encontrar una combinación de columnas básicas que se acerque a $\boldsymbol{b}$. Para $A\boldsymbol{v} = \sigma\boldsymbol{u}$, podemos calcular vectores singulares precisos.

Repito, lo que no podemos hacer es aceptar a las primeras r columnas independientes de A automáticamente como una buena base para los cálculos en el espacio columna $\mathbf{C}(A)$.

Tres bases buenas

Revelaremos de inmediato las tres bases que proponemos estudiar. Es posible que sienta la tentación de calificarlas como oro, plata y bronce.

En parte, esos tres premios representan la calidad garantizada de las tres bases. Pero también sugieren que la SVD tiene el mayor costo. Las tres son ganadoras en cuanto a la construcción de una base para el espacio columna.

1. **Vectores singulares** $\boldsymbol{u_1}, \ldots, \boldsymbol{u_r}$ de la SVD, con $\sigma_1 \geq \sigma_2 \geq \ldots \geq \boldsymbol{\sigma_r}$.
2. **Vectores ortonormales** $\boldsymbol{q_1}, \ldots, \boldsymbol{q_r}$ de Gram-Schmidt. ¡ Use pivoteo por columnas !
3. **Columnas independientes** $\boldsymbol{c_1}, \ldots, \boldsymbol{c_r}$ tomadas directamente de A luego de intercambiar columnas.

Cada elección de base de $\mathbf{C}(A)$ da la primera matriz en una "factorización que revela el rango"

$A =$ matriz columna por matriz fila : $(\boldsymbol{m}$ **por** $\boldsymbol{n})$ **igual a** $(\boldsymbol{m}$ **por** $\boldsymbol{r})(\boldsymbol{r}$ **por** $\boldsymbol{n})$.

1. Los factores de A en la SVD reducida son $\boldsymbol{U_r}$ por $\boldsymbol{\Sigma_r V_r^{\mathrm{T}}}$
2. Los factores de A en Gram-Schmidt son $\boldsymbol{Q_{m\times r}}$ por $\boldsymbol{R_{r\times n}}$
3. Los factores de A en eliminación con pivoteo son $\boldsymbol{C_{m\times r}}$ por $\boldsymbol{Z_{r\times n}}$

Comentaremos ahora mismo las propiedades de estas tres factorizaciones importantes

1. La base del espacio columna $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r$ en U es *ortonormal*. Tiene la propiedad adicional de que las filas $\sigma_k \boldsymbol{v}_k^{\mathrm{T}}$ del segundo factor $\Sigma_r V_r^{\mathrm{T}}$ también son ortogonales.
2. La base del espacio columna $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_r$ en Q es *ortonormal*. Las filas del segundo factor R no son ortogonales pero están bien condicionadas (independientes sin dudas).
3. La base del espacio columna $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_r$ en C *no es ortogonal*. Pero ambas C y el segundo factor Z pueden estar bien condicionadas. Esto se logra permitiendo intercambios de columna (no solo permitiendo sino insistiendo). C contiene r "columnas buenas" de A.

La tercera factorización $A = CZ$ puede llamarse descomposición interpolante (**ID**, por su sigla en inglés, en lugar de **SVD** o **QR**). Como este es un nuevo concepto para esta sección, me enfocaré en sus propiedades. Las notas y los artículos de Gunnar Martinsson y sus distinguidos coautores son la base para la explicación de la ID que presentamos a continuación (vea Martinsson and Tropp en Acta Numerica (2020) y arXiv:2002.01387).

Las columnas de C provienen directamente de A, pero las filas de Z no. Eso sería pedir demasiado. Luego tendremos $\boldsymbol{CMR}$ con columnas de A en C y filas de A en R, y una **matriz de mezcla** $\boldsymbol{M}$ para acercar el producto a A.

Descomposición interpolante = factorización columna / fila

He aquí cuatro ventajas importantes de CZ comparado con QR y $U\Sigma V^{\mathrm{T}}$. Recuerde que las columnas de C son **en verdad columnas de A** cuidadosamente elegidas. Eso le da serias ventajas a $A = CZ$.

- $A = CZ$ conlleva **menos tiempo de cálculo** y **menos espacio de almacenamiento** que $A = U\Sigma V^{\mathrm{T}}$ y $A = QR$.
- Cuando A es **rala, no negativa o ambas**, también lo es C. C proviene directamente de A.
- Cuando A se obtiene a partir de discretizar una ecuación diferencial o integral, las columnas de C **tienen más significado** que las bases ortonormales en U y Q.
- Cuando A es una matriz de datos, las columnas que se mantienen en C tienen **interpretaciones sencillas**.

Esos últimos tres puntos pueden hacer una gran diferencia en la interpretaión de los resultados una vez que los cálculos se han completado. Esta es una crítica de la SVD : los vectores singulares son algebraica y geométricamente perfectos pero son "humanamente" difíciles de entender.

Cuando los números en A son intrínsicamente positivos, el objetivo puede ser una factorización matricial no negativa (NMF). *Así ambos factores en* $A \approx MN$ *tienen elementos* ≥ 0. Esto es importante para tamaños moderados pero es exigir demasiado en el mundo del *big data*.

$A = CZ$ es apropiado para matrices grandes cuando Z tiene elementos pequeños (tales como $|z_{ij}| \leq 2$).

Factorizando A como CZ

Cuando C contiene una base para el espacio columna de A, el factor Z está completamente determinado. Cada columna de A es una combinación única de esas columnas básicas en C:

$$(\text{columna } j \text{ de } A) = (\text{matriz } C)\ (\text{vector columna } \boldsymbol{z}_j). \tag{1}$$

Esta es exactamente la afirmación $A = CZ$, columna por columna. Esta idea estaba en la Sección I.1 (con R en lugar de Z). Daba una linda prueba de que rango fila = rango columna. Pero es posible que las columnas de esa C no sean muy independientes. Esta sección pronto se enfocará seriamente sobre cómo elegir C rápido y bien (pero todavía no).

Primero podemos dar un paso más allá de $A = CZ$, mirando el espacio fila de C. En alguna parte C tiene r filas independientes. Si colocamos esas filas independientes $\boldsymbol{b}_1^{\mathrm{T}}, \ldots, \boldsymbol{b}_r^{\mathrm{T}}$ en una matriz B de r por r, entonces B es invertible. Cada fila de C es una combinación única de esas filas básicas en B:

$$(\text{fila } i \text{ de } C) = (\text{vector fila } \boldsymbol{y}_i^{\mathrm{T}})\ (\text{matriz invertible } B). \tag{2}$$

Esta es exactamente la afirmación $C = YB$, fila por fila. Combinémosla con $A = CZ$:

$$\boxed{\boldsymbol{A}_{m\times n} = \boldsymbol{C}_{m\times r}\ \boldsymbol{Z}_{r\times n} = \boldsymbol{Y}_{m\times r}\ \boldsymbol{B}_{r\times r}\ \boldsymbol{Z}_{r\times n}} \tag{3}$$

Todas esas matrices tienen rango r.

Ahora usemos el hecho de que las columnas de C se obtuvieron directamente de A y las filas de B provienen directamente de C. Para esas columnas y filas, los vectores $\boldsymbol{z}_j$ en (1) y los $\boldsymbol{y}_i^{\mathrm{T}}$ en (2) provienen de la matriz identidad I_r. Suponiendo que $\boldsymbol{z}_j$ e $\boldsymbol{y}_i^{\mathrm{T}}$ son las **primeras $\boldsymbol{r}$ columnas y filas en $\boldsymbol{A}$ y $\boldsymbol{C}$**, entonces $\boldsymbol{A = YBZ}$ tiene una forma especial:

$$\boxed{\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{I_r} \\ \boldsymbol{C_{m-r}\,B^{-1}} \end{bmatrix} \qquad \boldsymbol{B} = \text{submatriz de } \boldsymbol{A} \qquad \boldsymbol{Z} = \begin{bmatrix} \boldsymbol{I_r} & \boldsymbol{B^{-1}Z_{n-r}} \end{bmatrix}} \tag{4}$$

Solo estamos suponiendo que B, la esquina superior izquierda de r por r de A, es invertible. Toda matriz A de rango r tiene una submatriz B invertible de r por r en algún lado. (Tal vez hay muchas de esas B.) Cuando B está en esa esquina superior izquierda, por eliminación se obtienen Y y Z.

Ejemplo 1 Esta matriz de 3 por 4 de rango 2 comienza con una matriz B de 2 por 2 invertible:

$$A = \begin{bmatrix} \mathbf{1} & \mathbf{2} & 4 & 2 \\ \mathbf{0} & \mathbf{1} & 2 & 1 \\ 1 & 3 & 6 & 3 \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{2} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} & 0 & 0 \\ \mathbf{0} & \mathbf{1} & 2 & 1 \end{bmatrix} = YBZ \tag{5}$$

$$(3\times 2)(2\times 2)(2\times 4)$$

He aquí un punto importante que es necesario entender. Si C_{m-r} o Z_{n-r} o ambas contienen números grandes, los factores en (5) son una mala elección. Somos felices si esos otros elementos de Y y Z son pequeños. Lo mejor sería $|y_{ij}| \leq 1$ y $|z_{ij}| \leq 1$. ¡ Y podemos encontrar una **submatriz B que haga que esto sea cierto** !

Elegir B como la submatriz de r por r de A con el mayor determinante.

Entonces todos los elementos de Y y Z satisfacen $|\,y_{ij}\,| \leq 1$ y $|\,z_{ij}\,| \leq 1$.

Prueba. Esos y_{ij} y z_{ij} son los números en $C_{m-r}B^{-1}$ y en $B^{-1}Z_{n-r}$. Comenzamos con el z_{ij}. Dado que $B(B^{-1}Z_{n-r}) = Z_{n-r}$, sabemos que cada columna $\boldsymbol{z}_j$ de $B^{-1}Z_{n-r}$ resuelve este sistema de ecuaciones lineales :

$$\boldsymbol{Bz_j} = \textbf{columna } \boldsymbol{j} \textbf{ de } \boldsymbol{Z_{n-r}}.$$

Según la regla de Cramer, los números z_{ij} en la solución $\boldsymbol{z}_j$ son *cocientes de determinantes* :

$$z_{ij} = \frac{\det(B \text{ con su } i\text{-ésima columna reemplazada por esa columna } j)}{\text{determinante de } B}$$

Recuerde que esa B es la submatriz de r de r de A con el mayor determinante. La matriz del numerador es una de las muchas submatrices que *no* elegimos. Su determinante es menor que $\det B$. Así $|z_{ij}| \leq 1$.

De manera similar, las filas de Y provienen de $C_{m-r}B^{-1}$. Como $(C_{m-r}B^{-1})B = C_{m-r}$, esas filas resuelven las ecuaciones lineales $\boldsymbol{y}_i^{\mathrm{T}}B =$ fila i de C_{m-r}. Cuando trasponemos y usamos la regla de Cramer, las componentes y_{ij} de $\boldsymbol{y}_i^{\mathrm{T}}$ son nuevamente cocientes de determinantes. Y $\det B$ es de nuevo el denominador. Ya que $\det B$ es el más grande, tenemos $|y_{ij}| \leq 1$.

Hay que admitir un gran problema. Dijimos que "podemos encontrar B tal que $|y_{ij}| \leq 1$ y $|z_{ij}| \leq 1$." **Esto no es cierto**. En realidad, no podemos encontrar la submatriz B con máximo determinante. No sin una computadora cuántica (que todavía no existen). B está en alguna parte dentro de A, pero no tenemos idea de cuál submatriz es.

Lo asombroso es que con la aleatorización podemos identificar una buena submatriz B. Entonces hay una **muy alta probabiliad** (no una certeza) de que todo $|y_{ij}| \leq 2$ y $|z_{ij}| \leq 2$. Completaremos esta presentación sobre el tema descomposición interpolante (cuando las columnas o filas provienen directamente de A) en la siguiente sección. La selección de columnas o filas será **aleatoria con probabilidades cuidadosamente elegidas**.

Ejemplo 2 La matriz en el Ejemplo 1 tiene esta $B_{\text{máx}}$ con determinante máximo $= 2$. Así $|y_{ij}| \leq 1$ y $|z_{ij}| \leq 1$:

$$A = \begin{bmatrix} \mathbf{1} & 2 & \mathbf{4} & 2 \\ \mathbf{0} & 1 & \mathbf{2} & 1 \\ 1 & 3 & 6 & 3 \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{4} \\ \mathbf{0} & \mathbf{2} \end{bmatrix} \begin{bmatrix} \mathbf{1} & 0 & \mathbf{0} & 0 \\ \mathbf{0} & -.5 & \mathbf{1} & -.5 \end{bmatrix} = YB_{\text{máx}}Z \qquad (6)$$

Factorización CMR: elección de C y R

A continuación describiremos una elección recomendada de columnas y filas de A para usar en C y R. Una **matriz de mezcla** M siempre es necesaria para lograr $A \approx CMR$. Esto puede ser una igualdad $A = CMR$ si A tiene realmente rango pequeño. Para matrices grandes que son *aproximadamente de rango pequeño* (esto es lo que suponemos) podemos comenzar con vectores singulares aproximados en $A \approx U\Sigma V^{\mathrm{T}}$. Usaremos U y V para lograr una factorización CMR de rango r cercana a A.

Nuestra guía es *A DEIM induced CUR factorization* de D. C. Sorensen y M. Embree, arXiv: 1407.5516v2, 18 Sep 2015; SIAM J. Scientific Computing **38** (2016) 1454-1482.

El método de interpolación empírica discreta elige C y R. Se sustituye U por M.

Parece un poco extraño que CMR (que usa columnas y filas tomadas de A) comience con $U\Sigma V^{\mathrm{T}}$ (que encuentra combinaciones ortonormales de esas columnas y filas). Y el cálculo aproximado de $U\Sigma V^{\mathrm{T}}$ suele comenzar con QR, para obtener una matriz Q ortogonal de manera rápida. Así las bases C, U, Q para el espacio columna de A (¡ *aproximado* !) están todas interconectadas por algoritmos rápidos.

El error de la aproximación final está controlada por el siguiente valor singular σ_{r+1} en $A = U\Sigma V^{\mathrm{T}}$, y de las r columnas de U y V correspondientes a nuestras elecciones para C y R:

$$(\boldsymbol{m \times r})\,(\boldsymbol{r \times r})\,(\boldsymbol{r \times m}) \qquad \|\boldsymbol{A - CMR}\| \le (\|\boldsymbol{U_r^{-1}}\| + \|\boldsymbol{V_r^{-1}}\|)\,\boldsymbol{\sigma_{r+1}}. \tag{7}$$

Elección de columnas de A para conformar C

Comencemos con las r columnas de $U_{m\times r}$ (los vectores singulares a izquierda aproximados de A). Supongamos que s columnas de $E_{m\times r}$ provienen directamente de $I_{m\times r}$. Si $E^{\mathrm{T}}U$ es una matriz invertible, entonces

1 $P = U(E^{\mathrm{T}}U)^{-1}E^{\mathrm{T}}$ tiene $P^2 = P =$ matriz de proyección

2 $P\boldsymbol{x}$ es igual a $\boldsymbol{x}$ en esas s posiciones elegidas, así P es una **proyección interpolante**

La propiedad crucial es **2** (vea el trabajo de Sorensen y Embree para todas las pruebas). Para $s = 1$, el algoritmo DEIM elige el mayor elemento en valor absoluto del primer vector singular $\boldsymbol{u}_1$. Eso nos lleva a P_1. La siguiente elección (que lleva a P_2) se decide por el mayor elemento en $\boldsymbol{u}_2 - P_1\boldsymbol{u}_2$. Toda P_j futura se decide por el mayor elemento en $\boldsymbol{u}_j - P_{j-1}\boldsymbol{u}_j$. Esto corresponde a maximizar cada pivote en la eliminación ordinaria. Un pseudocódigo sencillo aparece en el trabajo de Sorensen y Embree.

Las filas de A para conformar R se seleccionan de la misma manera. Este procesamiento secuencial tiene una gran ventaja, en comparación con el muestreo basado en el cuadrado de las normas de todas las filas de U y V. El próximo paso es estimar el error en la proyección interpolante:

$$\|A - C(C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}}A\| \le \eta_C\sigma_{r+1} \qquad \text{y} \qquad \|A - AR^{\mathrm{T}}(RR^{\mathrm{T}})^{-1}R\| \le \eta_R\sigma_{r+1} \tag{8}$$

En la práctica, esas constantes η_C y η_R no son muy grandes (con orden de magnitud menor a 100). De nuevo esto es similar al pivoteo parcial: en teoría es posible un rápido crecimiento pero nunca se ha visto en la práctica.

La matriz mezcla M

La última decisión es la matriz mezcla en $A \approx CMR$. Sin esa matriz M, el producto CR normalmente no está cerca de A. La opción natural para M es

$$\boxed{M = (C^{\mathrm{T}}C)^{-1}C^{\mathrm{T}}AR^{\mathrm{T}}(RR^{\mathrm{T}})^{-1} = [\text{inversa a izquierda de } C]\, A\, [\text{inversa a derecha de } R].} \quad (9)$$

En la última página de la Sección I.1, esta es la opción que logró igualdad en $A = CMR$. En esa página el rango de A era exactamente r (en ese entonces éramos inocentes). Ahora r es solo el rango aproximado de una matriz A grande.

Para los algoritmos aleatorizados de la siguiente sección II.4 (donde A es demasiado grande para el algoritmo DEIM), veremos que Halko-Martinsson-Tropp y Mahoney-Drineas también hicieron esta elección. El análisis inicial de Stewart en (*Numerische Math.* **83** (1999) 313-323) señaló el camino para la estimación del error en (8).

Comenzando por $A = QR$ con pivoteo por columnas

La factorización QR es un punto de inicio favorito en muchos algoritmos de álgebra lineal numérica para matrices grandes. Con un costo razonable, produce una base ortonormal en Q para el espacio columna de A. A partir de Q podemos ir con rapidez y exactitud a las otras dos bases fundamentales :

Columnas de C (vienen directamente de A : descomposición interpolante $\boldsymbol{A = CMR}$)

Columnas de U (vectores ortonormales : descomposición en valores singulares $\boldsymbol{A = U\Sigma V^{\mathrm{T}}}$)

Es necesario comprender bien que las buenas elecciones de C y U dependerán de una buena elección de Q. Ya que esa matriz es ortogonal (por lo tanto perfectamente condicionada), es el otro factor R el que determina la calidad de $A = QR$.

El proceso ordinario de Gram-Schmidt en la Sección II.2 mantuvo las columnas de A en su orden original. El algoritmo QR *pivoteado* elige, al inicio de cada paso nuevo, la mayor de las columnas restantes (*pivoteo por columnas*). Esto produce una permutación Π de tal manera que las primeras k columnas de $A\Pi$ (y Q) son las columnas importantes :

$$\boldsymbol{A\Pi} = QR = \boldsymbol{Q}_{m\times m}\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{C} \end{bmatrix} \text{ con } A \text{ triangular de } k \times k \quad (10)$$

Una "fuerte factorización reveladora de rango" tiene esta forma con condiciones adicionales en los bloques : $\sigma_i(A)$ *no es pequeña*, $\sigma_j(C)$ *no es grande*, $A^{-1}B$ *no es grande*. Esas propiedades son valiosas a la hora de encontrar (computacionalmente) una base para el espacio nulo de A. Supongamos que estas propiedades se cumplen y avancemos, utilicemos las primeras k columnas de Q en las factorizaciones CMR y $U\Sigma V^{\mathrm{T}}$ de A.

Aproximación de rango pequeño con una QR parcial

Las páginas anteriores supusieron que el rango r de la matriz A era relativamente pequeño. A menudo, esta suposición es falsa pero efectivamente verdadera. Queremos decir que

$$A = (\textbf{matriz } A_r \textbf{ de rango pequeño } r) + (\textbf{matriz } E \textbf{ de norma pequeña}).$$

Queremos encontrar la matriz de rango pequeño A_r y posteriormente utilizarla para realizar cálculos. Podemos estimar el error se que comete al aproximar A con A_r, ignorando la matriz E (matriz grande, norma pequeña).

Martinsson afirma que la aproximación A_r de rango pequeño se puede calcular con QR. Los primeros r pasos de ese algoritmo (con pivoteo por columnas) producen $Q_r R_r$:

$$A = Q_r R_r + E = (r \text{ columnas } q_j)(r \text{ filas } r_j^{\mathrm{T}}) + (n - r \text{ columnas ortogonales a esos } q_j).$$

¡ $||E||$ es pequeña ! Como este algoritmo incluye intercambios de columnas, esas columnas en E pueden no ser las *últimas* $n-r$ columnas de A. Pero una permutación de columnas P de n por n moverá esas columnas al final de AP. Las r columnas importantes son $Q_r R_r$ al frente:

$$\boxed{AP = \begin{bmatrix} Q_r R_r & A_{n-r} \end{bmatrix} \quad \textbf{y} \quad A = \begin{bmatrix} Q_r R_r P^{\mathrm{T}} & A_{n-r} P^{\mathrm{T}} \end{bmatrix}.} \tag{11}$$

$Q_r R_r P^{\mathrm{T}}$ es la aproximación buena de rango r de A. Por suerte el algoritmo QR con pivoteo calcula las normas de las columnas, entonces sabemos cuando $||A_{n-r}||_F$ está por debajo de nuestro límite establecido ϵ. Allí nos detenemos.

Este algoritmo es efectivo, pero renuncia al objetivo de elegir columnas directamente de A. Logró su objetivo de columnas ortonormales (en Q).

Volvemos a la aproximación de rango pequeño en II.4 y en la Parte III de este libro.

Una SVD aproximada a partir de la QR parcial

Para completar el círculo ahora buscamos una buena aproximación a la SVD. Esta proviene de la aproximación cercana a $A = QR$ en la ecuación (11). La matriz error E satisface $||E||_F < \epsilon$.

$$\textbf{Error pequeño } E \qquad \underset{m \times n}{(A)} = \underset{m \times r}{(Q_r)} \; \underset{r \times n}{(R_r\, P^{\mathrm{T}})} + \underset{m \times n}{(E)} \tag{12}$$

Dos pasos rápidos son suficientes para producir una SVD muy cercana a A, con el mismo error E :

Primero, encontrar la SVD de la matriz $R_r P^{\mathrm{T}}$ con solo r filas : $\quad R_r P^{\mathrm{T}} = U_r \Sigma V^{\mathrm{T}}$

Segundo, multiplicar Q_r por U_r para encontrar $U = Q_r U_r =$ ortogonal por ortogonal :

$$\textbf{SVD aproximada con error } E \qquad \boxed{A = Q_r U_r \Sigma V^{\mathrm{T}} + E = U \Sigma V^{\mathrm{T}} + E} \tag{13}$$

Así una SVD pequeña y una QR grande dan una SVD grande (aproximada).

Problemas de la Sección II.3

1 El típico indicador de peligro por error de redondeo es el *número de condición* $||A||\,||A^{-1}||$. Muestre por qué ese número se eleva al cuadrado si se trabaja con $A^{\mathrm{T}}A$ en lugar de A.

2 Escriba una matriz A de 2 por 2 con número de condición > 1000. Calcule A^{-1} ¿Por qué A^{-1} tiene también número de condición > 1000?

3 La razón para que ambas $||A||$ y $||A^{-1}||$ aparezcan es que trabajamos con el *error relativo*. Si $A\boldsymbol{x} = \boldsymbol{b}$ y $A(\boldsymbol{x} + \Delta\boldsymbol{x}) = \boldsymbol{b} + \Delta\boldsymbol{b}$, entonces $A\,\Delta\boldsymbol{x} = \Delta\boldsymbol{b}$. Muestre que

$$\frac{||\Delta\boldsymbol{x}||}{||\boldsymbol{x}||} \leq ||A||\,||A^{-1}||\,\frac{||\Delta\boldsymbol{b}||}{||\boldsymbol{b}||}.$$

4 ¿Por qué $\lambda_{\text{máx}}/\lambda_{\text{mín}}$ es igual al número de condición cuando A es definida positiva?

5 *Importante* ¿Cuál es el número de condición de una matriz ortogonal Q?

6 Suponga que las columnas de C contienen una base ortonormal para el espacio columna de A y las filas de R contienen una base ortonormal para el espacio fila. ¿Será cierto que esas bases contienen a los vectores singulares $\boldsymbol{v}$ y $\boldsymbol{u}$ de la SVD?

7 Si C y R contienen bases para el espacio columna y espacio fila de A, ¿Por qué se cumple que $A = CMR$ para alguna matriz cuadrada e invertible M?

8 He aquí una matriz con *rango numérico* 2. El número $\epsilon =$ epsilon de máquina es 2^{-16}. ¿Qué vectores ortonormales $\boldsymbol{q}_1$ y $\boldsymbol{q}_2$ darán una buena base para el espacio columna: un espacio que será elegido probablemente utilizando QR con pivoteo?

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1+\epsilon & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

9 Aproxime esa matriz A por $QRP^{\mathrm{T}} +$ (orden ϵ) con P una matriz de permutación.

10 ¿Cuál submatriz $B_{\text{máx}}$ de 2 por 2 de A (rango 2) tiene el mayor determinante?

$$A = \begin{bmatrix} 2 & 5 & 1 \\ 1 & 3 & 5 \\ 3 & 1 & 1 \end{bmatrix}$$

Factorice $A = YB_{\text{máx}}Z$ como en la ecuación (6).

II.4. Álgebra lineal aleatorizada

Esta sección sobre aleatorización está incompleta. La primera razón es que no fue escrita por un experto. Este libro no es una guía detallada sobre cómo calcular la SVD o la QR. Aún así, es posible, y vale la pena, hacer aquí una breve exposición sobre ideas claves y algoritmos que hacen que esos cálculos sean posibles para matrices grandes.

Entre esas ideas aparecen enfoques nuevos e importantes que comienzan con vectores aleatorios $\boldsymbol{x}$. **Así los productos $A\boldsymbol{x}$ son muestras aleatorias del espacio columna de $\boldsymbol{A}$.** Con r de esos vectores (o $r + 10$ para estar seguros, protegiéndonos contra accidentes aleatorios) tenemos una matriz potencialmente delgada a partir de la cual realizar cálculos. La ganancia en velocidad es impresionante. Este es también el punto de partida en la Sección III.5 para "*compressed sensing*" que acelera la adquisición y procesamiento de señales digitales.

Esta sección presenta y describe los pasos básicos de los cálculos aleatorizados. Esa idea ha causado una revolución en el álgebra lineal numérica para grandes matrices.

El primer ejemplo es **multiplicación matricial**. Si A y B son de m por n y de n por p, entonces $C = AB$ normalmente necesita mnp multiplicaciones individuales: n multiplicaciones para cada uno de los mp productos internos en AB, o mp multiplicaciones para cada uno de los n productos externos (columnas por filas). *Multiplicar matrices muy grandes es costoso.*

Suponga que **solo hacemos un muestreo de $\boldsymbol{A}$ y $\boldsymbol{B}$** en lugar de usar las matrices completas. Unos pocos elementos a_{ij} y b_{jk} no nos dicen mucho. Pero s columnas $\boldsymbol{a}_k$ de A y s filas correspondientes $\boldsymbol{b}_k^{\mathrm{T}}$ de B nos darán s matrices de rango uno $\boldsymbol{a}_k\boldsymbol{b}_k^{\mathrm{T}}$. *Si esos son productos externos "típicos", podemos multiplicar su suma por n/s*, para estimar la verdadera AB = suma de n productos. Observe que usamos productos columna-fila (altamente recomendado) y no productos internos fila-columna (bajo nivel).

Esto no es todo. El muestreo aleatorio usa estadística básica. Los productos grandes $\boldsymbol{a}_k\boldsymbol{b}_k^{\mathrm{T}}$ obviamente contribuyen más a $C = AB$. Podemos incrementar las chances de obtener esas muestras más grandes (y lo haremos) si les damos probabilidades mayores a esos productos cambiando la probabilidad uniforme a "**muestreo por norma al cuadrado**". Ya que nuestro objetivo es obtener el valor esperado correcto y la menor varianza, tenemos que compensar en nuestras fórmulas. ¡ Usted verá el valor de las ideas estadísticas !

Nuestra presentación sigue principalmente las notas de clase de Michael Mahoney de su curso en UC Berkeley. Están bien organizadas y bien redactadas, son una contribución importante. Las notas del curso de 2013 fueron publicadas en 2016 y comienzan con este resumen de la multiplicación matricial aleatoria:

Una **matriz de muestreo $\boldsymbol{S}$** actuará sobre las columnas de A y las filas de B para producir C y R:

$$\boldsymbol{C = AS} \text{ y } \boldsymbol{R = S^{\mathrm{T}}B} \text{ y } \boldsymbol{CR = ASS^{\mathrm{T}}B \approx AB}. \tag{1}$$

Multiplicamos C y R en lugar de las matrices correctas y completas A y B. No será cierto que SS^{T} está cerca de I. Pero sí será cierto que *el valor esperado de SS^{T} es I*. Esta es la clave de la aleatorización.

1 **Muestreo columna-fila** Cada columna de S tiene un elemento no nulo (llamémoslo s_k). Así AS elige columnas individuales $\boldsymbol{a}_k$ directamente de A, y $S^{\mathrm{T}}B$ elige las filas correspondientes $\boldsymbol{b}_k^{\mathrm{T}}$ de B. Multiplicar AS por $S^{\mathrm{T}}B$ da una suma de **productos columna-fila $\boldsymbol{a_k b_k^{\mathrm{T}}}$ pesados por los números $\boldsymbol{s_k^2}$.**

Un producto $s_k^2 \boldsymbol{a}_k \boldsymbol{b}_k^{\mathrm{T}}$ proviene de una columna de AS y una fila de $S^{\mathrm{T}}B$:

$$\begin{bmatrix} & & \\ \cdot & \boldsymbol{a}_k & \cdot \\ & & \end{bmatrix} \begin{bmatrix} 0 \\ s_k \\ 0 \end{bmatrix} = s_k \boldsymbol{a}_k \qquad \begin{bmatrix} 0 & s_k & 0 \end{bmatrix} \begin{bmatrix} & \cdot & \\ & \boldsymbol{b}_k^{\mathrm{T}} & \\ & \cdot & \end{bmatrix} = s_k \boldsymbol{b}_k^{\mathrm{T}} \tag{2}$$

De esta manera AB es aproximada por la suma pesada $CR = \sum s_k^2 \boldsymbol{a}_k \boldsymbol{b}_k^{\mathrm{T}}$ de una *selección aleatoria* de s matrices de rango uno. La selección es al azar, pero elegimos los pesos.

2 **Muestreo via proyecciones aleatorias $\boldsymbol{S}$** La matriz S sigue siendo delgada, así AS tiene muchas menos columnas que A. Las columnas de S ahora contienen múltiples valores no nulos, así AS combina columnas al proyectar $\mathbf{C}(A)$ a dimensiones inferiores. Por lo tanto $ASS^{\mathrm{T}}B$ es una aproximación más uniforme a AB.

Comenzamos con muestreo aleatorio. Luego describimos proyecciones aleatorias. Esas pueden producir **precondicionadores** rápidos y útiles para la matriz original AB, y pueden reducir el costo computacional del agrupamiento de grafos (graph clustering).

Práctica calculando media y varianza

He aquí un problema de muestreo muy simplificado. En lugar de una matriz, comenzamos con un vector $\boldsymbol{v} = (a, b)$. Muestreamos dos veces (2 muestras independientes). Luego calcularemos la media m y la varianza σ^2. La Sección V.1 describirá muchos más ejemplos de m y σ^2.

Primera muestra : con probabilidad $\frac{1}{2}$ y $\frac{1}{2}$, elegir $(a, 0)$ o $(0, b)$
Segunda muestra : repetir exactamente. Luego sumar las dos muestras para obtener (x_1, x_2)

Calculando la media $\boldsymbol{m} = \mathrm{E}[(x_1, x_2)]$ = valor esperado = resultado promedio (x_1, x_2)
Primer método : el valor promedio de la muestra 1 es $\frac{1}{2}(a, 0) + \frac{1}{2}(0, b) = \frac{1}{2}(a, b)$
Tenemos dos ensayos idénticos independientes (dos muestras). Sumamos sus medias :

$$\textbf{Media global } \boldsymbol{m} = \mathrm{E}[(\boldsymbol{x}_1, \boldsymbol{x}_2)] = \frac{1}{2}(a, b) + \frac{1}{2}(a, b) = (\boldsymbol{a}, \boldsymbol{b})$$

Nuestro experimento con dos muestras fue *sin sesgos*. Se logró la media deseada.
Segundo método : el experimento dio los siguientes 4 resultados con cada probabilidad $\frac{1}{4}$

$$(a, 0) + (a, 0) = (\boldsymbol{2a}, \boldsymbol{0}) \quad (a, 0) + (0, b) = (\boldsymbol{a}, \boldsymbol{b}) = (0, b) + (a, 0) \quad (0, b) + (0, b) = (\boldsymbol{0}, \boldsymbol{2b})$$

La media es la suma de los cuatro resultados pesados con sus probabilidades $\left(\textbf{todos } \frac{\mathbf{1}}{\mathbf{4}}\right)$:

$$\textbf{Media global } \boldsymbol{m} = \frac{1}{4}(2a, 0) + \frac{1}{4}(a, b) + \frac{1}{4}(a, b) + \frac{1}{4}(0, 2b) = (\boldsymbol{a}, \boldsymbol{b}) \text{ como antes}$$

La varianza σ^2 = promedio pesado del cuadrado de las distancias entre los resultados y la media. Usaremos dos métodos equivalentes para calcular $\sigma^2 = \mathrm{E}\,[(x-\text{media})]^2 = \mathrm{E}\,[x^2] - (\text{media})^2$.
Primer método para encontrar la varianza: sumar todos los **(resultados − media)^2** pesados por sus probabilidades $\frac{1}{4}$

$$\frac{1}{4}\Big[(2a,0)-(a,b)\Big]^2+\frac{1}{4}\Big[(a,b)-(a,b)\Big]^2+\frac{1}{4}\Big[(a,b)-(a,b)\Big]^2+\frac{1}{4}\Big[(0,2b)-(a,b)\Big]^2=$$

$$\frac{1}{4}(a^2,b^2)+\frac{1}{4}(0,0)+\frac{1}{4}(0,0)+\frac{1}{4}(a^2,b^2)=\boldsymbol{\frac{1}{2}(a^2,b^2)}$$

Segundo método: sumar todos los **(resultados)^2** pesados por sus probabilidades y restar la **(media)^2**

$$\boldsymbol{\sigma^2} = \frac{1}{4}(2a,0)^2+\frac{1}{4}(a,b)^2+\frac{1}{4}(a,b)^2+\frac{1}{4}(0,2b)^2-(a,b)^2$$

$$=\left(a^2+\frac{a^2}{4}+\frac{a^2}{4}+0-a^2, 0+\frac{b^2}{4}+\frac{b^2}{4}+b^2-b^2\right)=\boldsymbol{\frac{1}{2}\left(a^2,b^2\right)} \tag{3}$$

Observación: si b es más grande que a, podemos mantener la media correcta (a,b) y reducir la varianza σ^2 dándole *mayor probabilidad* a las muestras más grandes $(0,b)$. El muestro matricial hace esto (vea el Problema 7 al final de esta Sección II.4).

Esta página usó $s = 2$ muestras para $n = 2$ números a, b. *No fue útil, no se ahorró tiempo.* Las siguientes páginas usan $s << n$ muestras para una matriz con n columnas. La media de AB se mantiene correcta.

Multiplicación matricial aleatoria con media correcta AB

La matriz de muestreo S de n por s contiene s columnas. Cada columna de S tiene *un valor no nulo*. Para la columna j de S, la posición de ese valor no nulo es aleatoria. Si la opción aleatoria es la fila $k = k(j)$, el valor no nulo en la fila k, columna j de S es s_{kj}. La matriz muestreada es AS:

Las columnas 1 a s de AS son los números s_{kj} por las columnas $k(1)$ a $k(s)$ de A.

He aquí un ejemplo con $s = 2$ muestras. Toma muestras de las columnas $k(1) = 1$ y $k(2) = 3$ de A:

$$AS = \begin{bmatrix} & & \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 & \boldsymbol{a}_3 \\ & & \end{bmatrix}\begin{bmatrix} s_{11} & 0 \\ 0 & 0 \\ 0 & s_{32} \end{bmatrix} = \begin{bmatrix} & \\ s_{11}\boldsymbol{a}_1 & s_{32}\boldsymbol{a}_3 \\ & \end{bmatrix}$$

La pregunta clave es: **¿ Cómo elegimos esos números s_{kj} ?** La respuesta es: **¡ Usando probabilidades !** Nuestra intención es usar muestreo aleatorio. Debemos elegir esas s columnas de A al azar (dándole a cada columna una chance de ser elegida) en la multiplicación aleatoria

Asignar **probabilidad p_j** a cada una de las n columnas de A, con $p_1 + \cdots + p_n = 1$

Elegir **s columnas con reemplazo** (así las columnas pueden ser elegidas más de una vez)

Si se elige la columna k de A (con la fila k de B), **multiplicar ambas por $1/\sqrt{sp_k}$**

Luego (columna k de A) (fila k de B)$/sp_k$ es un sumando de nuestro producto aleatorio AB.

Conclusión a verificar : **el valor esperado de la matriz SS^{T} de n por n es I**

La misma conclusión en otras palabras : **el valor esperado de $ASS^{\mathrm{T}}B$ es AB**

Así el muestreo aleatorio calcula el producto matricial que tiene **media correcta AB**

Prueba. Hay s muestras idénticas. Cada muestra elige un par columna-fila de A y B con probabilidades p_1 a p_n (columna de A por fila de B dividido por $\sqrt{sp_j}^{\,2} = sp_j$). Así el valor esperado = *media* = resultado promedio de cada muestra es

$$p_1 \frac{(\text{columna 1 de } A)\ (\text{fila 1 de } B)}{sp_1} + \cdots + p_n \frac{(\text{columna } n \text{ de } A)\ (\text{fila } n \text{ de } B)}{sp_n}. \tag{4}$$

Los p se cancelan. **Esto es exactamente AB/s.** Y ya que hay s muestras, **el valor esperado para la multiplicación aleatorizada $(AS)(S^{\mathrm{T}}B)$ es AB.**

Conclusión Todo bien por ahora, pero tenemos que elegir las probabilidades p_1 a p_n. Cualquier elección (de valores que sumen 1) da el valor esperado correcto AB (la media). Pero la elección de los p puede afectar marcadamente la varianza.

Un muestreo uniforme elegiría probabilidades iguales $p = 1/n$. Esto es razonable si las columnas de A (y también las filas de B) tienen magnitudes similares. Pero supongamos que (columna 1 de A) (fila 1 de B) conforma la mayor parte de AB: domina los otros productos externos columna-fila. Entonces no queremos perdernos este término por el azar.

Nuestro enfoque aquí es usar *probabilidades diferentes* p_j. Ahora definimos y calculamos los mejores p. Y mencionamos una opción totalmente distinta : introducir una matriz de mezcla M y trabajar con AM y $M^{-1}B$. Luego usamos probabilidades iguales p_j para las columnas de A y filas de B mezcladas al azar.

El muestreo por norma al cuadrado minimiza la varianza

En el muestreo por norma al cuadrado se eligen las probabilidades p_j proporcionales a los números $||\text{columna } j \text{ de } A||\ ||\text{fila } j \text{ de } B||$. **En el caso importante $B = A^{\mathrm{T}}$**, los p_j son proporcionales a $||\text{columna } j \text{ de } A||^2$. El nombre "*norma al cuadrado*" o "*longitud al cuadrado*" es entonces natural. Todavía tenemos que escalar todas las probabilidades p_j por una constante apropiada C para que sumen 1 :

$$\boxed{p_j = \frac{1}{C}\,||\textbf{columna } j \textbf{ de } A||\ ||\textbf{fila } j \textbf{ de } B|| = \frac{||a_j||\ ||b_j^{\mathrm{T}}||}{C} \text{ con } C = \sum_{j=1}^{n} ||a_j||\ ||b_j^{\mathrm{T}}||.} \tag{5}$$

Ahora calcularemos minuciosamente la varianza para la multiplicación matricial aleatorizada con probabilidades p_j y verificaremos que *las elecciones de p en la ecuación* (5) *minimizan la varianza.* Es mejor elegir columnas y filas grandes con más probabilidad. Las líneas que siguen a la ecuación (4) ya mostraron que todas las opciones dan la media correcta $\mathrm{E}\,[ASS^{\mathrm{T}}B] = AB$.

Cada una de las s muestras produce una matriz $X_j = \boldsymbol{a}_j \boldsymbol{b}_j^{\mathrm{T}}/sp_j$ con probabilidad p_j. Su elemento i,k es $(X_j)_{ik} = a_{ij}b_{jk}/sp_j$. En cada muestra calculamos la media : los p_j se cancelan.

$$\textbf{Media} \qquad \mathrm{E}[X] = \sum_{j=1}^{n} p_j X_j = \frac{1}{s}\sum_{1}^{n} \boldsymbol{a}_j \boldsymbol{b}_j^{\mathrm{T}} = \frac{1}{s} AB \ \text{ como en } (4)$$

La varianza de una muestra es por definición $\mathbf{E[X^2] - (E[X])^2}$. Sumar los resultados para s ensayos independientes multiplica la media de un ensayo por s y la varianza por s^2. La media se vuelve AB como la conocemos. La varianza será calculada en la norma de Frobenius (suma de cuadrados de los elementos de una matriz). Comparemos la AB correcta con la CR aleatoria :

$$\begin{aligned} \textbf{Varianza} \quad \mathrm{E}\left[\left|\left|CR - AB\right|\right|_F^2\right] &= \sum_{i,k}\sum_{j=1}^{n} p_j \frac{a_{ij}^2\, b_{jk}^2}{sp_j^2} - \frac{1}{s}\left|\left|AB\right|\right|_F^2 \\ \text{(sumando primero sobre } i \text{ y } k) &= \sum_{j=1}^{n} \frac{||\boldsymbol{a}_j||^2\, ||\boldsymbol{b}_j^{\mathrm{T}}||^2}{sp_j} - \frac{1}{s}\left|\left|AB\right|\right|_F^2 \end{aligned} \tag{6}$$

Por último elegimos probabilidades $p_1, \ldots, p_n$ para minimizar esta varianza. La ecuación (5) revela la opción minimizante, demostrada a continuación. Para esa opción $p_j = ||\boldsymbol{a}_j||\, ||\boldsymbol{b}_j^{\mathrm{T}}||/\boldsymbol{C}$ de (5), los términos $||\boldsymbol{a}_j||^2\, ||\boldsymbol{b}_j^{\mathrm{T}}||^2/p_j$ en la ecuación (6) se vuelven $C||\boldsymbol{a}_j||\, ||\boldsymbol{b}_j^{\mathrm{T}}||$. **Su suma es $\boldsymbol{C^2}$.** La **menor varianza** (con esos p_j óptimos) es nuestro resultado final :

$$\boxed{\mathrm{E}\left[\left|\left|AB - CR\right|\right|_F^2\right] = \frac{1}{s}\left[\sum_{j=1}^{n} ||\boldsymbol{a}_j||\, ||\boldsymbol{b}_j^{\mathrm{T}}||\right]^2 - \frac{1}{s}||AB||_F^2 = \frac{\mathbf{1}}{\boldsymbol{s}}\left(\boldsymbol{C^2} - \left|\left|\boldsymbol{AB}\right|\right|_{\boldsymbol{F}}^{\mathbf{2}}\right).} \tag{7}$$

Aquí está la prueba de que (5) da las probabilidades p_j que minimizan la varianza en (6). Multiplicamos la restricción $p_1 + \cdots + p_n = 1$ por un multiplicador de Lagrange λ. Lo sumamos a la función en (6). Esta es la clave de los multiplicadores de Lagrange :

$$L(p_1, \ldots, p_n, \lambda) = \sum_{j=1}^{n} \frac{||\boldsymbol{a}_j||^2\, ||\boldsymbol{b}_j^{\mathrm{T}}||^2}{sp_j} - \frac{1}{s}\left|\left|AB\right|\right|_F^2 + \lambda\left(\sum_{1}^{n} p_j - 1\right)$$

Tomamos derivadas parciales $\partial L/\partial p_j$ para encontrar los p_j minimizantes (probabilidades óptimas) :

$$\boxed{\frac{\partial L}{\partial p_j} = 0 \ \text{ se convierte en } \ \frac{1}{sp_j^2}\, ||\boldsymbol{a}_j||^2\, ||\boldsymbol{b}_j^{\mathrm{T}}||^2 = \lambda} \tag{8}$$

Esto dice que $p_j = ||\boldsymbol{a}_j||\, ||\boldsymbol{b}_j^{\mathrm{T}}||/\sqrt{s\lambda}$. Elegimos el multiplicador de Lagrange λ para que $\sum p_j = 1$.

$$\sum_{1}^{n} p_j = \sum_{1}^{n} \frac{||\boldsymbol{a}_j||\, ||\boldsymbol{b}_j^{\mathrm{T}}||}{\sqrt{s\lambda}} = 1 \text{ da } \sqrt{\boldsymbol{s\lambda}} = \boldsymbol{C} \text{ y } p_j = \frac{||\boldsymbol{a}_j||\, ||\boldsymbol{b}_j^{\mathrm{T}}||}{\boldsymbol{C}} \text{ como se predijo en (5).}$$

El muestreo por norma al cuadrado usando las probabilidades óptimas $\boldsymbol{p_j}$ da varianza mínima.

Para matrices muy grandes, almacenadas por fuera de la memoria de acceso aleatorio (RAM, por su sigla en inglés), el muestreo por norma al cuadrado puede pedirte leer la matriz dos veces. La primera lectura calcula la magnitud al cuadrado de cada columna de A y fila de B. Luego (dentro de la RAM) se encuentran las probabilidades p_j y se eligen s columnas y filas para el muestreo. La segunda lectura coloca la aproximación del muestreo CR en la memoria rápida.

Aplicaciones de multiplicación matricial aleatorizada

El muestreo por norma al cuadrado = muestreo por longitud al cuadrado puede ayudar a resolver estos problemas centrales del álgebra lineal numérica :

1	**Aproximación interpolante $A \approx CMR$: C y R usan columnas y filas de A**
2	**Aproximación de A por una matriz de rango pequeño**
3	**Aproximación de la SVD de A**

CMR busca producir un "bosquejo" preciso de A a partir de k de sus columnas y sus filas. Las columnas irán en C y las filas irán en R. Luego una *matriz de mezcla* M conecta C con R para producir $CMR \approx A$. Las dimensiones son $(m \times k)\,(k \times k)\,(k \times n) = (m \times n)$. Si A es rala, entonces C y R serán ralas, porque provienen directamente de A.

Observe la multiplicación rápida $(C(M(Rv)))$. Nunca multiplicamos explícitamente CMR.

Entendamos primero que $A \approx CR$ probablemente no sea cierto. El espacio columna de A será captado con precisión por C (eso esperamos). El espacio fila de A será captado por R. "*Los espacios generados son los mismos pero no la matriz.*" Toda matriz de la forma CMR tiene los mismos espacios columna y espacios fila (para M invertible). Queremos elegir M de tal manera que CMR sea cercana a A. Seguimos evitando la notación CUR y reservamos la letra U para la SVD.

Para comenzar, buscaremos la matriz de mezcla M que en teoría sea la mejor. Las opciones buenas para M han sido desarrolladas y probadas en la bibliografía sobre muestreo. He aquí seis referencias importantes de álgebra lineal aleatorizada :

N. Halko, P. -G. Martinsson, y J. A. Tropp, Finding structure with randomness : probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review* **53** (2011) 217-288.
R. Kannan y S. Vempala, Randomized algorithms in numerical linear algebra, *Acta Numerica* **26** (2017) 95-135.
E. Liberty, F. Woolfe, P. -G. Martinsson, V. Rokhlin y M. Tygert, Randomized algorithms for the low-rank approximation of matrices, *PNAS* **104** (2007) no. 51, 20167-20172.
M. W. Mahoney, Lecture Notes on Randomized Linear Algebra. arXiv :1608.04481.
P.-G. Martinsson, Compressing rank-structured matrices via randomized sampling, arXiv :1503.07152.
D. P. Woodruff, Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science* **10** (2014) 1-157.

La mejor M en $A \approx CMR$: norma de Frobenius y norma L^2

Tenemos A, C, R. Supongamos que Q_C contiene una base ortonormal para el espacio columna de C. Así $Q_C Q_C^{\mathrm{T}}$ es la matriz de proyección sobre ese subespacio. De manera similar Q_R contiene una base ortonormal para $\mathbf{C}(R^{\mathrm{T}})$ y $Q_R Q_R^{\mathrm{T}}$ es la matriz de proyección en ese espacio fila. $Q_R^{\perp}$ contiene una base ortonormal para el espacio nulo $\mathbf{N}(R)$, y $Q_C^{\perp}$ para $\mathbf{N}(C^{\mathrm{T}})$.

Por definición, la proyección de A en espacios fila/columna es $\boldsymbol{\widehat{A} = Q_C^{\mathrm{T}} A Q_R}$. El propósito de estas proyecciones es separar los subespacios de donde elegimos M de los subespacios que no podemos cambiar. Yuji Nakatsukasa aclaró y resolvió este problema (conversaciones en Oxford). En la norma de Frobenius, la solución es muy prolija.
Las matrices ortogonales $\left[\begin{array}{cc} Q_C & Q_C^{\perp} \end{array}\right]$ y $\left[\begin{array}{cc} Q_R & Q_R^{\perp} \end{array}\right]$ no cambiarán las normas de Frobenius y L^2 de $A - CMR$. Pero nos ayudan a encontrar la M óptima que minimiza este error.

$$\left[\begin{array}{cc} Q_C & Q_C^{\perp} \end{array}\right]^{\mathrm{T}} (A - CMR) \left[\begin{array}{cc} Q_R & Q_R^{\perp} \end{array}\right] = \left[\begin{array}{cc} \boldsymbol{\widehat{A}} & \boldsymbol{X} \\ \boldsymbol{Y} & \boldsymbol{Z} \end{array}\right] - \left[\begin{array}{cc} \boldsymbol{\widehat{C}M\widehat{R}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{array}\right]. \tag{9}$$

¿Deberíamos elegir M para que $\widehat{A} = \widehat{C}M\widehat{R}$? En la norma de Frobenius, esto es óptimo.

Frobenius trata por separado cada elemento en cada bloque de estas matrices. El error es más pequeño cuando el bloque superior izquierdo es exacto : $\widehat{C}M\widehat{R} = \widehat{A}$. Este bloque de la esquina es el (único) problema que controlamos. Hemos encontrado la misma M que en las secciones II.3 y I.1.

Norma de Frobenius

$$\boxed{\min_M \; ||A - CMR||_{\boldsymbol{F}} = \left\| \left[\begin{array}{cc} \mathbf{0} & \boldsymbol{X} \\ \boldsymbol{Y} & \boldsymbol{Z} \end{array}\right] \right\|_{\boldsymbol{F}}.} \tag{10}$$

En la norma matricial L^2, podríamos esperar que el mismo plan tenga éxito. Pero un bloque cero en esa esquina puede no dar la norma L^2 más pequeña.

Ejemplo $\left\| \left[\begin{array}{cc} \mathbf{-1} & 1 \\ 1 & 1 \end{array}\right] \right\|_{\boldsymbol{L^2}}$ es más pequeña que $\left\| \left[\begin{array}{cc} \mathbf{0} & 1 \\ 1 & 1 \end{array}\right] \right\|_{\boldsymbol{L^2}}$ (11)

A la izquierda, las columnas son ortogonales con longitud $\sqrt{2}$. Los cuadrados de los valores singulares son ambos $\sigma^2 = 2$. A la derecha, el cuadrado del mayor valor singular es $\sigma_1^2 = \frac{1}{2}(3 + \sqrt{5}) > 2$. **La submatriz cero hizo que la norma $\boldsymbol{L^2}$ mayor de toda la matriz fuera mayor.**

El problema de la submatriz óptima es un problema hermoso resuelto por Davis, Kahan, y Weinberger en 1982. La M óptima a menudo no satisface $\widehat{C}M\widehat{R} = \widehat{A}$. Pero reduce la norma L^2 al mayor número entre $||[Y \;\; Z]||$ y $||[X^{\mathrm{T}} \;\; Z^{\mathrm{T}}]||$: una norma L^2 más pequeña es imposible en (10).

En el ejemplo, la norma L^2 más pequeña la logra esa submatriz -1 (**no nula**).

Factorizaciones matriciales aleatorizadas

Haremos ahora un resumen de nuestro tema. Las matrices ortogonales son el objetivo en $A = QR$ y $A = U\Sigma V^{\mathrm{T}}$. La matriz A es demasiado grande para factorizaciones exactas, tal vez demasiado grande para leer cada elemento a_{ij}. Si partimos de una aproximación $\boldsymbol{A \approx CB}$ $(m \times k)$ $(k \times n)$, entonces Halko-Martinsson-Tropp encuentran las factorizaciones QR y $U\Sigma V^{\mathrm{T}}$ rápido y con precisión.

He aquí descomposiciones rápidas a partir de una $A \approx CB$ aleatorizada.

$(\boldsymbol{A \approx QR})$	Factorizar C en Q_1R_1. Factorizar R_1B en Q_2R_2. Luego $A \approx (Q_1Q_2)R_2$.
$(\boldsymbol{A \approx U\Sigma V^{\mathrm{T}}})$	A partir de $C = Q_1R_1$ factorizar R_1B en $U_2\Sigma V^{\mathrm{T}}$. Elegir $U = Q_1U_2$.

Proyecciones J y matrices de proyecciones P

Una "*proyección*" del espacio m-dimensional $\mathbf{R}^m$ al espacio k-dimensional $\mathbf{R}^k$ es una matriz de k por m. En caso de que esa matriz $\boldsymbol{J}$ tenga rango completo por filas k, entonces $k \leq m$ y las proyecciones completan todo $\mathbf{R}^k$.

Un caso especialmente lindo es cuando las k filas de J son ortonormales. Lo que significa que $JJ^{\mathrm{T}} = I$. Así el rango es sin dudas k y el espacio columna $\mathbf{C}(J)$ es todo $\mathbf{R}^k$.

Observemos que una *proyección* J es diferente de una *matriz de proyección* P. P es cuadrada (m por m). Su rango es $k < m$ (excepto cuando $P = I$). Su espacio columna es un subespacio k-dimensional de $\mathbf{R}^m$ (muy distinto del espacio vectorial $\mathbf{R}^k$). Y la propiedad clave de la matriz de proyección es $\boldsymbol{P^2 = P}$. Si proyectamos *perpendicularmente* sobre el espacio columna, la proyección $P\boldsymbol{b}$ es la más cercana a $\boldsymbol{b}$ en la norma usual $||\boldsymbol{b} - P\boldsymbol{b}||$. En consecuencia P también es simétrica.

J y P tienen una linda conexión. **En el caso que $\boldsymbol{J}$ tenga filas ortonormales, entonces $\boldsymbol{J^{\mathrm{T}}J}$ es una matriz de proyección simétrica $\boldsymbol{P}$.** Podemos observar que $P^2 = J^{\mathrm{T}}(JJ^{\mathrm{T}})J = J^{\mathrm{T}}IJ = P$.

Ejemplo 1 **Proyección** $\boldsymbol{J} = [\,\cos\theta \;\; \operatorname{sen}\theta\,]$ con $JJ^{\mathrm{T}} = [\cos^2\theta + \operatorname{sen}^2\theta] = [1]$

$$\textbf{Matriz de proyección } \boldsymbol{P = J^{\mathrm{T}}J} = \begin{bmatrix} \cos^2\theta & \cos\theta\operatorname{sen}\theta \\ \cos\theta\operatorname{sen}\theta & \operatorname{sen}^2\theta \end{bmatrix} = \boldsymbol{P^2}$$

Una P simétrica proyecta **ortogonalmente** sobre su espacio columna. Aquí este es $\mathbf{C}(J^{\mathrm{T}})$.

Proyecciones aleatorias

Supongamos que los elementos de una proyección J de k por m son variables aleatorias independientes. En el caso más simple son tomadas de una distribución normal (una gaussiana) con media $m = 0$ y varianza $\sigma^2 = 1/k$. Mostraremos que **el valor esperado de $\boldsymbol{P = J^{\mathrm{T}}J}$ es la matriz identidad**. En otras palabras, la **longitud esperada de la proyección $\boldsymbol{v = Ju}$ es igual a la longitud de $\boldsymbol{u}$.**

1. El elemento (i,i) sobre la diagonal de $J^{\mathrm{T}}J$ es la suma $J_{1i}^2 + \cdots + J_{ki}^2$. Esos cuadrados son muestras independientes, cada una con media $1/k$. (Si cada elemento de J tiene media cero, el valor esperado del cuadrado es la varianza $\sigma^2 = 1/k$.) Entonces la media (el valor esperado) de la suma de los k términos es $\boldsymbol{k(1/k) = 1}$.

2. El elemento (i,j) fuera de la diagonal de $J^{\mathrm{T}}J$ es la suma $J_{1i}J_{1j} + \cdots + J_{ki}J_{kj}$. Cada uno de esos términos es el producto de dos variables independientes con media cero. Así la media de cada término es cero, y la media de su suma $(J^{\mathrm{T}}J)_{ij}$ también es cero.

Por lo tanto $\mathbf{E}[\boldsymbol{J^{\mathrm{T}}J}] = \boldsymbol{I}$ = matriz identidad de tamaño m.

Usando columnas por filas, cada matriz (fila i de $J)^{\mathrm{T}}$ (fila i de J) tiene esperanza I/k. La suma tiene esperanza I. Por favor observe lo que eso significa.

"En esperanza" las m columnas de J son ortonormales.
Por lo tanto, en esperanza, $J\boldsymbol{x}$ tiene la misma longitud que $\boldsymbol{x}$:

$$\mathbf{E}\left[\,||J\boldsymbol{x}||^2\,\right] = \mathbf{E}\left[\,\boldsymbol{x}^{\mathrm{T}}J^{\mathrm{T}}J\boldsymbol{x}\,\right] = \mathbf{E}\left[\,\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}\,\right] = ||\boldsymbol{x}||^2. \tag{12}$$

Esta proyección aleatoria muestra por qué el álgebra lineal y la probabilidad a veces parecen mundos diferentes dentro de la matemática. En álgebra lineal, una matriz J de k por m con $k < m$ no podría tener rango m. **Entonces $J^{\mathrm{T}}J$ no podría tener rango m.** Pero tomemos ahora los valores esperados: E[J] puede ser la matriz cero, E[$J^{\mathrm{T}}J$] puede ser la matriz identidad. Para matrices de 1 por 1 esto es lo más común del mundo : *media cero y varianza uno.*

Este es el punto clave. En promedio, la distancia al cuadrado entre $\boldsymbol{x}$ e $\boldsymbol{y}$ en $\mathbf{R}^m$ es la misma que la distancia al cuadrado entre $J\boldsymbol{x}$ y $J\boldsymbol{y}$ en $\mathbf{R}^k$. "La proyección J en un espacio de menos dimensiones preserva distancias en un sentido promedio." **Lo que en realidad queremos es preservar las distancias originales dentro de un factor $1+\epsilon$, para un conjunto de n puntos dados en $\mathbf{R}^m$.**

Esa propiedad de J es el tema del famoso lema de Johnson-Lindenstrauss. Parece increíble que puntos en una dimensión grande puedan transformarse linealmente a puntos en una dimensión pequeña, con cambios muy pequeños en la distancia entre cada par de puntos

¿ *Qué tan pequeña puede ser la dimensión* ? Esta es una pregunta crucial.

La dimensión $k = 1$ es demasiado pequeña

Supongamos que tenemos $n = 3$ puntos $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$ en el plano $\mathbf{R}^2$. ¿ Existe alguna matriz J de 1 por 2 que casi preserve sus distancias ? Sin duda podemos lograr $||J\boldsymbol{x}_1 - J\boldsymbol{x}_2|| = ||\boldsymbol{x}_1 - \boldsymbol{x}_2||$. Si el tercer punto $\boldsymbol{x}_3 = \frac{1}{2}(\boldsymbol{x}_1 + \boldsymbol{x}_2)$ está a mitad de camino entre los dos, entonces por la linealidad $J\boldsymbol{x}_3$ será $\frac{1}{2}(J\boldsymbol{x}_1 + J\boldsymbol{x}_2)$ y todas las distancias son perfectas. Pero si $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$ dan un triángulo equilátero, entonces $\boldsymbol{x}_3$ tiene una componente en el espacio nulo de J. Esa componente se perderá (se proyecta a cero). Las longitudes $||J\boldsymbol{x}_1 - J\boldsymbol{x}_3||$ y $||J\boldsymbol{x}_2 - J\boldsymbol{x}_3||$ se verán muy reducidas.

Johnson-Lindenstrauss trabajaron con **proyecciones aleatorias de k por m de n puntos de $\mathbf{R}^m$**. Probaron que **si la dimensión k es suficientemente grande**, entonces una de esas proyecciones (de hecho la mayoría) casi preservarán las distancias entre los n puntos.

"En dimensiones grandes, los vectores aleatorios son virtualmente ortogonales con probabilidad cercana a 1."

El lema de Johnson-Lindenstrauss

Supongamos que $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ son n puntos cualesquiera en $\mathbf{R}^m$, y $k \geq (8\log n)/\epsilon^2$. Entonces existe una proyección J de $\mathbf{R}^m$ en $\mathbf{R}^k$ tal que todas las distancias entre los n puntos están casi preservados :

$$(1-\epsilon)\,||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 \leq ||J\boldsymbol{x}_i - J\boldsymbol{x}_j||^2 \leq (1+\epsilon)\,||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 \tag{13}$$

Un punto clave es que la dimensión k debe crecer como $(\log n)/\epsilon^2$. Un paso sorprendente en la prueba muestra que es muy probable que una *proyección aleatoria J de k por m* mantenga los n puntos separados. Entonces debe haber *muchas J específicas* que confirmen la ecuación (13) en el lema.

> Si una opción aleatoria tiene una probabilidad positiva de éxito entonces debe existir una opción exitosa. Hipótesis probabilística, conclusión determinista.

Ese es el "*método probabilístico*". Múltiples pruebas de Johnson-Lindenstrauss pueden encontrarse con facilidad, y nos gustó la de este sitio web : **cseweb.ucsd.edu/∼dasgupta/papers/jl.pdf.**

Uno de los proyectos del curso 18.065 trató un ejemplo donde el lema requiere de $k > 2800$. La precisión de las distancias $||\boldsymbol{x}_i - \boldsymbol{x}_j||$ se estropeó para $k = 2700$. Otra cosa interesante : el agrupamiento de los puntos sobrevivió incluso cuando las distancias fallaron.

Aproximación matricial aleatorizada

Para resumir este tema seguiremos los pasos de *Randomized methods for matrix computations* de Per-Gunnar Martinsson, arXiv : 1607.01649. Primero identificamos los objetivos.

1 Factorizaciones de rango k $\boldsymbol{A \approx Y(Y^+A)}$ y $\boldsymbol{S \approx UDU^{\mathrm{T}}}$ y $\boldsymbol{A \approx (QU)DV^{\mathrm{T}}}$.

2 Descomposiciones interpolantes $\boldsymbol{A \approx CMR}$ y $\boldsymbol{A \approx CZ}$ con columnas de A en C.

El factor aleatorizado Y de m por k es AG, con una matriz aleatoria gaussiana G de n por k. La proyección ortogonal en el espacio columna de Y es siempre YY^+. Entonces YY^+A es casi seguramente una aproximación muy buena de rango k de A. Esta es la parte aleatoria.

¿ Cómo se obtiene ahora una SVD aproximada de $A \approx YY^+A$? Para una base ortonormal del espacio columna, primero aplicar la factorización QR a Y. QQ^T es igual a la proyección YY^+. Luego encontrar la SVD de la matriz pequeña $Q^{\mathrm{T}}A = UDV^{\mathrm{T}}$. La SVD deseada es $\boldsymbol{A \approx (QU)DV^{\mathrm{T}}}$.

Observemos la construcción en dos etapas de esa SVD. Primero fijamos un espacio columna aproximado, reduciendo el problema a "tamaño k". Luego cualquier factorización deseada es determinista y rápida, y en esencia exacta. Martinsson muestra cómo esas etapas se pueden combinar en un *algoritmo de transmisión* que accede a cada elemento de A solo una vez. El precio es el mal condicionamiento y la cura es el sobremuestreo.

Para S semidefinida positiva, un *paso de Nyström* adicional mejora a bajo costo los factores.

Finalmente consideramos $A \approx CZ$ o CMR, donde C contiene columnas de A : preservando raleza y no negatividad. La precisión de un algoritmo determinista es la misma que la de $AP \approx QR$ con pivoteo (no siempre tan cercana como Eckart-Young). Los algoritmos aleatorizados son más rápidos y mejores si los valores singulares de A decaen rápidamente como en la Sección III.3.

Para una explicación clara de algoritmos matriciales aleatorizados, lea el artículo de Martinsson y Randomized algorithms in numerical linear algebra de Kannan-Vempala, *Acta Numerica* (2017), 95-135. Estos autores descubrieron el muestreo por norma al cuadrado.

Problemas de la Sección II.4

1 Dados los números positivos $a_1, \ldots, a_n$ encuentre los números positivos $p_1, \ldots, p_n$ tales que

$$p_1 + \cdots + p_n = 1 \quad \text{y} \quad V = \frac{a_1^2}{p_1} + \cdots + \frac{a_n^2}{p_n} \text{ alcanza su mínimo } (a_1 + \cdots + a_n)^2.$$

Las derivadas de $L(p, \lambda) = V + \lambda(p_1 + \cdots + p_n - 1)$ son cero como en la ecuación (8).

2 (*para funciones*) Dada $a(x) > 0$ encuentre $p(x) > 0$ por analogía con el Problema 1, tal que

$$\int_0^1 p(x)\, dx = 1 \quad \text{y} \quad \int_0^1 \frac{(a(x))^2}{p(x)}\, dx \quad \text{es un mínimo.}$$

3 Demuestre que $n(a_1^2 + \cdots + a_n^2) \geq (a_1 + \cdots + a_n)^2$. Este es el Problema 1 con $p_i = 1/n$. En los Problemas I.11, demostró que $||\boldsymbol{a}||_1 \leq \sqrt{n}||\boldsymbol{a}||_2$.

4 Si $M = \mathbf{1}\,\mathbf{1}^{\mathrm{T}}$ es la matriz de unos de n por n, pruebe que $nI - M$ es semidefinida positiva. En el Problema 3 hicimos la prueba de la energía. Para el Problema 4, encuentre los autovalores de $nI - M$.

5 En el caso $B = A^{\mathrm{T}}$ muestre que las probabilidades de "norma al cuadrado" o "longitud al cuadrado" p_j en la ecuación (5) son $||\boldsymbol{a}_j||^2/||A||_F^2$. ¿Por qué es $C = \sum ||\boldsymbol{a}_j||\, ||\boldsymbol{b}_j|| = ||A||_F^2$?

6 ¡La varianza calculada en la ecuación (7) no puede ser negativa ! Muéstrelo directamente :

$$||AB||_F^2 \leq (\sum ||\boldsymbol{a}_j||\, ||\boldsymbol{b}_j^{\mathrm{T}}||)^2.$$

En el Problema 7 regresamos al ejemplo del texto, de muestreo de (a, b) para obtener $(a, 0)$ o $(0, b)$. Si $b > a$, entonces la varianza se reducirá cuando b se elige más a menudo. Esto se logra optimizando las probabilidades p y $1 - p$ que minimizan σ^2 :

$$\text{Varianza} \quad \sigma^2 = p\,\frac{a^2}{p^2} + (1-p)\,\frac{b^2}{(1-p)^2} - (\text{media})^2$$

7 Muestre que $p = a/(a + b)$ y $1 - p = b/(a + b)$ minimizan esa varianza. (La media es la misma para todos los p.) Este p óptimo coincide con la ecuación (5) cuando se aplica a la multiplicación de matrices pequeñas $AB = [1]\,[a \;\; b]$. En este caso $C = a + b$ en la ecuación (5).

8 En la construcción aleatorizada de la página anterior, muestre por qué $(QU)DV^{\mathrm{T}}$ (*la* SVD *aproximada en cursiva*) está cerca de A. Use los pasos $A \approx YY^{+}A$ e $Y \approx QR$ y $Q^{\mathrm{T}}A = UDV^{\mathrm{T}}$. *El problema* II.2.11 *en la página* 135 *contiene una clave.*

Parte III

Rango bajo y compressed sensing

III.1 **Cambios en A^{-1} a partir de cambios en A**

III.2 **Autovalores entrelazados y señales de rango bajo**

III.3 **Valores singulares que decaen rápido**

III.4 **Algoritmos de partición para $\ell^2 + \ell^1$**

III.5 **Compressed sensing y completación de matrices**

Parte III : Rango bajo y compressed sensing

En esta parte del libro se analizan tres clases de matrices de rango bajo :

1. Matrices que tienen **rango notoriamente bajo** ($\boldsymbol{uv}^{\mathrm{T}}$ es un caso extremo con rango uno)
2. Matrices que tienen valores singulares exponencialmente decrecientes (**rango efectivo bajo**).
3. **Matrices incompletas** (con elementos faltantes) que se completan con matrices de rango bajo.

Las de la primera clase no son invertibles (porque el rango $< n$). Las de la segunda clase son, en teoría, invertibles, pero no en la práctica. La matriz de Hilbert con elementos $(i+j-1)^{-1}$ es un famoso ejemplo. ¿ Cómo podemos reconocer que esta u otra matriz tiene un rango efectivo muy bajo ?

La tercera cuestión—la completación de matrices—se abarca en la Sección III.5. Creamos un problema de minimización que se aplica a las matrices de recomendación :

Minimizar $||A||_N$ sobre todas las opciones posibles para los elementos faltantes.

Esa "norma nuclear" da origen a un problema bien planteado que reemplaza a un problema no convexo : minimizar el rango. *Se conjetura que las normas nucleares son importantes en los métodos de descenso por gradiente.*

El rango de una matriz se corresponde de alguna manera profunda con la cantidad de valores no nulos de un vector. Con esa analogía, una matriz de rango bajo es como un **vector ralo**. Repito: la cantidad de valores no nulos en $\boldsymbol{x}$ no es una norma. Ese número a veces se escribe $||\boldsymbol{x}||_0$, pero esta **"norma $\boldsymbol{\ell^0}$"** no cumple con el requisito $||2\boldsymbol{x}|| = 2||\boldsymbol{x}||$. No se duplica la cantidad de valores no nulos.

Es esencial encontrar soluciones ralas de $A\boldsymbol{x} = \boldsymbol{b}$. Parece un milagro que al minimizar la **norma $\boldsymbol{\ell^1}$** $||\boldsymbol{x}||_1 = |x_1| + \cdots + |x_n|$ se obtengan soluciones ralas. Este hecho nos lleva a un nuevo mundo del **compressed sensing**, con aplicaciones en ingeniería y medicina (que incluye modificaciones en las máquinas de resonancia magnética). Los algoritmos para la minimización $\boldsymbol{\ell^1}$ se desarrollan y comparan en la Sección III.4.

La Sección III.1 inicia este capítulo con una fórmula famosa para $(I - \boldsymbol{uv}^{\mathrm{T}})^{-1}$ y $(A - \boldsymbol{uv}^{\mathrm{T}})^{-1}$. Es la fórmula de Sherman-Morrison-Woodbury. Muestra que el cambio en la matriz inversa también tiene rango 1 (si la matriz es invertible). Esta fórmula con su extensión $(A - UV^{\mathrm{T}})^{-1}$ a perturbaciones de mayor rango es fundamental.

También calculamos las derivadas de $A(t)^{-1}$ y $\lambda(t)$ y $\sigma(t)$ cuando A varía con t.

III.1. Cambios en A^{-1} a partir de cambios en A

Supongamos que le restamos a A una matriz de rango pequeño. La siguiente sección estima el cambio que esto produce en los autovalores y en los valores singulares. Esta sección encuentra una fórmula exacta para el cambio en A^{-1}. Algunos autores llaman a esta fórmula **lema de inversión de matriz**. Para otros es conocida como la **fórmula de Sherman-Morrison-Woodbury**. Esos nombres de la ingeniería y la estadística corresponden con las fórmulas de actualización (*updating/downdating*) del análisis numérico.

Esta fórmula es la clave para actualizar la solución de un sistema lineal $A\boldsymbol{x} = \boldsymbol{b}$. El cambio puede estar en una fila o columna existente de A, o en agregar o quitar una fila o columna. Esos serían cambios de rango uno. Comencemos con este sencillo ejemplo, cuando $A = I$.

$$\boxed{\textbf{La inversa de } \ M = I - uv^{\mathrm{T}} \ \textbf{ es } \ M^{-1} = I + \frac{uv^{\mathrm{T}}}{1 - v^{\mathrm{T}}u}} \tag{1}$$

Hay dos características notables de esta fórmula. La primera es que la corrección en M^{-1} **es también de rango uno**. Es el último término $uv^{\mathrm{T}}/(1-v^{\mathrm{T}}u)$ de la fórmula. La segunda característica es que el término de corrección podría ser infinito. **Entonces M no es invertible:** no existe M^{-1}.

Esto ocurre si el número $v^{\mathrm{T}}u$ es 1. La Ecuación (1) termina con una división por cero. En este caso la fórmula falla. $M = I - uv^{\mathrm{T}}$ no es invertible porque $Mu = 0$:

$$\text{si} \quad v^{\mathrm{T}}u = 1, \ Mu = (I - uv^{\mathrm{T}})u = u - u(v^{\mathrm{T}}u) = 0. \tag{2}$$

La prueba más simple de la fórmula (1) es la multiplicación directa de M por M^{-1}:

$$MM^{-1} = (I - uv^{\mathrm{T}})\left(I + \frac{uv^{\mathrm{T}}}{1 - v^{\mathrm{T}}u}\right) = I - uv^{\mathrm{T}} + \frac{(I - uv^{\mathrm{T}})uv^{\mathrm{T}}}{1 - v^{\mathrm{T}}u} = I - uv^{\mathrm{T}} + uv^{\mathrm{T}}. \tag{3}$$

Podemos ver cómo el número $v^{\mathrm{T}}u$ se mueve afuera de la matriz uv^{T} en ese último paso clave.

Hemos mostrado que la fórmula (1) es correcta, pero no de dónde vino. Una buena manera es introduciendo una "extensión" de I a una matriz E con una fila y una columna nuevas:

$$\boxed{\textbf{Matriz extendida} \qquad E = \begin{bmatrix} I & u \\ v^{\mathrm{T}} & 1 \end{bmatrix} \text{ tiene determinante } D = 1 - v^{\mathrm{T}}u}$$

La eliminación da dos maneras de encontrar E^{-1}. Primero, restar de la fila 2 de E, v^{T} por la fila 1:

$$\begin{bmatrix} I & 0 \\ -v^{\mathrm{T}} & 1 \end{bmatrix} E = \begin{bmatrix} I & u \\ 0 & D \end{bmatrix}. \quad \text{Entonces } \ E^{-1} = \begin{bmatrix} I & u \\ 0 & D \end{bmatrix}^{-1} \begin{bmatrix} I & 0 \\ -v^{\mathrm{T}} & 1 \end{bmatrix} \tag{4}$$

La segunda manera es restar a la primera fila de E, $\boldsymbol{u}$ por su segunda fila:

$$\begin{bmatrix} I & -\boldsymbol{u} \\ 0 & 1 \end{bmatrix} E = \begin{bmatrix} I - \boldsymbol{u}\boldsymbol{v}^{\mathrm{T}} & 0 \\ \boldsymbol{v}^{\mathrm{T}} & 1 \end{bmatrix}. \quad \text{Entonces } E^{-1} = \begin{bmatrix} I - \boldsymbol{u}\boldsymbol{v}^{\mathrm{T}} & 0 \\ \boldsymbol{v}^{\mathrm{T}} & 1 \end{bmatrix}^{-1} \begin{bmatrix} I & -\boldsymbol{u} \\ 0 & 1 \end{bmatrix}. \tag{5}$$

Ahora comparemos esas dos fórmulas para **la misma $\boldsymbol{E^{-1}}$**. El Problema 2 muestra el álgebra:

$$\textbf{Dos formas de } \boldsymbol{E^{-1}} \qquad \begin{bmatrix} \boldsymbol{I + uD^{-1}v^{\mathrm{T}}} & \boldsymbol{-uD^{-1}} \\ -D^{-1}\boldsymbol{v}^{\mathrm{T}} & D^{-1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{M^{-1}} & \boldsymbol{-M^{-1}u} \\ -\boldsymbol{v}^{\mathrm{T}}M^{-1} & \boldsymbol{1 + v^{\mathrm{T}}M^{-1}u} \end{bmatrix} \tag{6}$$

Los bloques $1,1$ dicen que $M^{-1} = I + \boldsymbol{u}D^{-1}\boldsymbol{v}^{\mathrm{T}}$. *Esto es la fórmula* (1), con $D = 1 - \boldsymbol{v}^{\mathrm{T}}\boldsymbol{u}$.

La inversa de $M = I - UV^{\mathrm{T}}$

Podemos dar un gran paso sin esfuerzo. En lugar de una perturbación $\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}$ de rango 1, supongamos que tenemos una perturbación UV^{T} de rango k. La matriz $\boldsymbol{U}$ **es de $\boldsymbol{n}$ por $\boldsymbol{k}$** y la matriz $\boldsymbol{V^{\mathrm{T}}}$ **es de $\boldsymbol{k}$ por $\boldsymbol{n}$**. Así tenemos k columnas y k filas exactamente como teníamos una columna $\boldsymbol{u}$ y una fila $\boldsymbol{v}^{\mathrm{T}}$.

La fórmula para M^{-1} se mantiene igual pero ahora tenemos I_n e I_k de dos tamaños:

$$\boxed{\textbf{La inversa de } \boldsymbol{M = I_n - UV^{\mathrm{T}}} \textbf{ es } \boldsymbol{M^{-1} = I_n + U(I_k - V^{\mathrm{T}}U)^{-1}V^{\mathrm{T}}}} \tag{7}$$

Aquí se destaca un punto importante sobre estas fórmulas inversas. **Estamos intercambiando una inversa de tamaño $\boldsymbol{n}$ con una inversa de tamaño $\boldsymbol{k}$.** Como al principio de esta sección era $k = 1$, teníamos una inversa de tamaño 1 que era una división por el número $1 - \boldsymbol{v}^{\mathrm{T}}\boldsymbol{u}$. Ahora $V^{\mathrm{T}}U$ es $(k \times n)\,(n \times k)$. Tenemos una matriz $I_k - V^{\mathrm{T}}U$ de k por k para invertir, no de n por n.

La prueba rápida para la fórmula (7) es de nuevo una verificación directa de que $MM^{-1} = I$:

$$(I_n - UV^{\mathrm{T}})(I_n + U(I_k - V^{\mathrm{T}}U)^{-1}V^{\mathrm{T}}) = I_n - UV^{\mathrm{T}} + (\boldsymbol{I_n - UV^{\mathrm{T}}})\boldsymbol{U}(I_k - V^{\mathrm{T}}U)^{-1}V^{\mathrm{T}}.$$

Reemplacemos $\boldsymbol{(I_n - UV^{\mathrm{T}})U}$ en esa ecuación por $\boldsymbol{U(I_k - V^{\mathrm{T}}U)}$. ¡Esta es una identidad estupenda! El lado derecho resulta $I_n - UV^{\mathrm{T}} + UV^{\mathrm{T}}$ que es I_n. Esto prueba la fórmula (7).

De nuevo tenemos una matriz extendida E de tamaño $n + k$ que contiene la clave:

$$\boxed{\boldsymbol{E} = \begin{bmatrix} \boldsymbol{I_n} & \boldsymbol{U} \\ \boldsymbol{V^{\mathrm{T}}} & \boldsymbol{I_k} \end{bmatrix} \textbf{ tiene determinante } = \det(\boldsymbol{I_n - UV^{\mathrm{T}}}) = \det(\boldsymbol{I_k - V^{\mathrm{T}}U}).} \tag{8}$$

Si $k << n$, el lado derecho de (7) probablemente será más sencillo y rápido que un ataque directo al lado izquierdo. **La matriz $\boldsymbol{V^{\mathrm{T}}U}$ de tamaño $\boldsymbol{k}$ es más pequeña que $\boldsymbol{UV^{\mathrm{T}}}$ de tamaño $\boldsymbol{n}$.**

Ejemplo 1 ¿Cuál es la inversa de $M = I - \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$? En este caso $\boldsymbol{u} = \boldsymbol{v} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$.

Solución Aquí $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{u} = 3$ y $M^{-1} = I + \dfrac{\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}}{1-3}$. Entonces M^{-1} es igual a $I - \dfrac{1}{2}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$.

Ejemplo 2 Si $M = I - \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \boldsymbol{I - UV^{\mathrm{T}}}$ entonces $\boldsymbol{M^{-1}} = \begin{bmatrix} 1 & 1 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$

Eso se obtuvo a partir de escribir la primera matriz mostrada como UV^{T} e intercambiarla por $V^{\mathrm{T}}U$:

$$UV^{\mathrm{T}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{y} \quad V^{\mathrm{T}}U = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

Entonces M^{-1} es $\boldsymbol{I_3 + U\left[\ I_2 - V^{\mathrm{T}}U\ \right]^{-1} V^{\mathrm{T}}} = I_3 + U \begin{bmatrix} \mathbf{1} & \mathbf{-1} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}^{-1} V^{\mathrm{T}}$.

El meollo de la cuestión es que la matriz M^{-1} de 3 por 3 se obtuvo invirtiendo la matriz 2 por 2 que está en negrita.

Perturbando una matriz invertible A

Hasta ahora hemos comenzado con la matriz identidad $I = I_n$. La modificamos para que sea $I - \boldsymbol{uv}^{\mathrm{T}}$ y luego $I - UV^{\mathrm{T}}$. Cambios de rango 1 y luego cambios de rango k. Para obtener la idea completa de Sherman-Morrison-Woodbury, vamos más allá: *comenzamos con A en lugar de I.*

Perturbamos una A invertible por una matriz UV^{T} de rango k. Ahora $M = A - UV^{\mathrm{T}}$.

$$\boxed{\begin{array}{c} \textbf{Fórmula de Sherman-Morrison-Woodbury} \\ \boldsymbol{M^{-1} = (A - UV^{\mathrm{T}})^{-1} = A^{-1} + A^{-1}U(I - V^{\mathrm{T}}A^{-1}U)^{-1}V^{\mathrm{T}}A^{-1}} \end{array}} \tag{9}$$

Hasta ahora A era I_n. La fórmula (9) final todavía se conecta con una matriz de extensión E.

Supongamos que A es invertible $\quad E = \begin{bmatrix} A & U \\ V^{\mathrm{T}} & I \end{bmatrix}$ es invertible cuando $M = A - UV^{\mathrm{T}}$ es invertible.

Para encontrar la inversa de E, podemos hacer operaciones de filas para reemplazar V^{T} con ceros:

Multiplicar la fila 1 por $V^{\mathrm{T}}A^{-1}$ y restar de la fila 2 para obtener $\begin{bmatrix} A & U \\ 0 & I - V^{\mathrm{T}}A^{-1}U \end{bmatrix}$

O podemos realizar operaciones de columna para reemplazar U con ceros:

Multiplicar la columna 1 por $A^{-1}U$ y restar de la columna 2 para obtener $\begin{bmatrix} A & 0 \\ V^{\mathrm{T}} & I - V^{\mathrm{T}}A^{-1}U \end{bmatrix}$

Como en la ecuación (6), tenemos dos maneras de invertir E. Estas dos formas de E^{-1} deben coincidir.

$$\begin{bmatrix} A^{-1} + A^{-1}UC^{-1}V^{\mathrm{T}}A^{-1} & -A^{-1}UC^{-1} \\ -C^{-1}VA^{-1} & C^{-1} \end{bmatrix} = \begin{bmatrix} M^{-1} & -M^{-1}U \\ V^{\mathrm{T}}B^{-1} & I_k + V^{\mathrm{T}}M^{-1}U \end{bmatrix} \tag{10}$$

Aquí C es $I - V^{\mathrm{T}}A^{-1}U$ y M es $A - UV^{\mathrm{T}}$. La matriz deseada es M^{-1} (la inversa de A perturbada). Comparando los bloques $(1,1)$ en la ecuación (10) obtenemos la ecuación (9).

Resumen La inversa de $M = A - UV^{\mathrm{T}}$ de n por n se obtiene de la inversa de A de n por n y la inversa de $C = I - V^{\mathrm{T}}A^{-1}U$ de k por k. **Para una prueba rápida, multiplique (9) por $\boldsymbol{A - UV^{\mathrm{T}}}$.**

Este es un buen momento para recopilar cuatro identidades matriciales estrechamente relacionadas. En todos los casos, una matriz B o A^{T} o U a izquierda reaparece a la derecha, incluso si no conmuta con A o V. Como en muchas pruebas, la *ley asociativa* se esconde a simple vista: $B(AB) = (BA)B$.

$$\boxed{\begin{aligned} B(I_m + AB) &= (I_n + BA)B \\ B(I_m + AB)^{-1} &= (I_n + BA)^{-1}B \\ A^{\mathrm{T}}(AA^{\mathrm{T}} + \lambda I_n)^{-1} &= (A^{\mathrm{T}}A + \lambda I_m)^{-1}A^{\mathrm{T}} \\ U(I_k - V^{\mathrm{T}}U) &= (I_n - UV^{\mathrm{T}})U \end{aligned}}$$

A es de m por n y B es de n por m. La segunda identidad incluye el hecho de que $I + AB$ es invertible exactamente cuando $I + BA$ es invertible. En otras palabras, -1 no es un autovalor de AB exactamente cuando -1 no es un autovalor de BA. *AB y BA tienen los mismos autovalores no nulos.*

La clave, como en la Sección I.6, es que $(I + AB)\boldsymbol{x} = \mathbf{0}$ implica $(I + BA)B\boldsymbol{x} = \mathbf{0}$.

La derivada de A^{-1}

Esta sección se enfocará en las aplicaciones de las fórmulas de inversas. Pero primero trabajaremos en el cálculo matricial. El objetivo principal del cálculo de derivadas es encontrar el cambio en la función $f(x)$ cuando x *cambia ligeramente*. Ese cambio Δx produce un cambio Δf. Así el cociente Δf sobre Δx se aproxima a la derivada df/dx.

Aquí x es una matriz A. La función es $f(A) = A^{-1}$. ¿Cómo cambia A^{-1} cuando cambia A? Hasta ahora el cambio $\boldsymbol{uv}^{\mathrm{T}}$ o UV^{T} era de rango pequeño. Ahora el cambio deseado en A será infinitesimalmente pequeño, de cualquier rango.

Comenzamos con la letra $B = A + \Delta A$, y escribimos esta fórmula matricial muy útil:

$$\boxed{\boldsymbol{B^{-1} - A^{-1} = B^{-1}\,(A - B)\,A^{-1}}} \tag{11}$$

Vemos que esta igualdad es cierta. Del lado derecho AA^{-1} es I y $B^{-1}B$ es I. De hecho, (11) podría llevarnos a las fórmulas anteriores de $(A - UV^{\mathrm{T}})^{-1}$. Se ve de inmediato que si $A - B$ tiene rango 1 (o k), entonces $B^{-1} - A^{-1}$ tiene rango 1 (o k). Se supone que las matrices A y B son invertibles, entonces la multiplicación por B^{-1} o A^{-1} no afecta el rango.

Ahora pensemos en $A = A(t)$ como una matriz que cambia con la variable tiempo t. Su derivada en cada instante de tiempo t es dA/dt. Por supuesto A^{-1} también cambia con t. **Queremos encontrar su derivada $\boldsymbol{dA^{-1}/dt}$.** Entonces dividimos esos cambios $\Delta A = B - A$ y $\Delta A^{-1} = B^{-1} - A^{-1}$ por Δt. Ahora insertamos $A + \Delta A$ como B en la ecuación (11) y hacemos $\Delta t \to 0$.

$$\boxed{\frac{\Delta A^{-1}}{\Delta t} = -(A + \Delta A)^{-1}\,\frac{\Delta A}{\Delta t}\,A^{-1} \text{ aproxima a } \boldsymbol{\frac{dA^{-1}}{dt} = -A^{-1}\frac{dA}{dt}A^{-1}}} \tag{12}$$

Para una matriz $A = t$ de 1 por 1, con $dA/dt = 1$, recuperamos la derivada de $1/t$ como $-1/t^2$.

¡ El Problema 8 enfatiza que la derivada de A^2 **no es** $2A\,dA/dt$!

Actualizando mínimos cuadrados

En la Sección II.2 se discutió la ecuación de mínimos cuadrados $A^{\mathrm{T}}A\widehat{\boldsymbol{x}} = A^{\mathrm{T}}\boldsymbol{b}$: las "ecuaciones normales" para minimizar $||\boldsymbol{b} - A\boldsymbol{x}||^2$. *Supongamos que llega una nueva ecuación.* Así A tiene una nueva fila $\boldsymbol{r}$ (1 por n) y hay una nueva medición b_{m+1} y un nuevo $\widehat{\boldsymbol{x}}$:

$$\begin{bmatrix} A^{\mathrm{T}} & \boldsymbol{r}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} A \\ \boldsymbol{r} \end{bmatrix} \widehat{\boldsymbol{x}} = \begin{bmatrix} A^{\mathrm{T}} & \boldsymbol{r}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{b} \\ b_{m+1} \end{bmatrix} \quad \text{es} \quad \left[A^{\mathrm{T}}A + \boldsymbol{r}^{\mathrm{T}}\boldsymbol{r} \right] \widehat{\boldsymbol{x}} = A^{\mathrm{T}}\boldsymbol{b} + \boldsymbol{r}^{\mathrm{T}} b_{m+1}. \tag{13}$$

La matriz en las nuevas ecuaciones normales es $A^{\mathrm{T}}A + \boldsymbol{r}^{\mathrm{T}}\boldsymbol{r}$. Esta es una corrección de rango uno de la $A^{\mathrm{T}}A$ original. Para actualizar $\widehat{\boldsymbol{x}}$ *no* queremos crear y resolver todo un nuevo conjunto de ecuaciones normales. En cambio usamos la fórmula de actualización:

$$\left[A^{\mathrm{T}}A + \boldsymbol{r}^{\mathrm{T}}\boldsymbol{r} \right]^{-1} = (A^{\mathrm{T}}A)^{-1} - c\,(A^{\mathrm{T}}A)^{-1}\,\boldsymbol{r}^{\mathrm{T}}\boldsymbol{r}\,(A^{\mathrm{T}}A)^{-1} \text{ con } c = 1/(1+\boldsymbol{r}(A^{\mathrm{T}}A)^{-1}\,\boldsymbol{r}^{\mathrm{T}}) \tag{14}$$

Para encontrar rápido a c solo necesitamos resolver la vieja ecuación $(A^{\mathrm{T}}A)\,\boldsymbol{y} = \boldsymbol{r}^{\mathrm{T}}$.

El Problema 4 producirá la solución de mínimos cuadrados $\widehat{\boldsymbol{x}}_{\text{nuevo}}$ como una actualización de $\widehat{\boldsymbol{x}}$. La misma idea se aplica cuando A tiene M filas nuevas en lugar de una. Esto es **mínimos cuadrados recursivos**.

El filtro de Kalman

Kalman notó que esta idea de actualización también se aplicaba a **mínimos cuadrados dinámicos**. Esa palabra *dinámico* significa que incluso sin datos nuevos, el vector de estado $\boldsymbol{x}$ cambia con el tiempo. Si $\boldsymbol{x}$ indica la posición de un satélite GPS, esa posición se desplazará un $\Delta\boldsymbol{x} = \boldsymbol{v}\Delta t$ ($\boldsymbol{v}$ = velocidad). Esta aproximación o una mejor será la **ecuación de estado** para $\boldsymbol{x}_{n+1}$ en el nuevo t. Luego una nueva medición b_{m+1} en el tiempo $t + \Delta t$ actualizará aún más esa posición aproximada a $\widehat{\boldsymbol{x}}_{\text{nuevo}}$. Espero que el lector pueda ver que estamos sumando **dos ecuaciones nuevas** (ecuación de estado y ecuación de medición) al sistema $A\boldsymbol{x} \approx \boldsymbol{b}$ original:

Original
Actualización de estado
Actualización de medición

$$A_{\text{nuevo}} = \begin{bmatrix} A & 0 \\ -I & I \\ 0 & \boldsymbol{r} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{\text{viejo}} \\ \boldsymbol{x}_{\text{nuevo}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{v}\Delta t \\ b_{m+1} \end{bmatrix}. \tag{15}$$

Queremos la solución por mínimos cuadrados de (15). Y hay un giro más que hace que las fórmulas del filtro de Kalman sean impresionantes (o muy complicadas). La ecuación de estado y la ecuación de medición tienen sus propias matrices de covarianza. Esas ecuaciones son inexactas (por supuesto). La varianza o covarianza V mide sus diferentes grados de fiabilidad. Las ecuaciones normales $A^{\mathrm{T}}A\widehat{\boldsymbol{x}} = A^{\mathrm{T}}\boldsymbol{b}$ deberían ser pesadas adecuadamente por V^{-1} convirtiéndose en $A^{\mathrm{T}}V^{-1}A\widehat{\boldsymbol{x}} = A^{\mathrm{T}}V^{-1}\boldsymbol{b}$. Y en verdad hay que actualizar V en cada paso.

A pesar de todo esto, Kalman persiguió el objetivo de usar fórmulas de actualización. En lugar de resolver las ecuaciones normales completas para conocer $\widehat{\boldsymbol{x}}_{\text{nuevo}}$, actualizó $\widehat{\boldsymbol{x}}_{\text{viejo}}$ en dos pasos.

La predicción $\widehat{\boldsymbol{x}}_{\text{estado}}$ viene de la ecuación de estado. Luego viene la corrección a $\widehat{\boldsymbol{x}}_{\text{nuevo}}$, con la nueva medición b_{m+1}: corrección cero.

$\boldsymbol{K}$ = **Matriz de ganancia de Kalman**

$$\widehat{\boldsymbol{x}}_{\text{nuevo}} = \widehat{\boldsymbol{x}}_{\text{estado}} + K(b_{m+1} - \boldsymbol{r}\,\widehat{\boldsymbol{x}}_{\text{estado}}) \tag{16}$$

La *matriz de ganancia* K se crea a partir de A y $\boldsymbol{r}$ y las matrices de covarianza V_{estado} y $\boldsymbol{V_b}$. Podemos ver que si la nueva b_{m+1} concuerda a la perfección con la predicción $\widehat{\boldsymbol{x}}_{\text{estado}}$, entonces la corrección en (16) de $\widehat{\boldsymbol{x}}_{\text{estado}}$ a $\widehat{\boldsymbol{x}}_{\text{nuevo}}$ es cero.

También necesitamos actualizar la covarianza de todo el sistema: medir la fiabilidad de $\widehat{x}_{\text{nuevo}}$. De hecho esta V a menudo es la información más importante. Mide la precisión de todo el sistema de sensores que produjo $\widehat{x}_{\text{final}}$.

Para la aplicación de GPS, nuestro texto con Kai Borre proporciona muchos más detalles: *Algorithms for Global Positioning* (Wellesley-Cambridge Press). El objetivo es estimar la precisión de las mediciones del GPS : altísima precisión para la medición de las placas tectónicas, menor precisión para los satélites y mucha menor precisión para la ubicación de un automóvil.

Métodos de actualización quasi-Newton

Una actualización completamente diferente ocurre en métodos de Newton aproximados para resolver $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$. Aquí hay n ecuaciones y n incógnitas $x_1, \ldots, x_n$. El método clásico de Newton usa la matriz jacobiana $\boldsymbol{J}(\boldsymbol{x})$ que contiene las primeras derivadas de cada componente de $\boldsymbol{f}$:

$$\textbf{Newton} \qquad \boldsymbol{J_{ik}} = \frac{\boldsymbol{\partial f_i}}{\boldsymbol{\partial x_k}} \quad \text{y} \quad \boldsymbol{x}_{\text{nuevo}} = \boldsymbol{x}_{\text{viejo}} - \boldsymbol{J}(\boldsymbol{x}_{\text{viejo}})^{-1}\boldsymbol{f}(\boldsymbol{x}_{\text{viejo}}). \tag{17}$$

Esto se basa en la aproximación fundamental $\boldsymbol{J\,\Delta x \approx \Delta f}$, con $\boldsymbol{\Delta f} = \boldsymbol{f}(\boldsymbol{x}_{\text{nuevo}}) - \boldsymbol{f}(\boldsymbol{x}_{\text{viejo}}) = -\boldsymbol{f}(\boldsymbol{x}_{\text{viejo}})$. Elegimos $\boldsymbol{x}_{\text{nuevo}}$ de esta manera porque nuestro plan es lograr $\boldsymbol{f}(\boldsymbol{x}_{\text{nuevo}}) \approx \boldsymbol{0}$.

La dificultad es la matriz jacobiana J. Para n grande, incluso la diferenciación automática (la clave para la *backpropagation* del Capítulo VII) será lenta. En lugar de recalcular $\boldsymbol{J}$ en cada iteración, *los métodos de Quasi-Newton usan un método de actualización* $\boldsymbol{J}(\boldsymbol{x}_{\text{nuevo}}) = \boldsymbol{J}(\boldsymbol{x}_{\text{viejo}}) + \boldsymbol{\Delta J}$.

En principio $\boldsymbol{\Delta J}$ usa las derivadas de J y por lo tanto **derivadas segundas de** $\boldsymbol{f}$. La recompensa es precisión de segundo orden y convergencia rápida del método de Newton. Pero el precio de calcular todas las derivadas segundas cuando n es grande (como en aprendizaje profundo) podría ser imposiblemente alto.

Los métodos de quasi-Newton crean una actualización de rango pequeño de $\boldsymbol{J}(\boldsymbol{x}_{\text{viejo}})$ en lugar de calcular un jacobiano nuevo completo en $\boldsymbol{x}_{\text{nuevo}}$. La actualización utiliza la nueva información que viene con el cálculo de $\boldsymbol{x}_{\text{nuevo}}$ en (17). Debido a que en el método de Newton aparece J^{-1}, la actualización tiene en cuenta su cambio de rango uno para obtener J^{-1}_{nuevo}, **sin recalcular** $\boldsymbol{J^{-1}}$. Aquí está la clave, y las derivadas de $f_1, \ldots, f_n$ no son necesarias :

$$\textbf{Condición de quasi-Newton} \qquad J_{\text{nuevo}}\,(\boldsymbol{x}_{\text{nuevo}} - \boldsymbol{x}_{\text{viejo}}) = \boldsymbol{f}_{\text{nuevo}} - \boldsymbol{f}_{\text{viejo}} \tag{18}$$

Esto es información $\boldsymbol{J\Delta x} = \boldsymbol{\Delta f}$ en la dirección en que nos hemos desplazado. Como la ecuación (17) usa J^{-1} en lugar de J, actualizamos J^{-1} para que satisfaga (18). Esto se logra usando la fórmula de Sherman-Morrison. O la "corrección BFGS", que es una matriz de rango 2 descubierta por cuatro autores al mismo tiempo. Otro enfoque es actualizar los factores LU o LDL^{T} de J_{viejo}.

A menudo las n ecuaciones originales $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ surgen de minimizar una función $F(x_1, \ldots, x_n)$. Así $\boldsymbol{f} = (\partial F/\partial x_1, \ldots, \partial F/\partial x_n)$ es el gradiente de esta función F, y $\boldsymbol{f} = \boldsymbol{0}$ en el punto mínimo. Ahora la matriz jacobiana J (primeras derivadas de $\boldsymbol{f}$) resulta una matriz hessiana H (segundas derivadas de F). Sus elementos son $H_{jk} = \partial^2 F/\partial x_j\,\partial x_k$.

Si todo sale bien, el método de Newton encuentra rápido el punto $\boldsymbol{x}^*$ donde se minimiza F y sus derivadas son $\boldsymbol{f}(\boldsymbol{x}^*) = \boldsymbol{0}$. El método de quasi-Newton que actualiza J aproximadamente en lugar de recalcular J es mucho más asequible para n grande. El costo todavía puede ser excesivo (como en los problemas de aprendizaje automático) para n extremadamente grande.

Problemas de la Sección III.1

1 Otra manera de hallar una fórmula para $(I-\boldsymbol{uv}^{\mathrm{T}})^{-1}$ se obtiene si se comienza con la fórmula de la serie geométrica: $(1-x)^{-1}=1+x+x^2+x^3+\cdots$ Aplique esa fórmula a $x=\boldsymbol{uv}^{\mathrm{T}}=$ matriz:

$$\begin{aligned}(I-\boldsymbol{uv}^{\mathrm{T}})^{-1} &= I+\boldsymbol{uv}^{\mathrm{T}}+\boldsymbol{uv}^{\mathrm{T}}\boldsymbol{uv}^{\mathrm{T}}+\boldsymbol{uv}^{\mathrm{T}}\boldsymbol{uv}^{\mathrm{T}}\boldsymbol{uv}^{\mathrm{T}}+\cdots \\ &= I+\boldsymbol{u}\,[1+\boldsymbol{v}^{\mathrm{T}}\boldsymbol{u}+\boldsymbol{v}^{\mathrm{T}}\boldsymbol{uv}^{\mathrm{T}}\boldsymbol{u}+\cdots]\,\boldsymbol{v}^{\mathrm{T}}.\end{aligned}$$

Tome $x=\boldsymbol{v}^{\mathrm{T}}\boldsymbol{u}$ para hallar $I+\dfrac{\boldsymbol{uv}^{\mathrm{T}}}{1-\boldsymbol{v}^{\mathrm{T}}\boldsymbol{u}}$. Esto es exactamente (1) para $(I-\boldsymbol{uv}^{\mathrm{T}})^{-1}$.

2 Encuentre E^{-1} de la ecuación (4) con $D=1-\boldsymbol{v}^{\mathrm{T}}\boldsymbol{u}$, y también de la ecuación (5):

De (4) $\quad E^{-1}=\begin{bmatrix} I & -\boldsymbol{u}D^{-1} \\ 0 & D^{-1}\end{bmatrix}\begin{bmatrix} I & 0 \\ -\boldsymbol{v}^{\mathrm{T}} & 1\end{bmatrix}=\begin{bmatrix} & \\ & \end{bmatrix}$

De (5) $\quad E^{-1}=\begin{bmatrix} (I-\boldsymbol{uv}^{\mathrm{T}})^{-1} & 0 \\ -\boldsymbol{v}^{\mathrm{T}}(I-\boldsymbol{uv}^{\mathrm{T}})^{-1} & 1\end{bmatrix}\begin{bmatrix} I & -\boldsymbol{u} \\ 0 & 1\end{bmatrix}=\begin{bmatrix} & \\ & \end{bmatrix}$

Compare los bloques $1,1$ para hallar $M^{-1}=(I-\boldsymbol{uv}^{\mathrm{T}})^{-1}$ en la fórmula (1).

3 La fórmula final (9) de Sherman-Morrison-Woodbury perturba A por UV^{T} (rango k). Escriba esa fórmula en el caso importante correspondiente a $k=1$:

$$M^{-1}=(A-\boldsymbol{uv}^{\mathrm{T}})^{-1}=A^{-1}+\underline{\qquad\qquad}.$$

Pruebe la fórmula con este pequeño ejemplo:

$$A=\begin{bmatrix} 3 & 0 \\ 0 & 2\end{bmatrix}\qquad \boldsymbol{u}=\begin{bmatrix} 1 \\ 0\end{bmatrix}\quad \boldsymbol{v}=\begin{bmatrix} 1 \\ 0\end{bmatrix}\qquad A-\boldsymbol{uv}^{\mathrm{T}}=\begin{bmatrix} 2 & 0 \\ 0 & 2\end{bmatrix}$$

4 En el problema 3 se encontró la matriz inversa $M^{-1}=(A-\boldsymbol{uv}^{\mathrm{T}})^{-1}$. Al resolver la ecuación $M\boldsymbol{y}=\boldsymbol{b}$, calculamos **sólo la solución** $\boldsymbol{y}$ y no toda la matriz inversa M^{-1}. Puede encontrar $\boldsymbol{y}$ en dos pasos sencillos:

Paso 1 Resolver $A\boldsymbol{x}=\boldsymbol{b}$ y $A\boldsymbol{z}=\boldsymbol{u}$. Calcular $D=1-\boldsymbol{v}^{\mathrm{T}}\boldsymbol{z}$.

Paso 2 Entonces $\boldsymbol{y}=\boldsymbol{x}+\dfrac{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{x}}{D}\boldsymbol{z}$ es la solución de $M\boldsymbol{y}=(A-\boldsymbol{uv}^{\mathrm{T}})\boldsymbol{y}=\boldsymbol{b}$.

Verifique que $(A-\boldsymbol{uv}^{\mathrm{T}})\boldsymbol{y}=\boldsymbol{b}$. **Resolvimos dos ecuaciones con A, ninguna con M.**

5 **¡Demuestre que la fórmula final (9) es correcta! Multiplique la ecuación (9) por $A-UV^{\mathrm{T}}$.**

Busque el momento en que $(A-UV^{\mathrm{T}})A^{-1}U$ se convierte en $U(I-V^{\mathrm{T}}A^{-1}U)$.

6 En el caso trivial $U=V=I_n$, ¿qué fórmula da la ecuación (9) para $(A-I)^{-1}$? ¿Puede demostrarlo directamente?

7 El problema 4 se extiene a un cambio de rango k, $M^{-1} = (A - UV^{\mathrm{T}})^{-1}$. Para resolver la ecuación $My = b$, calculamos solo la solución y y no toda la matriz inversa M^{-1}.

Paso 1 Resolver $Ax = b$ y las k ecuaciones $AZ = U$ (U y Z son de n por k)

Paso 2 Formar la matriz $C = I - V^{\mathrm{T}}Z$ y resolver $Cw = V^{\mathrm{T}}x$. El $y = M^{-1}b$ deseado es $y = x + Zw$.

Use (9) para verificar que $(A - UV^{\mathrm{T}})\,y = b$. Resolvimos $k + 1$ ecuaciones con A y multiplicamos $V^{\mathrm{T}}Z$, pero nunca usamos $M = A - UV^{\mathrm{T}}$.

8 ¿Cuál es la derivada de $(A(t))^2$? **La derivada correcta no es $2\,A(t)\,\dfrac{dA}{dt}$.** Debe calcular $(A + \Delta A)^2$ y restar A^2, luego dividir por Δt y hacer tender Δt a 0.

9 Pruebe que la fórmula (12) es cierta para la derivada de $A^{-1}(t)$ cuando

$$A(t) = \begin{bmatrix} 1 & t^2 \\ 0 & 1 \end{bmatrix} \quad \text{y} \quad A^{-1}(t) = \begin{bmatrix} 1 & -t^2 \\ 0 & 1 \end{bmatrix}.$$

10 Suponga que conoce el promedio $\widehat{x}_{\mathrm{old}}$ de $b_1, b_2, \ldots, b_{999}$. Cuando obtiene b_{1000}, verifique que el nuevo promedio sea una combinación de $\widehat{x}_{\mathrm{old}}$ y la diferencia $b_{1000} - \widehat{x}_{\mathrm{old}}$:

$$\widehat{x}_{\mathbf{new}} = \frac{b_1 + \cdots + b_{1000}}{1000} = \frac{b_1 + \cdots + b_{999}}{999} + \frac{1}{1000}\left(b_{1000} - \frac{b_1 + \cdots + b_{999}}{999}\right).$$

Esto es un "filtro de Kalman" $\widehat{x}_{\mathrm{new}} = \widehat{x}_{\mathrm{old}} + \frac{1}{1000}\,(b_{1000} - \widehat{x}_{\mathrm{old}})$ con matriz de ganancia $\frac{1}{1000}$.

11 El fitro de Kalman incluye también una *ecuación de estado $x_{k+1} = Fx_k$ con su propia varianza de error s^2*. El problemas de mínimos cuadrados dinámico permite que x "se desplace" a medida que k aumenta:

$$\begin{bmatrix} 1 & & \\ -F & 1 & \\ & 1 & \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} b_0 \\ 0 \\ b_1 \end{bmatrix} \text{ con varianzas } \begin{bmatrix} \sigma^2 \\ s^2 \\ \sigma^2 \end{bmatrix}.$$

Con $F = 1$, divida ambos lados de esas tres ecuaciones por σ, s, y σ. Encuentre $\widehat{x_0}$ y $\widehat{x_1}$ usando mínimos cuadrados, lo que da mayor peso al b_1 reciente.

El artículo de William Hager *Updating the Inverse of a Matrix* fue de gran ayuda para escribir esta sección del libro: SIAM Review **31** (1989).

III.2. Autovalores entrelazados y señales de rango bajo

La sección anterior mostró el cambio que se produce en A^{-1} a raíz de un cambio en A. Pudimos permitir cambios infinitesimales dA y también cambios finitos $\Delta A = -UV^{\mathrm{T}}$. Los resultados fueron o un cambio infinitesimal o un cambio finito en la matriz inversa :

$$\frac{dA^{-1}}{dt} = -A^{-1}\frac{dA}{dt}A^{-1} \quad \text{y} \quad \Delta A^{-1} = A^{-1}U(I - V^{\mathrm{T}}A^{-1}U)^{-1}V^{\mathrm{T}}A^{-1} \tag{1}$$

Esta sección plantea las mismas preguntas sobre los autovalores y valores singulares de A.

¿ Cómo cambia cada λ y cada σ a raíz de cambios en la matriz A ?

Veremos fórmulas lindas para $\boldsymbol{d\lambda/dt}$ y $\boldsymbol{d\sigma/dt}$. Pero no mucho es lineal respecto a los autovalores o valores singulares. El cálculo es exitoso para cambios infinitesimales $d\lambda$ y $d\sigma$, porque la derivada es un operador lineal. Pero no podemos esperar conocer los valores exactos en los saltos a $\lambda(A + \Delta A)$ o $\sigma(A + \Delta A)$. Los autovalores son más complicados que las inversas.

Pero hay buenas noticias. Lo que se puede lograr es impresionante. He aquí una muestra para una matriz simétrica S. Supongamos que S cambia a $S + \boldsymbol{uu}^{\mathrm{T}}$ (un cambio "positivo" de rango 1). Sus autovalores cambian de $\lambda_1 \geq \lambda_2 \geq \ \ldots$ a $z_1 \geq z_2 \geq \ \ldots$ Esperamos un aumento en los autovalores ya que $\boldsymbol{uu}^{\mathrm{T}}$ es semidefinida positiva. *Pero, ¿ qué tan grandes son los aumentos ?*

Cada autovalor $\boldsymbol{z_i}$ de $\boldsymbol{S + uu^{\mathrm{T}}}$ no es menor que $\boldsymbol{\lambda_i}$ ni mayor que $\boldsymbol{\lambda_{i-1}}$. Entonces los $\boldsymbol{\lambda}$ y los $\boldsymbol{z}$ están "entrelazados". Cada $\boldsymbol{z_2, \ldots, z_n}$ se encuentra entre dos $\boldsymbol{\lambda}$:

$$\boldsymbol{z_1 \geq \lambda_1 \geq z_2 \geq \lambda_2 \geq \ \ldots \ \geq z_n \geq \lambda_n.} \tag{2}$$

Tenemos cotas superiores de los cambios en los autovalores aunque no tengamos fórmulas para $\Delta\lambda$. Hay un punto a tener en cuenta para que no se malinterprete. Supongamos que el cambio $\boldsymbol{uu}^{\mathrm{T}}$ en la matriz es $C\boldsymbol{q}_2\boldsymbol{q}_2^{\mathrm{T}}$ (donde $\boldsymbol{q}_2$ es el segundo autovector unitario de S). Entonces $S\boldsymbol{q}_2 = \lambda_2\boldsymbol{q}_2$ verá un salto en ese autovalor a $\lambda_2 + C$, porque $(S + C\boldsymbol{q}_2\boldsymbol{q}_2^{\mathrm{T}})\boldsymbol{q}_2 = (\lambda_2 + C)\boldsymbol{q}_2$. Ese salto será grande si C es grande. *¿ Cómo podría el segundo autovalor de $S + \boldsymbol{uu}^{\mathrm{T}}$ cumplir $z_2 = \lambda_2 + C \leq \lambda_1$?*

Respuesta : si C es un número grande, entonces $\lambda_2 + C$ no es el segundo autovalor de $S + \boldsymbol{uu}^{\mathrm{T}}$. El **mayor autovalor** de la nueva matriz $S + C\boldsymbol{q}_2\boldsymbol{q}_2^{\mathrm{T}}$ será z_1 (y su autovector es $\boldsymbol{q}_2$). El autovalor original principal λ_1 de S ahora es el segundo autovalor z_2 de la nueva matriz. Así la afirmación (2) de que $z_2 \leq \lambda_1 \leq z_1$ es una afirmación completamente verdadera (en este ejemplo) ya que $z_2 = \lambda_1$ es más chico que $z_1 = \lambda_2 + C$.

Conectaremos este entrelazamiento con el hecho de que los autovectores entre $\lambda_1 = \lambda_{\text{máx}}$ y $\lambda_n = \lambda_{\text{mín}}$ son todos **puntos de ensilladuras del cociente $\boldsymbol{R(x) = x^{\mathrm{T}}Sx/x^{\mathrm{T}}x}$.**

La derivada de un autovalor

Tenemos una matriz $A(t)$ que cambia con el tiempo t. Entonces sus autovalores $\lambda(t)$ también cambian. Supondremos que *ningún autovalor de* $A(0)$ *se repite*, así cada autovalor $\lambda(0)$ de $A(0)$ puede seguirse por lo menos por un tiempo corto t, a medida que $\lambda(0)$ cambia a un autovalor $\lambda(t)$ de $A(t)$. **¿Cuál es su derivada $d\lambda/dt$?**

La clave para encontrar $d\lambda/dt$ es reunir la información que tenemos. La primera es $A(t)\boldsymbol{x}(t) = \lambda(t)\boldsymbol{x}(t)$. La segunda es que la matriz traspuesta $A^{\mathrm{T}}(t)$ también tiene el autovalor $\lambda(t)$, porque $\det(A^{\mathrm{T}} - \lambda I) = \det(A - \lambda I)$. Probablemente A^{T} tenga un autovector distinto $\boldsymbol{y}(t)$. Si $\boldsymbol{x}$ es la columna k de la matriz de autovectores X para A, e $\boldsymbol{y}$ es la columna k de la matriz de autovectores $(X^{-1})^{\mathrm{T}}$ para A^{T}. (La razón es que $A = X^{-1}\Lambda X$ lleva a $A^{\mathrm{T}} = X^{\mathrm{T}}\Lambda(X^{-1})^{\mathrm{T}}$). Las longitudes de $\boldsymbol{x}$ e $\boldsymbol{y}$ son normalizadas a $X^{-1}X = I$. *Esto requiere de* $\boldsymbol{y}^{\mathrm{T}}(t)\boldsymbol{x}(t) = \mathbf{1}$ *para todo* t.

En el siguiente cuadro mostramos resultados conocidos con la fórmula deseada para $d\lambda/dt$

Sabemos	$\boldsymbol{A}(t)\,\boldsymbol{x}(t) = \boldsymbol{\lambda}(t)\,\boldsymbol{x}(t) \quad \boldsymbol{y}^{\mathrm{T}}(t)\boldsymbol{A}(t) = \boldsymbol{\lambda}(t)\boldsymbol{y}^{\mathrm{T}}(t) \quad \boldsymbol{y}^{\mathrm{T}}(t)\,\boldsymbol{x}(t) = \mathbf{1}$	(3)
Fórmulas	$\boldsymbol{\lambda}(t) = \boldsymbol{y}^{\mathrm{T}}(t)\boldsymbol{A}(t)\,\boldsymbol{x}(t) \qquad \text{y} \qquad \dfrac{\boldsymbol{d\lambda}}{\boldsymbol{dt}} = \boldsymbol{y}^{\mathrm{T}}(t)\dfrac{\boldsymbol{dA}}{\boldsymbol{dt}}\,\boldsymbol{x}(t)$	(4)

Para encontrar esa fórmula $\lambda = \boldsymbol{y}^{\mathrm{T}}A\boldsymbol{x}$, solo hay que multiplicar la primera afirmación $A\boldsymbol{x} = \lambda\boldsymbol{x}$ por $\boldsymbol{y}^{\mathrm{T}}$ y usar $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x} = 1$. O multiplicar el segundo resultado $\boldsymbol{y}^{\mathrm{T}}A = \lambda\boldsymbol{y}^{\mathrm{T}}$ a la derecha por $\boldsymbol{x}$.

Ahora tomemos la derivada de $\boldsymbol{\lambda = y^{\mathrm{T}}A\,x}$. La regla del producto da tres términos en $d\lambda/dt$:

$$\boxed{\frac{d\lambda}{dt}} = \frac{d\boldsymbol{y}^{\mathrm{T}}}{dt}A\,\boldsymbol{x} + \boxed{\boldsymbol{y}^{\mathrm{T}}\,\frac{\boldsymbol{dA}}{\boldsymbol{dt}}\,\boldsymbol{x}} + \boldsymbol{y}^{\mathrm{T}}A\,\frac{d\boldsymbol{x}}{dt} \tag{5}$$

El término del medio es la derivada correcta $d\lambda/dt$. El primer y tercer término suman cero:

$$\frac{d\boldsymbol{y}^{\mathrm{T}}}{dt}A\,\boldsymbol{x} + \boldsymbol{y}^{\mathrm{T}}A\,\frac{d\boldsymbol{x}}{dt} = \lambda\left(\frac{d\boldsymbol{y}^{\mathrm{T}}}{dt}\,\boldsymbol{x} + \boldsymbol{y}^{\mathrm{T}}\,\frac{d\boldsymbol{x}}{dt}\right) = \lambda\,\frac{d}{dt}\left(\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}\right) = \lambda\,\frac{d}{dt}(1) = \mathbf{0}. \tag{6}$$

También existen fórmulas para $d^2\lambda/dt^2$ y $d\boldsymbol{x}/dt$ (pero son más complicadas).

Ejemplo $A = \begin{bmatrix} 2t & 1 \\ 2t & 2 \end{bmatrix}$ tiene $\lambda^2 - 2(1+t)\lambda + 2t = 0$ y $\boldsymbol{\lambda = 1 + t \pm \sqrt{1 + t^2}}$.

En $t = 0$, $\lambda_1 = \mathbf{2}$ y $\lambda_2 = \mathbf{0}$ y las derivadas de λ_1 y λ_2 son $1 \pm t(1+t^2)^{-1/2} = \mathbf{1}$.

Los autovectores correspondientes a $\lambda_1 = 2$ en $t = 0$ son $\boldsymbol{y}_1^{\mathrm{T}} = \begin{bmatrix} 0 & 1 \end{bmatrix}$ y $\boldsymbol{x}_1 = \begin{bmatrix} 1/2 \\ 1 \end{bmatrix}$.

Los autovectores correspondientes a $\lambda_2 = 0$ en $t = 0$ son $\boldsymbol{y}_2^{\mathrm{T}} = \begin{bmatrix} 1 & -\frac{1}{2} \end{bmatrix}$ y $\boldsymbol{x}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

Ahora la ecuación (5) confirma que $\dfrac{\boldsymbol{d\lambda_1}}{\boldsymbol{dt}} = \boldsymbol{y}_1^{\mathrm{T}}\dfrac{\boldsymbol{dA}}{\boldsymbol{dt}}\boldsymbol{x}_1 = \begin{bmatrix} \mathbf{0} & \mathbf{1} \end{bmatrix}\begin{bmatrix} \mathbf{2} & \mathbf{0} \\ \mathbf{2} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{1/2} \\ \mathbf{1} \end{bmatrix} = \mathbf{1}$.

La derivada de un valor singular

Una fórmula similar para $d\sigma/dt$ (derivada de un $\sigma(t)$ sin repetir) se obtiene de $A\boldsymbol{v} = \sigma\boldsymbol{u}$:

$$\boldsymbol{U^{\mathrm{T}}AV = \Sigma} \qquad \boldsymbol{u}^{\mathrm{T}}(t)\, A(t)\, \boldsymbol{v}(t) = \boldsymbol{u}^{\mathrm{T}}(t)\, \sigma(t)\, \boldsymbol{u}(t) = \sigma(t). \tag{7}$$

La derivada del lado izquierdo tiene, como en (5), tres términos por la regla del producto. El primer y el tercer término son cero porque $A\boldsymbol{v} = \sigma\boldsymbol{u}$ y $A^{\mathrm{T}}\boldsymbol{u} = \sigma\boldsymbol{v}$ y $\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u} = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{v} = 1$. Las derivadas de $\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}$ y $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}$ son cero, y entonces

$$\frac{d\boldsymbol{u}^{\mathrm{T}}}{dt}\, A(t)\, \boldsymbol{v}(t) = \sigma(t)\, \frac{d\boldsymbol{u}^{\mathrm{T}}}{dt}\, \boldsymbol{u}(t) = \boldsymbol{0} \quad \text{y} \quad \boldsymbol{u}^{\mathrm{T}}(t)\, A(t) \frac{d\boldsymbol{v}}{dt} = \sigma(t)\, \boldsymbol{v}^{\mathrm{T}}(t)\, \frac{d\boldsymbol{v}}{dt} = \boldsymbol{0}. \tag{8}$$

El segundo término en la regla del producto para $\boldsymbol{u}^{\mathrm{T}}A\boldsymbol{v}$ da una fórmula para $d\sigma/dt$:

Derivada de un valor singular

$$\boxed{\boldsymbol{u}^{\mathrm{T}}(t)\, \frac{\boldsymbol{dA}}{\boldsymbol{dt}}\, \boldsymbol{v}(t) = \frac{\boldsymbol{d\sigma}}{\boldsymbol{dt}}} \tag{9}$$

Cuando $A(t)$ es simétrica y definida positiva, $\boldsymbol{\sigma}(t) = \boldsymbol{\lambda}(t)$ y $\boldsymbol{u} = \boldsymbol{v} = \boldsymbol{y} = \boldsymbol{x}$ en (4) y (9).

Nota Las derivadas primeras de los *autovectores* van con derivadas segundas de autovalores (esto no es tan fácil). La cota de Davis-Kahan para el ángulo θ entre los autovectores unitarios de S y $S + T$ es $\operatorname{sen}\theta \leq ||T||/d$ (d es la menor distancia desde el autovalor de $S + T$ a todos los demás autovalores de S). Cotas más ajustadas que usan la estructura de S y T son muy valiosas en las aplicaciones a gradiente estocástico descendentes (vea Eldridge, Belkin y Wang).

C. Davis y W. M. Kahan, *Some new bounds on perturbation of subspaces*, Bull. Amer. Math. Soc. **75** (1969) 863 – 868.

J. Eldridge, M. Belkin, y Y. Wang, *Unperturbed: Spectral analysis beyond Davis-Kahan*, arXiv : 1706.06516v1, 20 junio 2017.

Una explicación gráfica del entrelazamiento

Esta página le debe todo al profesor Raj Rao Nadakuditi de la Universidad de Michigan. En sus visitas al MIT, en el curso 18.065, explicó la teoría y sus aplicaciones. Su software **OptShrink** para encontrar señales de rango pequeño se describe en *IEEE Transactions on Information Theory* **60** (mayo 2014) 3002 – 3018.

Cuando se suma una matriz de rango pequeña $\theta\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$ a una matriz simétrica de rango completo S **¿ Cómo cambian los $\boldsymbol{\lambda}$** ? Pensamos a S como ruido y a $\theta\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$ como la señal de rango uno. ¿ Cómo se ven afectados los autovalores de S al sumar esta señal ?

Quiero dejar en claro que *todos* los autovalores de S pueden cambiar al sumar $\theta\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$, no solo uno o dos de ellos. Pero veremos que solo uno o dos sufren cambios de orden θ. Esto los hace fáciles de encontrar. Veremos el efecto en los λ, si nuestros vectores representan videos y $\theta\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$ representa una luz encendida o apagada durante la filmación (una señal de rango uno).

Comencemos con un autovalor z y su autovector $\boldsymbol{v}$ de la nueva matriz $S + \theta\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$:

$$(S + \theta\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}})\boldsymbol{v} = z\boldsymbol{v}. \tag{10}$$

Reescribamos esa ecuación como

$$(zI - S)\boldsymbol{v} = \theta\boldsymbol{u}(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}) \quad \text{o} \quad \boldsymbol{v} = (zI - S)^{-1}\theta\boldsymbol{u}(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}). \tag{11}$$

Multipliquemos por $\boldsymbol{u}^{\mathrm{T}}$ y cancelemos el factor común $\boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}$. Esto elimina $\boldsymbol{v}$. Luego dividamos por θ. Esto conecta el nuevo autovalor z al cambio $\theta\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$ en la matriz simétrica S.

$$\boxed{\frac{1}{\theta} = \boldsymbol{u}^{\mathrm{T}}(zI - S)^{-1}\boldsymbol{u}.} \tag{12}$$

Para entender esta ecuación usemos los autovalores y autovectores de S. Si $S\boldsymbol{q}_k = \lambda_k \boldsymbol{q}_k$, entonces $(zI - S)\boldsymbol{q}_k = (z - \lambda_k)\boldsymbol{q}_k$ y $(zI - S)^{-1}\boldsymbol{q}_k = \boldsymbol{q}_k/(z - \lambda_k)$:

$$\boldsymbol{u} = \sum c_k \boldsymbol{q}_k \quad \text{implica} \quad (zI - S)^{-1}\boldsymbol{u} = \sum c_k (zI - S)^{-1}\boldsymbol{q}_k = \sum \frac{c_k \boldsymbol{q}_k}{z - \lambda_k}. \tag{13}$$

Por último, la ecuación (12) multiplica $(zI - S)^{-1}\boldsymbol{u}$ por $\boldsymbol{u}^{\mathrm{T}} = \sum c_k \boldsymbol{q}_k^{\mathrm{T}}$. **El resultado es $\boldsymbol{1/\theta}$.** Recordemos que los $\boldsymbol{q}$ son vectores unitarios ortogonales:

Ecuación secular

$$\boxed{\frac{\boldsymbol{1}}{\boldsymbol{\theta}} = \boldsymbol{u}^{\mathrm{T}}(zI - S)^{-1}\boldsymbol{u} = \sum_{\boldsymbol{k=1}}^{\boldsymbol{n}} \frac{\boldsymbol{c_k^2}}{\boldsymbol{z - \lambda_k}}} \tag{14}$$

Podemos graficar el lado izquierdo y el derecho. El lado izquierdo es constante, el lado derecho explota en cada autovalor $z = \lambda_k$ de S. Ambos lados son iguales en los n puntos $z_1, \ldots, z_n$ donde la recta horizontal $1/\theta$ toca las **curvas** de gran pendiente. **Esos $\boldsymbol{z}$ son los $\boldsymbol{n}$ autovalores de $\boldsymbol{S + \theta u u^{\mathrm{T}}}$. El gráfico muestra que cada $\boldsymbol{z_i}$ está por encima de $\boldsymbol{\lambda_i}$ y por debajo de $\boldsymbol{\lambda_{i-1}}$: entrelazados.**

El autovalor mayor z_1 es más probable que esté por encima de λ_1. Los z crecen a medida que θ crece, porque la recta $1/\theta$ se desplaza hacia abajo.

Sin duda los z dependen del vector $\boldsymbol{u}$ en la señal (como así también de θ). Si $\boldsymbol{u}$ fuera también un autovector de S, entonces su autovalor λ_k crecerá exactamente θ. Todos los demás autovectores se mantendrán iguales. Es mucho más probable que cada autovalor λ_k aumente un poco hacia z_k. **Lo principal que se ve en este gráfico es que $\boldsymbol{z_k}$ no es mayor que $\boldsymbol{\lambda_{k-1}}$.**

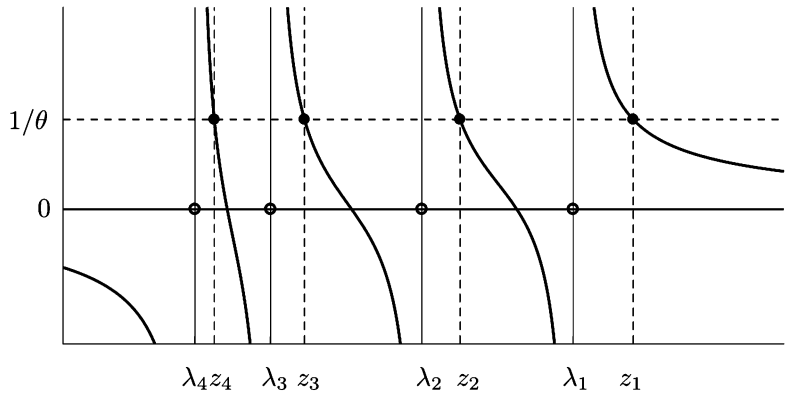


Figura III.1: autovalores z_i de $S + \theta\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$ donde la recta $\dfrac{1}{\theta}$ toca las curvas en (14).

Raj Rao Nadakuditi, *When are the most informative components for inference also the principal components* ? arXiv : 1302.1232, 5 de febrero de 2013.

El mayor autovalor de $S + T$

El mayor autovalor de una matriz simétrica S es el valor máximo de $\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}/\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$. Esta afirmación también se aplica a T (también simétrica). De inmediato deducimos lo siguiente acerca del mayor autovalor de $S + T$.

$$\boxed{\boldsymbol{\lambda}_{\max}(\boldsymbol{S}+\boldsymbol{T}) \leq \boldsymbol{\lambda}_{\max}(\boldsymbol{S}) + \boldsymbol{\lambda}_{\max}(\boldsymbol{T})} \tag{15}$$

El lado izquierdo es el valor máximo de $\dfrac{\boldsymbol{x}^{\mathrm{T}}(S+T)\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}$. Ese máximo se alcanza en un autovector $\boldsymbol{v}$ de $S + T$:

$$\boldsymbol{\lambda}_{\boldsymbol{S}+\boldsymbol{T}} = \frac{\boldsymbol{v}^{\mathrm{T}}(S+T)\boldsymbol{v}}{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}} = \frac{\boldsymbol{v}^{\mathrm{T}}S\boldsymbol{v}}{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}} + \frac{\boldsymbol{v}^{\mathrm{T}}T\boldsymbol{v}}{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}} \leq \max\frac{\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}} + \max\frac{\boldsymbol{x}^{\mathrm{T}}T\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}} = \boldsymbol{\lambda}_{\boldsymbol{S}} + \boldsymbol{\lambda}_{\boldsymbol{T}}.$$

El autovector $\boldsymbol{v}$ of $S + T$ maximiza ese primer cociente. Pero probablemente no maximice los últimos dos. Por lo tanto $\lambda_S + \lambda_T$ solo puede ser más grande que λ_{S+T}.

Esto muestra que disponemos de un principio del máximo, y también un principio del mínimo para el *menor autovalor*. Allí esperamos que $\lambda_{\text{mín}}(S+T) \geq \lambda_{\text{mín}}(S)+\lambda_{\text{mín}}(T)$. Esto se demuestra con el mismo razonamiento utilizando la propiedad del mínimo: la suma de los mínimos para S y T resulta menor que el mínimo para $S + T$. O aplicando el principio del máximo a $-S$ y $-T$.

Las dificultades surgen para los autovalores intermedios λ_2 a λ_{n-1}. Sus autovectores son **puntos de ensilladura** de la función $R(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}/\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$. Las derivadas de $R(\boldsymbol{x})$ son todas cero en los autovectores $\boldsymbol{q}_2$ a $\boldsymbol{q}_{n-1}$. Pero la matriz de las segundas derivadas de R es indefinida (autovalores positivos y negativos) en estos puntos de ensilladura. Eso hace que los autovalores sean difíciles de estimar y de calcular.

Ahora nos enfocamos en los puntos de ensilladura, que aparecen frecuentemente en los algoritmos de aprendizaje profundo. Necesitamos adquirir experiencia sobre ellos en el problema básico de autovalores. Los puntos de ensilladura también aparecen cuando hay restricciones. En lo posible queremos conectarlos a los *máximos de los mínimos* o a los *mínimos de los máximos*.

Esas ideas llevan a la mejor cota para cada autovalor de $S + T$:

Cotas superiores de Weyl $\qquad \boldsymbol{\lambda}_{\boldsymbol{i}+\boldsymbol{j}-\boldsymbol{1}}(\boldsymbol{S}+\boldsymbol{T}) \leq \boldsymbol{\lambda}_{\boldsymbol{i}}(\boldsymbol{S})+\boldsymbol{\lambda}_{\boldsymbol{j}}(\boldsymbol{T}).$ (16)

Puntos de ensilladura a partir de multiplicadores de Lagrange

En comparación con los puntos de ensilladura, calcular un máximo o mínimo de $F(\boldsymbol{x})$ es bastante sencillo. Cuando tenemos una solución aproximada $\widehat{\boldsymbol{x}}$, sabemos que $F(\widehat{\boldsymbol{x}})$ no está por debajo del mínimo de F ni por arriba del máximo: *por definición*. Pero no sabemos si $F(\widehat{\boldsymbol{x}})$ está *por arriba o por debajo* de un valor de punto de ensilladura. Similarmente, la matriz $H(\boldsymbol{x})$ de segundas derivadas de F es definida positiva (o semidefinida) en un mínimo y definida negativa en un máximo.

La matriz de segundas derivadas H en un punto de ensilladura es simétrica pero indefinida. H tiene autovalores positivos y negativos, lo que hace más difíciles a los puntos de ensilladura. El método de gradiente conjugado no suele ser adecuado para hallar puntos de ensilladura de $\boldsymbol{x}^{\mathrm{T}} H\boldsymbol{x}$.

Lagrange es responsable por muchos de los problemas de punto de ensilladura. Comenzamos por minimizar una energía definida positiva $\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}$, pero tenemos m restricciones, $A\boldsymbol{x} = \boldsymbol{b}$, en la solución. Para construir la función lagrangiana, multiplicamos esas restricciones por incógnitas nuevas $\lambda_1, \ldots, \lambda_m$ (los multiplicadores de Lagrange) :

$$\textbf{Lagrangiano} \qquad \boldsymbol{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x} + \boldsymbol{\lambda}^{\mathrm{T}}(A\boldsymbol{x} - \boldsymbol{b}).$$

Las $m + n$ ecuaciones $\partial L/\partial \boldsymbol{x} = \boldsymbol{0}$ y $\partial L/\partial \boldsymbol{\lambda} = \boldsymbol{0}$ producen un matriz indefinida en bloque:

$$\begin{bmatrix} \partial L/\partial \boldsymbol{x} \\ \partial L/\partial \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} S\boldsymbol{x} + A^{\mathrm{T}}\boldsymbol{\lambda} \\ A\boldsymbol{x} - \boldsymbol{b} \end{bmatrix} \quad \text{y} \quad H\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{S} & \boldsymbol{A^{\mathrm{T}}} \\ \boldsymbol{A} & \boldsymbol{0} \end{bmatrix}\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{b} \end{bmatrix} \tag{17}$$

Un pequeño ejemplo sería $H = \begin{bmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} \end{bmatrix}$ con determinante negativo -1 : sus autovalores tienen signos opuestos. Los problemas al final de la sección confirman que esta **"matriz KKT"** en la ecuación (17) es *indefinida*. La solución $(\boldsymbol{x}, \boldsymbol{\lambda})$ es un punto de ensilladura de la función lagrangiana L.

Puntos de ensilladura a partir de cocientes de Rayleigh

El máximo y el mínimo del cociente de Rayleigh $R(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}/\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$ son λ_1 y λ_n :

$$\textbf{Máximo } \frac{\boldsymbol{q}_1^{\mathrm{T}}S\boldsymbol{q}_1}{\boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{q}_1} = \boldsymbol{q}_1^{\mathrm{T}}\lambda_1\boldsymbol{q}_1 = \boldsymbol{\lambda_1} \qquad \textbf{Mínimo } \frac{\boldsymbol{q}_n^{\mathrm{T}}S\boldsymbol{q}_n}{\boldsymbol{q}_n^{\mathrm{T}}\boldsymbol{q}_n} = \boldsymbol{q}_n^{\mathrm{T}}\lambda_n\boldsymbol{q}_n = \boldsymbol{\lambda_n}$$

Nuestra pregunta es sobre los puntos de ensilladura: los otros puntos donde todas las derivadas del cociente $R(\boldsymbol{x})$ son cero. Confirmamos que **esos puntos de ensilladura ocurren en los otros autovectores $\boldsymbol{q_2}$ a $\boldsymbol{q_{n-1}}$ de $\boldsymbol{S}$. Nuestro objetivo es ver a los valores $\boldsymbol{\lambda_2}$ a $\boldsymbol{\lambda_{n-1}}$ como máximos de mínimos**. Esa perspectiva max-min es la clave del entrelazamiento.

Observe que los vectores $\boldsymbol{x}$ y $2\boldsymbol{x}$ y $c\boldsymbol{x}$ $(c \neq 0)$ todos producen el mismo cociente R:

$$R(2\boldsymbol{x}) = \frac{(2\boldsymbol{x})^{\mathrm{T}}S(2\boldsymbol{x})}{(2\boldsymbol{x})^{\mathrm{T}}(2\boldsymbol{x})} = \frac{4\,\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}}{4\,\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}} = \frac{\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}} = R(\boldsymbol{x}).$$

Así solo necesitamos considerar vectores unitarios con $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = 1$. Eso puede escribirse como una restricción :

$$\text{máx } \frac{\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}} \quad \text{es lo mismo que} \quad \text{máx } \boldsymbol{x^{\mathrm{T}}Sx} \quad \textbf{sujeto a } \boldsymbol{x^{\mathrm{T}}x = 1}. \tag{18}$$

¡ Un multiplicador de Lagrange puede lidiar con la restricción $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = 1$!

$$\textbf{Lagrangiano} \qquad \boldsymbol{L}(\boldsymbol{x}, \lambda) = \boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x} - \lambda(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} - 1). \tag{19}$$

Los puntos de ensilladura max-min tendrán $\partial L/\partial \boldsymbol{x} = 0$ y $\partial L/\partial \lambda = 0$ (como en la Sección I.9) :

$$\frac{\partial L}{\partial \boldsymbol{x}} = 2S\boldsymbol{x} - 2\lambda\boldsymbol{x} = \boldsymbol{0} \quad \text{y} \quad \frac{\partial L}{\partial \lambda} = 1 - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = 0. \tag{20}$$

Esto dice que el vector unitario $\boldsymbol{x}$ es un autovector con $S\boldsymbol{x} = \lambda\boldsymbol{x}$.

Ejemplo **Supongamos que S = matriz diagonal con elementos 5, 3, 1.** Para $\boldsymbol{x} = (u, v, w)$:

$$R = \frac{\boldsymbol{x}^{\mathrm{T}} S \boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}} = \frac{\boldsymbol{5u^2 + 3v^2 + w^2}}{\boldsymbol{u^2 + v^2 + w^2}} \text{ tiene un } \begin{array}{l} \textbf{valor máximo 5 en } \boldsymbol{x} = (1,0,0) \\ \textbf{valor mínimo 1 en } \boldsymbol{x} = (0,0,1) \\ \textbf{valor de punto de ensilladura 3 en } \boldsymbol{x} = (0,1,0) \end{array}$$

Al mirar a R podemos ver su máximo 5 y su mínimo 1. Todas las derivadas parciales de $R(u, v, w)$ son cero en estos tres puntos $(1,0,0), (0,0,1), (0,1,0)$. Estos son los autovectores de la matriz diagonal S. $R(\boldsymbol{x})$ es igual a los autovalores $5, 1, 3$ en esos tres puntos.

Máximos y mínimos sobre subespacios

Todos los autovectores intermedios $\boldsymbol{q}_2, \ldots, \boldsymbol{q}_{n-1}$ de S son puntos de ensilladura de $\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}/\boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}$. Este cociente es igual a $\lambda_2, \ldots, \lambda_{n-1}$ en esos autovectores. Todos los vectores singulares intermedios $\boldsymbol{v}_2, \ldots, \boldsymbol{v}_{n-1}$ son puntos de ensilladura del cociente creciente $||A\boldsymbol{x}||/||\boldsymbol{x}||$. El cociente es igual a $\sigma_2, \ldots, \sigma_{n-1}$ en esos vectores singulares. Esas afirmaciones están directamente conectadas por el hecho de que $\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x} = \boldsymbol{x}^{\mathrm{T}} A^{\mathrm{T}} A\boldsymbol{x} = ||A\boldsymbol{x}||^2$.

Pero los puntos de ensilladura son más difíciles de estudiar que los máximos o mínimos. Cuando uno se aleja de una ensilladura, el valor de la función cambia en ambos sentidos: hacia arriba y hacia abajo. En un máximo el único movimiento es hacia abajo. En un mínimo el único movimiento es hacia arriba. Entonces la mejor manera de estudiar los puntos de ensilladura es capturarlos en un principio "max-min" o "min-max".

Max-min para $\boldsymbol{\lambda_2}$

$$\boxed{\boldsymbol{\lambda_2} = \max_{\textbf{todos los espacios } Y \textbf{ bidimensionales}} \; \min_{\boldsymbol{x} \textbf{ está en } Y} \frac{\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}}} \tag{21}$$

En el ejemplo $5, 3, 1$, una opción del subespacio Y bidimensional es todos los vectores $\boldsymbol{x} = (\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{0})$. Esos vectores son combinaciones de $\boldsymbol{q}_1$ y $\boldsymbol{q}_2$. Dentro de este Y, el cociente mínimo $\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}/\boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}$ sin duda será $\lambda_2 = 3$. Ese mínimo está en $\boldsymbol{x} = \boldsymbol{q}_2 = (0,1,0)$ (entendemos los mínimos).

Punto clave: cada espacio bidimensional Y debe cruzarse con el espacio 2D de todos los vectores $(0, v, w)$. Esos espacios 2D en $\mathbf{R}^3$ seguro se cruzarán porque $2 + 2 > 3$. Para cualquier $\boldsymbol{x} = (0, v, w)$, sabemos que $\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}/\boldsymbol{x}^{\mathrm{T}} \boldsymbol{x} \leq \boldsymbol{\lambda_2}$. Así para cada Y el mínimo en (21) es $\leq \lambda_2$.

Conclusión: el máximo de los posibles mínimos es $\boldsymbol{\lambda_2}$ en (21) y $\boldsymbol{\lambda_i}$ en (22).

$$\boxed{\boldsymbol{\lambda_i(S)} = \max_{\dim \boldsymbol{V} = \boldsymbol{i}} \; \min_{\boldsymbol{x} \textbf{ en } \boldsymbol{V}} \frac{\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}} \qquad \boldsymbol{\sigma_i(A)} = \max_{\dim \boldsymbol{W} = \boldsymbol{i}} \; \min_{\boldsymbol{x} \textbf{ en } \boldsymbol{W}} \frac{||A\boldsymbol{x}||}{||\boldsymbol{x}||}} \tag{22}$$

Para $i = 1$ los espacios V y W son rectas unidimensionales. La recta V de dirección $\boldsymbol{x} = \boldsymbol{q}_1$ (primer autovector) hace que $\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}/\boldsymbol{x}^{\mathrm{T}} \boldsymbol{x} = \lambda_1$ sea un máximo. La recta W de dirección $\boldsymbol{x} = \boldsymbol{v}_1$ (primer vector singular) hace que $||A\boldsymbol{x}||/||\boldsymbol{x}|| = \sigma_1$ sea un máximo.

Para $i = 2$ los espacios V y W son planos bidimensionales. El V maximizante contiene los autovectores $\boldsymbol{q}_1, \boldsymbol{q}_2$ y el W maximizante contiene los vectores singulares $\boldsymbol{v}_1, \boldsymbol{v}_2$. El mínimo en ese V es λ_2, el mínimo en ese W es σ_2. Este patrón continúa para todo i. Esto produce los *principios max-min* de Courant-Fischer en la ecuación (22).

Entrelazamientos y las desigualdades de Weyl

Para cualquier par de matrices simétricas S y T, Weyl encontró cotas para los autovalores de $S+T$.

Desigualdades de Weyl $\boldsymbol{\lambda_{i+j-1}(S+T) \le \lambda_i(S) + \lambda_j(T)}$ (23)

$$\boldsymbol{\lambda_k(S) + \lambda_n(T) \le \lambda_k(S+T) \le \lambda_k(S) + \lambda_1(T)} \quad (24)$$

La ecuación (23) también prueba el entrelazamiento de los z que vimos en la Figura III.1. La matriz de rango uno T es $\theta \boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$ y su mayor autovalor es $\lambda_1(T) = \theta$. Todos los demás autovalores $\lambda_j(T)$ son cero. Así, para cada $j = 2, 3, \ldots$ la desigualdad de Weyl da $\boldsymbol{\lambda_{i+1}(S+T) \le \lambda_i(S)}$. Cada autovalor z_{i+1} de $S+T$ no puede ser más grande que el siguiente autovalor λ_i de S. Y para $j = 1$ tenemos $\boldsymbol{\lambda_1(S+T) \le \lambda_1(S) + \theta}$: una cota superior para el mayor autovalor de la señal con ruido.

He aquí un hemoso teorema de entrelazamiento para autovalores cuando se quitan la última columna y fila de una matriz simétrica S. Eso nos deja una matriz $\boldsymbol{S_{n-1}}$ de tamaño $n-1$.

Los $\boldsymbol{n-1}$ autovalores $\boldsymbol{\alpha_i}$ de la matriz $\boldsymbol{S_{n-1}}$ entrelazan los $\boldsymbol{n}$ autovalores de $\boldsymbol{S}$.

La idea de la prueba es que quitar la última fila y columna es lo mismo que forzar que todos los vectores sean ortogonales a $(0, \ldots, 0, 1)$. Así el mínimo en (22) podría desplazarse por debajo de λ_i. Pero α_i no se desplazará por debajo de λ_{i+1}, porque λ_{i+1} permite una opción libre con $\dim V = i+1$.

Ejemplo

$\boldsymbol{\lambda_i \ge \alpha_i \ge \lambda_{i+1}}$

$$\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \qquad \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \qquad \begin{bmatrix} 2 \end{bmatrix}$$

$$\boldsymbol{\lambda = 3,\ 3,\ 0} \qquad \boldsymbol{\alpha = 3,\ 1} \qquad \boldsymbol{3 > 2 > 1}$$

Entrelazamiento de valores singulares

Supongamos que A no es cuadrada y simétrica, entonces están involucrados sus valores singulares. Cada columna de A representa un cuadro en un vídeo. Queremos identificar una señal de rango uno $\beta \boldsymbol{x}\boldsymbol{y}^{\mathrm{T}}$ escondida en esas columnas. Esa señal está distorsionada por ruido aleatorio. Si una luz fue encendida o apagada durante el video, el objetivo es ver cuándo ocurrió.

Esto nos lleva a preguntar: **¿ Cuánto cambian los valores singulares de $\boldsymbol{A}$ a $\boldsymbol{A+B}$?** Entendemos los cambios en los autovalores de las matrices simétricas. Podemos estudiar o $A^{\mathrm{T}}A$ o AA^{T} o esta matriz simétrica de tamaño $m+n$ con autovalores σ_i y $-\sigma_i$:

$$\begin{bmatrix} 0 & A \\ A^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{v}_i \end{bmatrix} = \sigma_i \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{v}_i \end{bmatrix} \text{ y } \begin{bmatrix} 0 & A \\ A^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} -\boldsymbol{u}_i \\ \boldsymbol{v}_i \end{bmatrix} = -\sigma_i \begin{bmatrix} -\boldsymbol{u}_i \\ \boldsymbol{v}_i \end{bmatrix} \quad (25)$$

Recomendamos los fantásticos apuntes de Terry Tao: **https://terrytao.wordpress.com/2010/01/12/254a-notes-3a-eigenvalues-and-sums-of- hermitian-matrices/**

Desigualdades de Weyl $\boldsymbol{\sigma_{i+j-1}(A+B) \le \sigma_i(A) + \sigma_j(B)}$ (26)

$\boldsymbol{i \le m \le n}$ $\boldsymbol{|\sigma_i(A+B) - \sigma_i(A)| \le ||B||}$ (27)

Problemas de la Sección III.2

1 Un vector unitario $\boldsymbol{u}(t)$ describe un punto que se mueve sobre la esfera unitaria $\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u} = 1$. Muestre que el vector de velocidad $d\boldsymbol{u}/dt$ es ortogonal a la posición: $\boldsymbol{u}^{\mathrm{T}}(d\boldsymbol{u}/dt) = 0$.

2 Suponga que sumamos a S una matriz de **rango dos** semidefinida positiva. ¿Qué desigualdades entrelazadas conectan los autovalores λ de S y α de $S + \boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} + \boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}$?

3 (a) Encuentre los autovalores $\lambda_1(t)$ y $\lambda_2(t)$ de $A = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix} + t \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$.

(b) En $t = 0$, encuentre los autovectores de $A(0)$ y verifique que $\dfrac{d\lambda}{dt} = \boldsymbol{y}^{\mathrm{T}} \dfrac{dA}{dt}\, \boldsymbol{x}$.

(c) Verifique que el cambio $A(t) - A(0)$ es semidefinido positivo para $t > 0$. Luego verifique la ley de entrelazamiento $\lambda_1(t) \geq \lambda_1(0) \geq \lambda_2(t) \geq \lambda_2(0)$.

4 S es una matriz simétrica con autovalores $\lambda_1 > \lambda_2 > \ldots > \lambda_n$ y autovectores $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_n$. Determine cuáles de esos autovectores son una base para un subespacio i-dimensional Y con la propiedad: el mínimo de $\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}/\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$ para $\boldsymbol{x}$ en Y es λ_i.

5 Halle los autovalores de A_3, de A_2 y de A_1. Muestre que están entrelazados,:

$$A_3 = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \qquad A_2 = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} \qquad A_1 = \begin{bmatrix} 1 \end{bmatrix}$$

6 Sea D la matriz diagonal diag $(1, 2, \ldots, n)$ y sea S definida positiva.

1) Encuentre las derivadas en $t = 0$ de los autovalores $\lambda(t)$ de $D + tS$.
2) Muestre que para $t > 0$ pequeño, los λ entrelazan los números $1, 2, \ldots, n$.
3) Para $t > 0$, halle cotas para $\lambda_{\text{mín}}\,(D + tS)$ y $\lambda_{\text{máx}}\,(D + tS)$.

7 Suponga que D es de nuevo diag $(1, 2, \ldots, n)$ y A es una matriz de n por n.

1) Encuentre las derivadas en $t = 0$ de los valores singulares $\sigma(t)$ de $D + tA$.
2) ¿Qué dicen las desigualdades de Weyl sobre $\sigma_{\text{máx}}\,(D + tA)$ y $\sigma_{\text{mín}}\,(D + tA)$?

8 (a) Muestre que todo subespacio i-dimensional $\boldsymbol{V}$ contiene un vector no nulo $\boldsymbol{z}$ que es una combinación de $\boldsymbol{q}_i, \boldsymbol{q}_{i+1}, \ldots, \boldsymbol{q}_n$. (Esos $\boldsymbol{q}$ expanden un espacio $\boldsymbol{Z}$ de dimensión $n - i + 1$. Basado en las dimensiones i y $n - i + 1$, ¿por qué $\boldsymbol{Z}$ intesecta a $\boldsymbol{V}$?)

(b) ¿Por qué ese vector $\boldsymbol{z}$ satisface $\boldsymbol{z}^{\mathrm{T}}S\boldsymbol{z}/\boldsymbol{z}^{\mathrm{T}}\boldsymbol{z} \leq \boldsymbol{\lambda_i}$? Luego explique :

$$\boxed{\lambda_i = \max_{\dim \boldsymbol{V} = i} \min_{z \text{ in } \boldsymbol{V}} \frac{\boldsymbol{z}^{\mathrm{T}}S\boldsymbol{z}}{\boldsymbol{z}^{\mathrm{T}}\boldsymbol{z}}}$$

La ley de inercia

Definición Si S es simétrica y C es invertible, entonces la matriz $C^{\mathrm{T}}SC$ es "congruente a S". ¡ Esto no es similaridad $B^{-1}SB$! Los autovalores de $C^{\mathrm{T}}SC$ pueden diferir de los autovalores de S, *pero no pueden cambiar de signo*. Esto se llama la **"ley de inercia"** :

$C^{\mathrm{T}}SC$ tiene la misma cantidad de autovalores (positivos) (negativos) (nulos) que S.

Mi prueba favorita comienza con $C = QR$ (por Gram-Schmidt). A medida que R cambia gradualmente a I, $C^{\mathrm{T}}SC$ cambia gradualmente a $Q^{\mathrm{T}}SQ = Q^{-1}SQ$. Ahora tenemos similaridad ($Q^{-1}SQ$ tiene los mismos autovalores que S). Si R es invertible a medida que se convierte en I, entonces ningún autovalor puede cruzar al cero en el camino. *Sus signos son los mismos para $C^{\mathrm{T}}SC$, $Q^{-1}SQ$ y S.*

Los principios max-min también prueban la ley de inercia.

9 Si $S = LDL^{\mathrm{T}}$ tiene n pivotes distintos de cero en la eliminación, muestre que **los signos de los pivotes de S (en D) coinciden con los signos de los autovalores de S.** Aplique la ley a S y D.

10 Muestre que esta matriz KKT H de $2n \times 2n$ tiene n autovalores positivos y n autovalores negativos :

$$\begin{matrix} \boldsymbol{S} \textbf{ definida positiva} \\ \boldsymbol{C} \textbf{ invertible} \end{matrix} \qquad \boldsymbol{H} = \begin{bmatrix} \boldsymbol{S} & \boldsymbol{C} \\ \boldsymbol{C^{\mathrm{T}}} & \boldsymbol{0} \end{bmatrix}$$

Los primeros n pivotes de S son positivos. Los últimos n pivotes se obtienen de $-C^{\mathrm{T}}S^{-1}C$.

11 La matriz KKT H es simétrica e indefinida (este problema cuenta autovalores) :

$$H = \begin{bmatrix} \boldsymbol{S} & \boldsymbol{A^{\mathrm{T}}} \\ \boldsymbol{A} & \boldsymbol{0} \\ n & m \end{bmatrix} \begin{matrix} n \\ m \end{matrix} \quad \text{como en la ecuación (17)}$$

H se obtiene de minimizar $\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}$ (definida positiva) con m restricciones, $A\boldsymbol{x} = \boldsymbol{b}$. La eliminación en H comienza con S. Sabemos que sus n pivotes son todos positivos.

Luego la eliminación multiplica AS^{-1} por $\begin{bmatrix} S & A^{\mathrm{T}} \end{bmatrix}$ y resta de $\begin{bmatrix} A & 0 \end{bmatrix}$ para obtener $\begin{bmatrix} 0 & -AS^{-1}A^{\mathrm{T}} \end{bmatrix}$. Ese complemento de Schur $-AS^{-1}A^{\mathrm{T}}$ es definido negativo. ¿ Por qué ? Así los últimos m pivotes de H (provenientes de $-AS^{-1}A^{\mathrm{T}}$) son negativos.

12 Si $\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x} > 0$ para todo $\boldsymbol{x} \neq \boldsymbol{0}$ y C es invertible, ¿ por qué $(C\boldsymbol{y})^{\mathrm{T}}S(C\boldsymbol{y})$ es también positivo ? Esto también muestra que si S tiene todos autovalores positivos, lo mismo ocurre con $C^{\mathrm{T}}SC$.

III.3. Valores singulares que decaen rápido

Existen matrices importantes cuyos valores singulares cumplen $\sigma_k \le Ce^{-ak}$. Esos números decaen rápidamente. A menudo las matrices son invertibles (sus inversas son increíblemente grandes). Y a menudo tenemos una *familia* de matrices de todos los tamaños: matrices de Hilbert, Vandermonde, Hankel, Cauchy, Krylov y matrices espectrales de diferencias y más.

Trabajar con estas matrices es al mismo tiempo fácil y difícil. Fácil porque solo unos pocos valores singulares son significativos. Difícil porque la inversa tiene una norma gigante, que aumenta exponencialmente con el tamaño N de la matriz. Nos enfocaremos en dos de muchos ejemplos :

1 La transformada de Fourier discreta y *no uniforme* (**NUDFT**, por su sigla en inglés) tiene $\boldsymbol{U = A.* F}$ en lugar de F.

2 La *matriz de Vandermonde* $\boldsymbol{V}$ ajusta un polinomio de grado $N-1$ a N puntos de datos.

De hecho, estos ejemplos están relacionados. Una DFT estándar ajusta N valores f_0 a f_{N-1} en los N puntos $\omega^k = e^{-2\pi ik/N}$. ¡ La matriz de Fourier es una matriz de Vandermonde ! Pero en lugar de puntos reales entre -1 y 1, la DFT interpola en *puntos complejos* equiespaciados sobre el círculo unitario $|e^{i\theta}| = 1$. El caso real produce matrices de Vandermonde horribles (casi singulares). El caso complejo produce una hermosa matriz de Fourier (ortogonal).

Comenzamos con esa matriz de Fourier compleja F (con espaciado equidistante como siempre). Multiplicar por F es rapidísimo con la transformada de Fourier rápida de la Sección IV.1 : $\frac{1}{2}N\log_2 N$ operaciones.

Para espaciado no uniforme $\boldsymbol{x_j \neq j/N}$, han desaparecido las identidades especiales de la FFT han desaparecido. Pero Ruiz-Antolin y Townsend mostraron cómo recuperar casi la misma velocidad :
escribamos la U no uniforme como $A_{jk}F_{jk}$ donde $\boldsymbol{A}$ **está cerca de una matriz de rango bajo :**

$$\boldsymbol{F_{jk} = e^{-2\pi ikj/N}} \quad \text{y} \quad \boldsymbol{U_{jk} = e^{-2\pi ikx_j} = A_{jk}F_{jk}}$$

Cuando $U = F$ es la DFT con espaciado equidistante, cada $A_{jk} = 1$. *A tiene rango uno.* Con espaciado desigual, A es prácticamente una matriz de rango bajo, y las transformadas rápidas son posibles. He aquí símbolos para la multiplicación elemento por elemento de $A_{jk}F_{jk}$ y la división U_{jk}/F_{jk} :

$$\textbf{Multiplicación} \quad \boldsymbol{U = A .* F = A \bigcirc F} \qquad \textbf{División} \quad \boldsymbol{A = U \oslash F.} \tag{1}$$

La operación a ejecutar rápidamente es la NUDFT : $\boldsymbol{U}$ **multiplicada por** $\boldsymbol{c}$. La transformada rápida de Fourier calcula F por $\boldsymbol{c}$. Los cocientes U_{jk}/F_{jk} dan una matriz A que **es casi de rango bajo. Así la transformada no uniforme $\boldsymbol{U}$ se obtiene de una corrección $\boldsymbol{A}$ de la matriz de Fourier $\boldsymbol{F}$:**

$$\boxed{\boldsymbol{A \approx y_1 z_1^{\mathrm{T}} + \cdots + y_r z_r^{\mathrm{T}}} \quad \text{y} \quad \boldsymbol{Uc \approx Y_1 F Z_1 c + \cdots + Y_r F Z_r c.}} \tag{2}$$

Y_i y Z_i son matrices diagonales de N por N con $\boldsymbol{y}_i$ y $\boldsymbol{z}_i$ en sus diagonales principales. El espaciado equidistante implica que $U = F$, $A =$ **unos**, $\boldsymbol{y}_1 = \boldsymbol{z}_1 = (1, \ldots, 1)$ e $Y_1 = Z_1 = I$. Para el espaciado desigual, r está determinada por la no uniformidad de los puntos muestrales x_i.

Puntos muestrales x_j cerca de j/N

En este caso "apenas perturbado", podemos aparear los x_j no uniformes con las fracciones equidistantes j/N para $0 \le j < N$. Para cada elemento de U, A_{jk} es una corrección de F_{jk}:

$$\boxed{\boldsymbol{U_{jk} = A_{jk}F_{jk}} \text{ es } \boldsymbol{e^{-2\pi i k x_j} = e^{-2\pi i k(x_j - j/N)}\, e^{-2\pi i k j/N}}} \tag{3}$$

Entonces el paso clave para un algoritmo rápido es encontrar una aproximación cercana a esta matriz A (sabemos que F es rápida). El teorema de Eckart-Young sugiere que usemos la SVD de A. Pero la SVD es un paso mucho más costoso que la transformada rápida no equidistante.

A es una matriz interesante. Todas sus componentes en la ecuación (3) tienen la forma $e^{i\theta}$. Si reemplazamos $A_{jk} = e^{-i\theta}$ por su serie de potencias $1 - i\theta + \cdots$ entonces A comenzará con la matriz de unos.

El resto de esta sección desarrolla aproximaciones de rango pequeño usando la ecuación de Sylvester. Aquí Townsend toma un camino distinto: *aproximar la función* e^{-ixy}.

La idea clave es reemplazar la serie de Taylor de cada una de las componentes de esa matriz por una **expansión de Chebyshev**. Esto refleja una regla de gran importancia en el análisis numérico:

Las series de Fourier son buenas para funciones periódicas en un intervalo de tipo $|\theta| \le \pi$.

Las series de Chebyshev son buenas para funciones no periódicas en un intervalo de tipo $|x| \le 1$.

La conexión entre Fourier y Chebyshev es $\boldsymbol{\cos\theta = x}$. La función base de Fourier $\cos n\theta$ se vuelve el polinomio de Chebyshev $T_n(x) = \cos(n \arccos x)$. La base de Chebyshev comienza con $T_0 = 1$, $T_1 = x$ y $T_2 = 2x^2 - 1$ porque $\cos 2\theta = 2\cos^2\theta - 1$. Todas esas funciones base cumplen máx $|T_n(x)| =$ máx $|\cos n\theta| = 1$ sobre el intervalo que va de $x = -1$ a $x = 1$.

Un punto importante es que las n soluciones de $T_n(x) = 0$ *no* son equidistantes. Los ceros de $\cos n\theta$ están equiespaciados, pero en la variable x esos puntos se juntan cerca de los límites -1 y 1. La interpolación en estos $x = \cos(\pi(2k-1)/2n)$ es *mucho más estable* que la interpolación equidistante.

El sistema computacional altamente desarrollado en **chebfun.org** está basado en los polinomios de Chebyshev (para funciones en una o más dimensiones). Calcula un polinomio muy cercano a $f(x)$. Entonces todas las operaciones sobre funciones producen nuevos polinomios como series finitas de Chebyshev. El grado necesario para una precisión alta es determinado por el código **chebfun**.

Este enfoque de la matriz A llevó a Ruiz-Antolin y a Townsend a una prueba de *rango efectivo pequeño* (una aproximación cercana a A por una matriz de rango pequeño). Su artículo proporciona un código muy eficiente para la transformada de Fourier no uniforme.

La ecuación de Sylvester

Nos concentramos en el problema central del tema: **¿Qué familias de matrices tienen rango efectivo bajo?** El objetivo es encontrar un criterio que revele esta propiedad. A los ejemplos de la matriz de Hilbert y de la matriz A de NUDFT queremos agregar Vandermonde. Y esperamos un criterio que pueda aplicarse a clases de matrices mucho más amplias, más allá del ejemplo de Vandermonde.

He aquí el "**criterio de Sylvester**" = "**prueba A, B, C**" desarrollado por Beckermann y Townsend. Las palabras *desplazamiento de rango pequeño* y *matriz estructurada* suelen aplicarse a X.

> Si $\boldsymbol{AX - XB = C}$ **tiene rango** $\boldsymbol{r}$ para matrices normales A, B sin autovalores compartidos, entonces los valores singulares de X decaen a una tasa exponencial determinada por A, B, y C.

Una matriz es normal si $\overline{\boldsymbol{A}}^{\mathbf{T}}\boldsymbol{A} = \boldsymbol{A}\overline{\boldsymbol{A}}^{\mathbf{T}}$. Luego A tiene autovectores ortogonales: $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\overline{\boldsymbol{Q}}^{\mathbf{T}}$. Las matrices simétricas y las ortogonales son normales porque la prueba da $S^2 = S^2$ e $I = I$. La ecuación de la matriz de Sylvester $AX - XB = C$ es importante en teoría de control y el caso particular cuando $B = -\overline{A}^{\mathrm{T}}$ se llama la ecuación de Lyapunov.

¡ Esta prueba de Sylvester requiere que encontremos A, B, y C ! Esto se ha hecho para las importantísimas familias de matrices de Toeplitz, Hankel, Cauchy y Krylov (incluyendo a la matriz V de Vandermonde). Todas esas matrices resuelven la ecuación de Sylvester para elecciones simples de A, B, C. V es el ejemplo perfecto:

Matriz de Vandermonde
$$\boldsymbol{V} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ . & . & . & & . \\ 1 & x_n & x_n^2 & \dots & x_n^{n-1} \end{bmatrix} \tag{4}$$

V es la "matriz de interpolación" de n por n . Es invertible siempre y cuando los puntos $x_1, \dots, x_n$ sean todos distintos. Resolvemos $V\boldsymbol{c} = \boldsymbol{f}$ cuando queremos los coeficientes de un polinomio $p = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}$. Multiplicar V por $\boldsymbol{c}$ nos da el valor de p en los puntos $x = x_1, \dots, x_n$. Luego $V\boldsymbol{c} = \boldsymbol{f}$ dice que el polinomio interpolador tiene los valores deseados $f_1, \dots, f_n$ en esos n puntos. **El polinomio se ajusta con exactitud a los datos.**

Observamos que V es la matriz de Fourier F cuando elegimos puntos complejos $x_1 = \omega = e^{-2\pi i/N}$ y $x_k = \omega^k$. Esta F tiene rango completo. En el caso de Fourier A y B (debajo) tienen los mismos autovalores ω^k : *esto no está permitido*. A y B no satisfacen nuestro requisito de *autovalores bien separados*. Además, todos los valores singulares tienen el mismo tamaño : no decaen.

Las matrices de Vandermonde con *números reales* x_1 a x_n tienen decaimiento exponencial en sus valores singulares. Para confirmar esto aplicamos la prueba A, B, C usando estas matrices:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{x_1} & & & \\ & \boldsymbol{x_2} & & \\ & & . & \\ & & & \boldsymbol{x_n} \end{bmatrix} \quad \boldsymbol{B} = \begin{bmatrix} 0 & 0 & . & \boldsymbol{-1} \\ \boldsymbol{1} & 0 & 0 & . \\ 0 & \boldsymbol{1} & 0 & 0 \\ 0 & 0 & \boldsymbol{1} & 0 \end{bmatrix} \quad \boldsymbol{C} = \begin{bmatrix} 0 & 0 & 0 & x_1^n + 1 \\ 0 & 0 & 0 & x_2^n + 1 \\ 0 & 0 & 0 & . \\ 0 & 0 & 0 & x_n^n + 1 \end{bmatrix} \tag{5}$$

Esas matrices A y B sin duda son normales. (Este requisito podría ser debilitado pero aquí no es necesario.) Los autovalores de B son puntos equidistantes sobre la circunferencia unitaria. Esos λ tienen ángulos $\pi/n, 3\pi/n, \ldots, (2n-1)\pi/n$, y no son números reales cuando n es par. Así no tocan los autovalores reales $x_1, \ldots, x_n$ de A. Y C tiene rango 1. *Cumple el criterio A, B, C de Sylvester.*

El gráfico de valores singulares confirma que V está muy mal condicionada. Lo mismo para K.

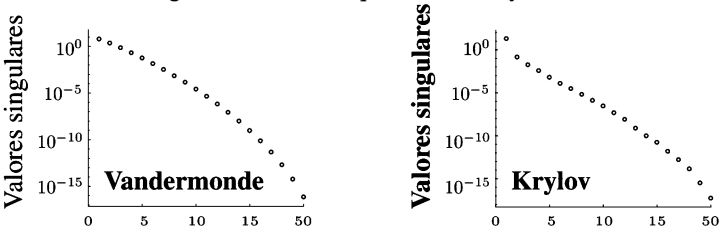


Figura III.2: valores singulares de Vandermonde y Krylov para V y $K = [\boldsymbol{b}\ A\boldsymbol{b} \ldots A^{n-1}\boldsymbol{b}]$. Aquí $\boldsymbol{b}$ = todos unos y A_{ij} = números aleatorios (normal estándar). V es igual a K cuando $A = \text{diag}\,(1/n, 2/n, \ldots, 1)$.

Una prueba de Sylvester mejorada

El requisito de que $AX - XB = C$ tenga rango bajo es mucho más estricto que la conclusión de que X tenga valores singulares que decaen rápido. Un teorema perfecto haría corresponder las hipótesis de C_n con la conclusión sobre X_n. Sin duda no es cierto que las matrices de Vandermonde o Krylov X_n de un tamaño creciente n tengan rango acotado. Así que tenemos que debilitar el requisito de rango bajo en C_n, mientras mantenemos el rápido decaimiento de los valores singulares de X_n.

Townsend ha encontrado ese teorema: **"rango logarítmico" para C lleva a "rango logarítmico" para X**. Un artículo reciente con Udell establece que el rango logarítmico es una propiedad muy extendida. He aquí la definición (muchos ejemplos de rango logarítmico tienen $q = 0$ o $q = 1$).

> Una familia de matrices C_n tiene **rango logarítmico** si $|\,(C_n - E_n)_{ij}\,| < \epsilon$ para matrices cercanas E_n con **rango** $(E_n) < c\,(\log n)^q$ (6)

Ejemplo 1 La **función núcleo de base radial** se usa frecuentemente en máquinas de vectores de soporte (support vector machines)

$$\textbf{Núcleo RBF} \qquad K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{x}'||^2}{2\sigma^2}\right)$$

Para un conjunto de vectores predictores $\boldsymbol{x}_i$ esto produce componentes $0 < K_{ij} < 1$ en una *matriz completa* K. Es imposible calcularlas a todas. Con buenas aproximaciones resolvemos problemas de clasificación no lineales con el llamado truco del núcleo de VII.5. Esa matriz tiene rango efectivo bajo.

1. M. D. Buhmann, *Radial basis functions*, Acta Numerica **9** (2000) 1-38.
2. B. Fornberg y N. Flyer, *A Primer on Radial Basis Functions*, SIAM (2015).
3. T. Hofmann, B. Scholkopf, y A. J. Smola, *Kernel methods in machine learning*, Annals of Statistics **36** (2008) 1171-1220 (con numerosas referencias).

ADI y el problema de Zolotarev

En un párrafo corto podemos apuntar a dos ideas que conducen al decaimiento rápido de los valores singulares de X. La primera es una iteración ADI para resolver la ecuación de Sylvester $AX - XB = C$. El algoritmo *implícito de direcciones alternantes* da una solución computacionalmente eficiente. La segunda idea conecta los autovalores de A y B (es necesario que no se superpongan) con un problema de *aproximación racional*.

Ese "problema de Zolotarev" busca un cociente $r(x) = p(x)/q(x)$ de polinomios que sea pequeño en los autovalores de A y grande en los autovalores de B. Aproximar con funciones racionales $r(x)$ puede ser exponencialmente mejor que con polinomios, un ejemplo famoso es la aproximación de Newman del valor absoluto $|x|$. La precisión exponencial de $r(x)$ se conecta al decaimiento exponencial de los valores singulares de X.

ADI resuelve $AX - XB = C$
Matrices $X_{j+1/2}$ y X_{j+1}

$$\begin{aligned} X_{j+1/2}(B - p_j I) &= C - (A - p_j I)\, X_j \\ (A - q_j I)X_{j+1} &= C - X_{j+1/2}(B - q_j I) \end{aligned}$$

Una buena funcion racional $r(x) = p(x)/q(x)$ tiene raices p_j en el numerador y q_j en el denominador. Fue Zolotarev en 1877 (!) quien encontró los mejores p y q en un problema modelo. Con A y B en la prueba de Sylvester, la cota para $\sigma_{1+kr}(X)$ es el "número Z" por $\sigma_1(X)$, lo que significa un decaimiento exponencial de los valores singulares.

Townsend y Fortunato desarrollaron con esta idea un resolvedor superrápido de la ecuación de Poisson en un cuadrado. Ya son conocidos los resolvedores rápidos cuando X es la aproximación usual por diferencias de $\partial^2 u/\partial x^2 + \partial^2 u/\partial y^2$ en 5-puntos. Su objetivo era un método espectral con complejidad óptima.

1. B. Beckermann, The condition number of real Vandermonde, Krylov, and positive definite Hankel matrices, *Numerische Mathematik* **85** (2000) 553-577.
2. B. Beckermann y A. Townsend, On the singular values of matrices with displacement structure, *SIAM J. Matrix Analysis*, arXiv : 1609.09494v1, 29 Sep 2016.
3. P. Benner, R.-C. Li, y N. Truhar, On the ADI method for Sylvester equations, *J. Comput. Appl. Math.* **233** (2009) 1035-1045.
4. D. Fortunato y A. Townsend, Fast Poisson solvers for spectral methods, arXiv : 1710.11259v1, 30 Oct 2017.
5. D. Ruiz-Antolin y A. Townsend, A nonuniform fast Fourier transform based on low rank approximation, arXiv: 1701.04492, *SIAM J. Sci. Comp.* **40(1)** (2018).
6. A. Townsend y H. Wilber, On the singular values of matrices with high displacement rank, arXiv :17120.5864, Linear Alg. Appl. **548** (2018) 19-41.
7. A. Townsend, **www.math.cornell.edu/∼ ajt/presentations/LowRankMatrices.pdf**
8. M. Udell y A. Townsend, Nice latent variable models have log-rank, arXiv : 1705.07474v2, *SIAM J. Math. of Data Science*, en prensa.

Problemas de la Sección III.3

1 Verifique que una **matriz de Krylov** $K = [\boldsymbol{b}\ A\boldsymbol{b} \dots A^{n-1}\boldsymbol{b}]$ satisface una ecuación de Sylvester $AK - KB = C$ con B como en la ecuación (6). Encuentre la matriz C.

2 Muestre que la **matriz de Hilbert** H cumple el criterio de Sylvester $AH - HB = C$

$$H_{ij} = \frac{1}{i+j-1} \qquad A = \frac{1}{2}\text{diag}\,(1, 3, \dots, 2n-1) \qquad B = -A \qquad C = \mathbf{unos}(n)$$

3 Una **matriz de Toeplitz** T tiene diagonales constantes (IV.5). Calcule $AT - TA^{\mathrm{T}} = C$:

$$T = \begin{bmatrix} t_0 & t_{-1} & \cdot & \cdot \\ t_1 & t_0 & t_{-1} & \cdot \\ \cdot & t_1 & t_0 & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \qquad A = \begin{bmatrix} 0 & 0 & 0 & \cdot \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \cdot & 0 & 1 & \cdot \end{bmatrix} \qquad B = A^{\mathrm{T}}$$

4 Una **matriz de Hankel** H tiene *anti*diagonales constantes, como la matriz de Hilbert. Así H_{ij} depende solo de $i + j$. Cuando H es simétrica y definida positiva, Beckermann y Townsend muestran que $H = \overline{K}^{\mathrm{T}} K$ para una matriz de Krylov K (como en el Problema 1). Entonces $\sigma_j(H) = |\sigma_j(K)|^2$ (*¿ Por qué ?*) y los valores singulares de H decaen rápido.

5 Una **matriz de Pick** tiene componentes $P_{jk} = (s_j + s_k)/(x_j + x_k)$ donde $\boldsymbol{x} = (x_1, \dots, x_n) > 0$, y $s = (s_1, \dots, s_n)$ puede contener componentes complejas. Muestre que $\boldsymbol{AP - P(-A)} = \boldsymbol{s1^{\mathrm{T}} + 1s^{\mathrm{T}}}$ para $\mathbf{1}^{\mathrm{T}} = [1 \quad 1 \dots 1]$ y $A = \text{diag}\,(x_1, \dots, x_n)$. A tiene autovalores positivos, $B = -A$ tiene autovalores negativos, y satisfacen el criterio de Sylvester.

6 Para una matriz invertible X que cumple con la ecuación de Sylvester $AX - XB = C$, encuentre una ecuación de Sylvester para X^{-1}.

7 Si $A = Q\Lambda\overline{Q}^{\mathrm{T}}$ tiene autovectores ortogonales complejos $\boldsymbol{q}_1, \dots, \boldsymbol{q}_n$ en las columnas de Q, verifique que $\overline{A}^{\mathrm{T}} A = A\overline{A}^{\mathrm{T}}$: *es decir, A es normal*. Los autovalores pueden ser complejos.

8 Si $S^{\mathrm{T}} = S$, $Z^{\mathrm{T}} = -Z$ y $SZ = ZS$, verifique que $A = S + Z$ es normal. Ya que S tiene autovalores reales y Z tiene autovalores imaginarios, es probable que $A = S + Z$ tenga autovalores complejos.

9 Muestre que la ecuación (3) para $U\boldsymbol{c}$ se sigue de la ecuación (2) para A.

III.4. Algoritmos de partición para $\ell^2 + \ell^1$

Estos temas son muy importantes. Se merecen un libro entero ya que incluyen **búsqueda de base (basis pursuit)** y **optimización LASSO** más **completación de matrices** y **compressed sensing** de III.5. Lo que podemos hacer aquí es presentar ideas básicas y algoritmos exitosos.

Comenzamos con un sistema lineal $A\boldsymbol{x} = \boldsymbol{b}$. Supongamos que A tiene muchas más columnas que filas ($m \ll n$). Entonces $A\boldsymbol{x}_n = \boldsymbol{0}$ tiene muchas soluciones (A tiene un espacio nulo grande). Si $A\boldsymbol{x} = \boldsymbol{b}$ tiene una solución, entonces $\boldsymbol{x} + \boldsymbol{x}_n$ es también otra solución. ¿ Cuál elegimos ?

Para minimizar la norma $\boldsymbol{\ell^2}$ de $\boldsymbol{x}$, quitamos su componente del espacio nulo. Eso nos deja la solución de norma mínima $\boldsymbol{x}^+ = A^+\boldsymbol{b}$, que se obtiene con la pseudoinversa de A. Esta es la solución de II.2 que usa la SVD. Pero $\boldsymbol{x}^+$ generalmente tendrá muchas componentes pequeñas, esto la hace muy difícil de interpretar. En MATLAB, **pinv(A)** $*$ $\boldsymbol{b}$ encuentra $\boldsymbol{x}^+$. La barra invertida $A\backslash\boldsymbol{b}$ encuentra una solución bastante rala, pero probablemente no encuentre el $\boldsymbol{x}$ óptimo para la *búsqueda de base*.

Una solución para $A\boldsymbol{x} = \boldsymbol{b}$ con **muchas componentes nulas** (*una solución rala*, si existe) se obtiene de minimizar la norma $\boldsymbol{\ell^1}$ en lugar de la norma $\boldsymbol{\ell^2}$:

$$\boxed{\textbf{Búsqueda de base} \qquad \textbf{Minimizar } ||\boldsymbol{x}||_1 = |\boldsymbol{x}_1| + \cdots + |\boldsymbol{x}_n| \textbf{ sujeto a } A\boldsymbol{x} = \boldsymbol{b}.} \tag{1}$$

Este es un problema de optimización convexo, porque la norma ℓ^1 es una función convexa de $\boldsymbol{x}$. La norma ℓ^1 es *lineal a trozos*, a diferencia de ℓ^2 y todas las otras normas ℓ^p excepto $||\boldsymbol{x}||_\infty = \text{máx}|x_i|$. La búsqueda de base no se resuelve con la SVD de A, que se conecta con la norma ℓ^2. Pero ℓ^1 es famosa por dar soluciones ralas, y se han encontrado algoritmos rápidos.

La solución más rala de $A\boldsymbol{x} = \boldsymbol{b}$ minimiza $||\boldsymbol{x}||_{\boldsymbol{0}} =$ *número de componentes no nulas de* $\boldsymbol{x}$. Pero esta no es una verdadera norma : $||2\boldsymbol{x}||_0 = ||\boldsymbol{x}||_0$. Los vectores con $||\boldsymbol{x}||_0 = 1$, es decir con solo una componente no nula, yacen a lo largo de los ejes de coordenadas, como $\boldsymbol{i} = (1, 0)$ y $\boldsymbol{j} = (0, 1)$. Entonces relajamos o "convexificamos" el problema $\boldsymbol{\ell^0}$ para conseguir una norma verdadera: **y esa norma es** la $\boldsymbol{\ell^1}$. Los vectores con $||\boldsymbol{x}||_1 = |x_1| + |x_2| \leq 1$ rellenan el diamante con esquinas en $\pm\boldsymbol{i}$ y $\pm\boldsymbol{j}$.

Un problema relacionado permite *ruido* en $A\boldsymbol{x} = \boldsymbol{b}$; no se requiere una solución exacta. Ahora la norma $\boldsymbol{\ell^1}$ entra como una penalización $\boldsymbol{\lambda}||\boldsymbol{x}||_{\boldsymbol{1}}$ o como una restricción $||\boldsymbol{x}||_{\boldsymbol{1}} \leq \boldsymbol{t}$:

$$\boxed{\textbf{LASSO} \text{ (en estadística)} \qquad \begin{matrix} \textbf{Mimimizar } \frac{1}{2}||A\boldsymbol{x} - \boldsymbol{b}||_2^2 + \boldsymbol{\lambda}||\boldsymbol{x}||_1 \text{ o} \\ \textbf{Mimimizar } \frac{1}{2}||A\boldsymbol{x} - \boldsymbol{b}||_2^2 \text{ con } ||\boldsymbol{x}||_1 \leq \boldsymbol{t} \end{matrix}} \tag{2}$$

Tibshirani inventó LASSO para mejorar la regresión vía mínimos cuadrados. Tiene soluciones más ralas que la "regresión ridge" que usa la norma ℓ^2 en la penalización y en la restricción. Geométricamente, la diferencia entre $||\boldsymbol{x}||_1 \leq 1$ y $||\boldsymbol{x}||_2 \leq 1$ está en la forma de esos dos conjuntos : **diamante para $\boldsymbol{\ell^1}$ contra esfera para $\boldsymbol{\ell^2}$**. Vea las dos figuras en la Sección I.11.

Un conjunto convexo $||A\boldsymbol{x} - \boldsymbol{b}||_2^2 = C$ tiene buenas probabilidades de cortar a ese diamante en una de sus puntas . *Los vértices más afilados son vectores ralos*. Pero la esfera no tiene vértices. El $\boldsymbol{x}$ óptimo casi nunca es ralo en la optimización ℓ^2. Ya que ese conjunto convexo redondo puede tocar la esfera en cualquier lado, la solución ℓ^2 tiene muchos valores no nulos.

Algortimos de partición para optimización ℓ^1

Las soluciones ralas son una razón esencial para minimizar en la norma ℓ^1. Los pequeños valores no nulos que aparecen en ℓ^2 desaparecerán en ℓ^1. Era frustrante saber esto cuando los algoritmos ℓ^1 eran muy lentos. Ahora la implementación numérica de la optimización ℓ^1 se ha puesto al día con la teoría.

La idea es esta. Muchos problemas importantes de optimización combinan **dos términos** :

$$\boxed{\boldsymbol{F_1}\text{ y }\boldsymbol{F_2}\textbf{ convexas} \qquad \textbf{Minimizar } \boldsymbol{F_1(x) + F_2(x)} \textbf{ para } \boldsymbol{x} \textbf{ en un conjunto convexo } \boldsymbol{K}} \tag{3}$$

F_1 involucra una norma tipo ℓ^1 y F_2 involucra una norma tipo ℓ^2. La convexidad es muy valiosa. Pero no se relacionan bien, las iteraciones ℓ^2 (por lo general suficientemente rápidas) se ralentizan esperando que ℓ^1 aprenda qué componentes deberían ser distintas de cero.

La solución es "particionar" el algoritmo: alternar entre pasos ℓ^2 y pasos ℓ^1. En casos importantes, será posible resolver ℓ^1 de forma explícita con una operación de "contracción". Las iteraciones particionadas son mucho más rápidas y más efectivas que un paso ℓ^1-ℓ^2 mixto. Entre los algortimos más populares podemos encontrar **ADMM**, **Bregman particionado** y **Kaczmarz particionado** (descriptos a continuación).

Una manera de comenzar es ver las palabras clave que describen el desarrollo de los algoritmos. Esas palabras describen los pasos que mejoraron el método anterior y que convirtieron al algoritmo ADMM final en un éxito :

Descomposición dual

Lagrangiano aumentado

Método de multiplicadores

ADMM : método de los multiplicadores con direcciones alternadas, por su sigla en inglés

Esta presentación paso por paso sigue el plan que establecieron Boyd, Parikh, Chu, Peleato, y Eckstein en su excelente libro que está a disposición online sobre ADMM : *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. Fue publicado en *Foundations and Trends in Machine Learning*, **3** (2010) 1-122.

Estos autores incluyen un lindo historial (con referencias) para cada uno de los cuatro pasos del ADMM. He aquí el planteo del problema, dejando una gran libertad en la función convexa $f(x)$.

$$\boxed{\textbf{Minimizar } \boldsymbol{f(x)} \textbf{ sujeto a } \boldsymbol{Ax = b} \text{: } A \text{ es de } m \text{ por } n} \tag{4}$$

Este es el *problema primal* para $\boldsymbol{x} = (x_1, \ldots, x_n)$. El primer paso es combinar $A\boldsymbol{x} = \boldsymbol{b}$ con la función costo $f(\boldsymbol{x})$ introduciendo multiplicadores de Lagrange $y_1, \ldots, y_m$:

$$\boxed{\textbf{Lagrangiano} \qquad L(\boldsymbol{x}, \boldsymbol{y}) = f(\boldsymbol{x}) + \boldsymbol{y}^{\mathrm{T}}(A\boldsymbol{x} - \boldsymbol{b}) = f(\boldsymbol{x}) + \boldsymbol{y}^{\mathrm{T}}A\boldsymbol{x} - \boldsymbol{y}^{\mathrm{T}}\boldsymbol{b}} \tag{5}$$

La solución combinada $\boldsymbol{x}^*, \boldsymbol{y}^*$ es un *punto de ensilladura* de L: $\min_{\boldsymbol{x}} \max_{\boldsymbol{y}} L = \max_{\boldsymbol{y}} \min_{\boldsymbol{x}} L$. Las ecuaciones a resolver son $\partial L/\partial \boldsymbol{x} = \mathbf{0}$ y $\partial L/\partial \boldsymbol{y} = \mathbf{0}$. De hecho $\partial L/\partial \boldsymbol{y} = \mathbf{0}$ nos devuelve exactamente la restricción $A\boldsymbol{x} = \boldsymbol{b}$. Esta es una idea fundamental en optimización con restricciones. En el Capítulo VI, los y se interpretan como las derivadas $\partial L/\partial b$ en el $\boldsymbol{x}^*$ óptimo.

Ahora nos encontramos con el paso clave para ir del *problema primal* para $\boldsymbol{x} = (x_1, \ldots, x_n)$ al *problema dual* para $\boldsymbol{y} = (y_1, \ldots, y_m)$. **Minimizar $\boldsymbol{L(x, y)}$ sobre $\boldsymbol{x}$.** El mínimo ocurre en un punto $\boldsymbol{x}^*$ dependiente de $\boldsymbol{y}$. Así el problema dual es maximizar $\boldsymbol{m(y)} = \boldsymbol{L}(\boldsymbol{x}^*(\boldsymbol{y}), \boldsymbol{y})$.

El aumento más pronunciado de $m(y)$

Para maximizar una función $m(\boldsymbol{y})$, buscamos un punto $\boldsymbol{y}^*$ donde todas las derivadas parciales son cero. En otras palabras, **el gradiente es cero**: $\boldsymbol{\nabla m} = (\partial m/\partial y_1, \ldots, \partial m/\partial y_m) = \mathbf{0}$. El problema en cuestión tiene una linda fórmula para esas derivadas con respecto a $\boldsymbol{y}$ de m: $\boldsymbol{\nabla m} = A\boldsymbol{x}^* - \boldsymbol{b}$.

¿Cómo se maximiza una función $m(\boldsymbol{y})$ cuando conocemos sus derivadas primeras? Esta será la pregunta central del Capítulo VI, que conduce a los algoritmos que optimizan los pesos en aprendizaje profundo. Allí minimizaremos una función de pérdida; aquí maximizamos $m(\boldsymbol{y})$. Allí seguiremos el gradiente cuesta abajo (*el descenso más pronunciado*). Aquí seguiremos el gradiente cuesta arriba (*el ascenso más pronunciado*). En ambos casos el gradiente nos indica la dirección de mayor pendiente.

Aumento más pronunciado para máx mín $\boldsymbol{L} =$ máx $\boldsymbol{m}$

Encontrar $\boldsymbol{x_{k+1}}$ y seguir $\boldsymbol{\nabla m} = \boldsymbol{A x_{k+1}} - \boldsymbol{b}$

$$\boldsymbol{x}_{k+1} = \operatorname{argmin} L(\boldsymbol{x}, \boldsymbol{y}_k) \tag{6}$$

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + s_k(A\boldsymbol{x}_{k+1} - \boldsymbol{b}) \tag{7}$$

Ese número s_k es el **tamaño del paso**. Determina cuán lejos nos movemos en dirección cuesta arriba $\boldsymbol{\nabla m}$. No esperamos que alcance el máximo de m en un solo paso, solo damos pasos cuidadosos hacia arriba. Es común que los primeros pasos sean exitosos y los posteriores solo den pequeños aumentos de m. En matemática financiera esa variable dual $\boldsymbol{y}$ a menudo representa "precios".

Observe que "**argmin**" en la ecuación (6) es el punto $\boldsymbol{x}$ donde esa función L se minimiza. Esta introducción a la dualidad (minimizar sobre $\boldsymbol{x}$ y maximizar sobre $\boldsymbol{y}$ en cualquier orden) no fue solo un impulso descabellado. La razón se dará a continuación.

Descomposición dual

Supongamos que la función original $f(\boldsymbol{x})$ es separable: $f(\boldsymbol{x}) = f_1(\boldsymbol{x_1}) + \cdots + f_N(\boldsymbol{x}_N)$. Esos $\boldsymbol{x}_i$ son subvectores de $\boldsymbol{x} = (x_1, \ldots, x_n)$. Particionamos las columnas de A de la misma manera, por lo que $A = [A_1 \ \ldots \ A_N]$. Así la función de Lagrange $L(\boldsymbol{x}, \boldsymbol{y})$ se divide en N lagrangianos más sencillos, L_1 a L_N:

$$f(\boldsymbol{x}) + \boldsymbol{y}^{\mathrm{T}}(A\boldsymbol{x} - \boldsymbol{b}) = \sum_1^N L_i(\boldsymbol{x}_i, \boldsymbol{y}) = \sum_1^N \left[f_i(\boldsymbol{x}_i) + \boldsymbol{y}^{\mathrm{T}} A_i \boldsymbol{x}_i - \frac{1}{N}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{b} \right] \tag{8}$$

Ahora la minimización en $\boldsymbol{x}$ de $\boldsymbol{L}$ se divide en $\boldsymbol{N}$ minimizaciones a resolver en paralelo.

Problema dual descompuesto	$\boldsymbol{x}_i^{k+1} = \operatorname{argmin} L_i\,(\boldsymbol{x}_i, \boldsymbol{y}^k)$	(9)
$\boldsymbol{N}$ **problemas duales en paralelo**	$\boldsymbol{y}^{k+1} = \boldsymbol{y}^k + s_k(A\boldsymbol{x}^{k+1} - \boldsymbol{b})$	(10)

Los N nuevos $\boldsymbol{x}_i^{k+1}$ de (9) se agrupan en $A\boldsymbol{x}^{k+1}$ en (10). Luego el $\boldsymbol{y}^{k+1}$ resultante se distribuye en los N procesadores que ejecutan minimizaciones separadas en la próxima iteración de (9). Cuando $f(\boldsymbol{x})$ no es separable se puede ahorrar mucho más que si se realizara una gran minimización (6).

Lagrangianos aumentados

Para que las iteraciones (9)–(10) sean más robustas (para ayudarlas a converger si $f(x)$ no es estrictamente convexa) podemos aumentar $f(x)$ con un término de penalización de factor variable ρ:

$$\textbf{Lagrangiano aumentado} \quad L_\rho(\boldsymbol{x}, \boldsymbol{y}) = f(\boldsymbol{x}) + \boldsymbol{y}^{\mathrm{T}}(A\boldsymbol{x} - \boldsymbol{b}) + \frac{1}{2}\,\rho\,||A\boldsymbol{x} - \boldsymbol{b}||_2^2. \qquad (11)$$

Este es el lagrangiano para minimizar $f(\boldsymbol{x}) + \frac{1}{2}\,\rho\,||A\boldsymbol{x} - \boldsymbol{b}||^2$ con la restricción $A\boldsymbol{x} = \boldsymbol{b}$.

Los mismos pasos llevan a la maximización como en (6)–(7). Y la constante de penalización ρ se vuelve un tamaño de paso s apropiado: podemos mostrar que cada nuevo $(\boldsymbol{x}_{k+1}, \boldsymbol{y}_{k+1})$ satisface $\boldsymbol{\nabla f}(\boldsymbol{x}_{k+1}) + A^{\mathrm{T}}\boldsymbol{y}_{k+1} = \boldsymbol{0}$. Pero hay una gran desventaja al agregar el término de penalización con ρ: *incluso si $f(\boldsymbol{x})$ es separable el lagrangiano L_ρ no lo es.*

Necesitamos un paso más para mantener la ventaja de la separabilidad (que lleva a N maximizaciones más simples que se resuelven en paralelo) junto con la ventaja que se obtiene al incluir el término de penalización $\frac{1}{2}\,\rho\,||A\boldsymbol{x} - \boldsymbol{b}||^2$.

ADMM: método de los multiplicadores con direcciones alternantes

La nueva idea clave es particionar. La $f(x)$ original se divide en dos partes (posiblemente una parte ℓ^1 y una parte ℓ^2). Podemos llamar a estas partes f_1 y f_2, pero para evitar subíndices serán f y g. Permitimos que g tenga una nueva variable $\boldsymbol{z}$ (en lugar de $\boldsymbol{x}$), pero *recuperamos el problema original al agregar la restricción $\boldsymbol{x} = \boldsymbol{z}$*. Esta maniobra dudosa pero válida nos permite tener un problema separable, con la gran ventaja de hacer cálculos paralelos, las partes de $\boldsymbol{x}_{k+1}$ y $\boldsymbol{z}_{k+1}$ se distribuyen en diferentes computadoras.

La nueva restricción $\boldsymbol{x} = \boldsymbol{z}$ se une a la $A\boldsymbol{x} = \boldsymbol{b}$ original en un total de p restricciones lineales $A\boldsymbol{x} + B\boldsymbol{z} = \boldsymbol{c}$. **Ahora estamos minimizando $\boldsymbol{f(x) + g(z)}$**. Y como antes, para una convergencia más segura, aumentamos el lagrangiano:

$$\boxed{L_\rho(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}) = f(\boldsymbol{x}) + g(\boldsymbol{z}) + \boldsymbol{y}^{\mathrm{T}}(A\boldsymbol{x} + B\boldsymbol{z} - \boldsymbol{c}) + \tfrac{1}{2}\,\rho\,||A\boldsymbol{x} + B\boldsymbol{z} - \boldsymbol{c}||^2.} \qquad (12)$$

Ahora tenemos una ecuación adicional para actualizar $\boldsymbol{z}$ en cada paso. Como antes, el tamaño del paso s puede ser el coeficiente de regularización ρ. La ventaja clave del ADMM es que $\boldsymbol{x}$ y $\boldsymbol{z}$ se actualizan secuencialmente y no en conjunto. *Las dos funciones $f(\boldsymbol{x})$ y $g(\boldsymbol{z})$ se alternan*. Para la minimización en paralelo, una f o g separables permitirán la distribución de las partes.

He aquí los tres pasos para obtener un nuevo $\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}$ más cercano a $\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{y}^*$:

ADMM

$$\boxed{\begin{aligned} \boldsymbol{x}_{k+1} &= \operatorname*{argmin}_{\boldsymbol{x}} \; L_\rho\left(\boldsymbol{x}, \boldsymbol{z}_k, \boldsymbol{y}_k\right) && (13)\\ \boldsymbol{z}_{k+1} &= \operatorname*{argmin}_{\boldsymbol{z}} \; L_\rho\left(\boldsymbol{x}_{k+1}, \boldsymbol{z}, \boldsymbol{y}_k\right) && (14)\\ \boldsymbol{y}_{k+1} &= \boldsymbol{y}_k + \rho\left(A\boldsymbol{x}_{k+1} + B\boldsymbol{z}_{k+1} - \boldsymbol{c}\right) && (15) \end{aligned}}$$

En la práctica, ADMM puede ser lento para alcanzar una precisión alta, pero sorprendentemente rápido para alcanzar una precisión aceptable. Mencionamos aquí (adelantándonos al Capítulo VII sobre aprendizaje profundo) que una precisión modesta suele ser suficiente y hasta deseable (en situaciones donde el sobreajuste de los datos de entrenamiento lleva a resultados insatisfactorios en los datos de prueba).

Siguiendo con Boyd et al., reescalamos la variable dual $\boldsymbol{y}$. La nueva variable es $\boldsymbol{u} = \boldsymbol{y}/\rho$ y los términos lineales y cuadráticos se combinan en el lagrangiano. Esto produce un **ADMM escalado** que tiene ciertas ventajas en la práctica.

$$\boxed{\begin{aligned} \boldsymbol{x}_{k+1} &= \operatorname*{argmin}_{\boldsymbol{x}} \; \left(f(\boldsymbol{x}) + \tfrac{1}{2}\,\rho\,||A\boldsymbol{x} + B\boldsymbol{z}_k - \boldsymbol{c} + \boldsymbol{u}_k||^2\right) && (16)\\ \boldsymbol{z}_{k+1} &= \operatorname*{argmin}_{\boldsymbol{z}} \; \left(g(\boldsymbol{z}) + \tfrac{1}{2}\,\rho\,||A\boldsymbol{x}_{k+1} + B\boldsymbol{z} - \boldsymbol{c} + \boldsymbol{u}_k||^2\right) && (17)\\ \boldsymbol{u}_{k+1} &= \boldsymbol{u}_k + A\boldsymbol{x}_{k+1} + B\boldsymbol{z}_{k+1} - \boldsymbol{c} && (18) \end{aligned}}$$

En cualquier optimización, y sin dudas en esta, deberíamos identificar las ecuaciones que estamos resolviendo. Esas ecuaciones se satisfacen (como queríamos) por el par $\boldsymbol{x}^*, \boldsymbol{z}^*$ óptimo del problema primal y por $\boldsymbol{y}^*$ o $\boldsymbol{u}^*$ del problema dual :

$$\begin{aligned} \mathbf{0} &= \boldsymbol{\nabla f}(\boldsymbol{x}^*) + A^{\mathrm{T}}\boldsymbol{y}^* & \mathbf{0} &\in \boldsymbol{\partial f}(\boldsymbol{x}^*) + A^{\mathrm{T}}\boldsymbol{y}^* && (19)\\ \mathbf{0} &= \boldsymbol{\nabla g}(\boldsymbol{z}^*) + B^{\mathrm{T}}\boldsymbol{y}^* & \mathbf{0} &\in \boldsymbol{\partial g}(\boldsymbol{z}^*) + B^{\mathrm{T}}\boldsymbol{y}^* && (20)\\ &\text{Gradientes } \boldsymbol{\nabla f} \text{ y } \boldsymbol{\nabla g} & &\text{Subdiferenciales } \boldsymbol{\partial f} \text{ y } \boldsymbol{\partial g} \end{aligned}$$

Un tratamiento adecuado de este problema (y una prueba de convergencia para ADMM) requeriría más análisis convexo del que vemos en este libro. Pero ya se verá porqué se dio cada paso para llegar al ADMM escalado en (16)–(18). La convergencia se mantiene incluso si f y g no son estrictamente convexas (ver los últimos artículos de Hajinezhad). (Deben ser *cerradas* y *propias*. Esto permite $f = 0$ en un conjunto convexo no vacío cerrado y $f = +\infty$ en el complemento. Para tal función aparecen los subdiferenciales = *derivadas multivaluadas* en el borde de K.) Y el lagrangiano sin aumentar debe tener un punto de ensilladura.

Las siguientes páginas desarrollan cuatro ejemplos principales, siguiendo el libro sobre ADMM de Boyd.

1 D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Athena Scientific (1986).

2 M. Fortin y R. Glowinski, *Augmented Lagrangian Methods*, North-Holland (1983-5).

3 D. Hajinezhad y Q. Shi, *ADMM for a class of nonconvex bilinear optimization*, Journal of Global Optimization **70** (2018) 261–288.

Ejemplo 1 Este problema clásico de optimización convexa será el punto de partida del Capítulo VI. La función $f(\boldsymbol{x})$ es convexa. El conjunto K es cerrado y convexo.

$$\boxed{\text{Minimizar } f(\boldsymbol{x}) \text{ para } \boldsymbol{x} \text{ en } K.} \tag{21}$$

ADMM reescribe "$\boldsymbol{x}$ en $\boldsymbol{K}$" como una mimimización de $\boldsymbol{g}$. Conecta $\boldsymbol{x}$ a $\boldsymbol{z}$ con una restricción.

$$\boxed{\text{Minimizar } f(\boldsymbol{x}) + g(\boldsymbol{z}) \text{ sujeto a } \boldsymbol{x} - \boldsymbol{z} = \boldsymbol{0}.} \tag{22}$$

g es la **función indicadora** del conjunto K : $g(\boldsymbol{z}) = 0$ o $+\infty$ para $\boldsymbol{z}$ dentro o fuera del conjunto K. Así $g = 0$ en el mínimo, lo cual fuerza al minimizador a pertenecer a K. La función indicadora $g(\boldsymbol{z})$ es cerrada y convexa, porque el "cilindro" que queda por encima de su gráfico (sobre K) es cerrado y convexo. El lagrangiano aumentado escalado incluye el término de penalidad :

$$L(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}) = f(\boldsymbol{x}) + g(\boldsymbol{z}) + \frac{1}{2}\rho\,||\boldsymbol{x} - \boldsymbol{z} + \boldsymbol{u}||^2. \tag{23}$$

Note como el $\boldsymbol{\lambda}^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{z})$ usual fue absorbido en la ecuación (23) por el paso de escalamiento. Entonces ADMM separa (22) en *una minimización alternada con una proyección* :

ADMM

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \text{argmin}\,[f(\boldsymbol{x}) + \tfrac{1}{2}\rho\,||\boldsymbol{x} - \boldsymbol{z}_k + \boldsymbol{u}_k||^2] \\ \boldsymbol{z}_{k+1} &= \text{proyección de } \boldsymbol{x}_{k+1} + \boldsymbol{u}_k \text{ sobre } K \\ \boldsymbol{u}_{k+1} &= \boldsymbol{u}_k + \boldsymbol{x}_{k+1} - \boldsymbol{z}_{k+1} \end{aligned}$$

Ejemplo 2 **Umbrales suaves** El siguiente importante problema ℓ^1 tiene una solución exacta.

$f(\boldsymbol{x}) = \lambda||\boldsymbol{x}||_1 = \lambda|x_1| + \cdots + \lambda|x_n|$ se divide en n funciones escalares $\lambda\,|x_i|$.

La separación por el ADMM lleva a la minimización de cada función especial f_1 a f_n :

$$f_i(x_i) = \lambda|x_i| + \frac{1}{2}\rho\,(x_i - v_i)^2 \text{ con } v_i = z_i - u_i$$

La solución x_i^* es el "umbral suave" de v_i dibujado en la Sección VI.4 :

$$x_i^* = \left(v_i - \frac{\lambda}{\rho}\right)_+ - \left(-v_i - \frac{\lambda}{\rho}\right)_+ = v_i\left(1 - \frac{\lambda}{\rho\,|v_i|}\right)_+ \tag{24}$$

No solo es esta función umbral x_i^* una solución explícita a un problema no lineal importante, sino que también es un **operador de contracción : todo $\boldsymbol{v_i}$ se desplaza hacia cero**.

Pronto veremos este umbral suave como un operador "proximal".

Ejemplo 3 **Factorización matricial no negativa $A \approx CR$** con $C_{ij} \geq 0$ y $R_{ij} \geq 0$

ADMM comienza con una *minimización alternante*: nuestra manera favorita de factorizar una matriz.

Encontrar $C \geq 0$	Minimizar $\|\|A - CR\|\|_F^2$ con $R \geq 0$ fija
Encontrar $R \geq 0$	Minimizar $\|\|A - CR\|\|_F^2$ con $C \geq 0$ fija

Boyd et al. señalan un problema equivalente con C y R en la restricción $X = CR$:

$$\textbf{NMF} \qquad \text{Minimizar } \|\|A - X\|\|_F^2 + I_+(C) + I_+(R) \text{ con } X = CR$$

ADMM agrega un tercer paso que actualiza la variable dual U e introduce una nueva variable X que satisface la restricción $X = CR \geq 0$. La función indicadora $I_+(C)$ es cero para $C \geq 0$ e infinita en otro caso. Ahora ADMM se divide en una minimización en X y C, alternando con una minimización en R:

$$(X_{k+1}, C_{k+1}) = \text{argmin} \left[\|\|A - X\|\|_F^2 + \frac{1}{2}\rho \, \|\|X - CR_k + U_k\|\|_F^2 \right] \text{ con } X \geq 0, C \geq 0$$

$$R_{k+1} = \text{argmin} \, \|\|X_{k+1} - C_{k+1}R + U_k\|\|_F^2 \text{ con } R \geq 0$$

$$U_{k+1} = U_k + X_{k+1} - C_{k+1}R_{k+1}$$

Las filas de X_{k+1}, C_{k+1} y luego las columnas de R_{k+1} todas pueden encontrarse por separado. La separación promueve el cálculo en paralelo.

Ejemplo 4 LASSO busca una solución rala para $A\boldsymbol{x} = \boldsymbol{b}$ al incluir una penalización $\boldsymbol{\ell^1}$:

$$\textbf{LASSO} \qquad \text{Minimizar} \quad \frac{1}{2}\|\|A\boldsymbol{x} - \boldsymbol{b}\|\|^2 + \lambda \, \|\|\boldsymbol{x}\|\|_1 \tag{25}$$

Inmediatamente ese problema se divide en $f(\boldsymbol{x}) + g(\boldsymbol{z})$ con la restricción $\boldsymbol{x} - \boldsymbol{z} = \boldsymbol{0}$. El subproblema a resolver para hallar $\boldsymbol{x}$ es de mínimos cuadrados, encontrándonos con $A^{\mathrm{T}}A$. Aumentamos con $\frac{1}{2}\rho \, \|\|A\boldsymbol{x} - \boldsymbol{b}\|\|^2$.

ADMM escalado $\boldsymbol{x}_{k+1} = (A^{\mathrm{T}}A + \rho \, I)^{-1}(A^{\mathrm{T}}\boldsymbol{b} + \rho \, (\boldsymbol{z}_k - \boldsymbol{u}_k))$

Umbral suave $\boldsymbol{z}_{k+1} = S_{\lambda/\rho}(\boldsymbol{x}_{k+1} + \boldsymbol{u}_k)$

Variable dual $\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \boldsymbol{x}_{k+1} - \boldsymbol{z}_{k+1}$

Al almacenar los factores LU de $A^{\mathrm{T}}A + \rho I$, el primer paso se reduce a una sustitución hacia atrás. Boyd et al. observan que reemplazar $\|\|\boldsymbol{x}\|\|_1 = \sum |x_i|$ por $\|\|F\boldsymbol{x}\|\|_1 = \sum |x_{i+1} - x_i|$ convierte este ejemplo en *"total variation denoising"*. Su libro proporciona ejemplos excelentes y convincentes de ADMM.

Partición de matrices y algoritmos proximales

Una primera conexión de $A\boldsymbol{x} = \boldsymbol{b}$ con ADMM se obtiene al particionar esa matriz en $A = B+C$. En ejemplos clásicos B podría ser la parte diagonal de A o la parte triangular inferior de A (Partición de Jacobi o Gauss-Seidel). Esos métodos fueron mejorados por Douglas-Rachford y Peaceman-Rachford, quienes regularizaron sumando αI y alternando entre B y C :

$$\boxed{\begin{array}{l}(B+\alpha I)\boldsymbol{x}_{k+1} = \boldsymbol{b} + (\alpha I - C)\boldsymbol{z}_k \\ (C+\alpha I)\boldsymbol{z}_{k+1} = \boldsymbol{b} + (\alpha I - B)\boldsymbol{x}_{k+1}\end{array}} \tag{26}$$

Esta idea apareció en 1955, cuando los problemas eran lineales. Llevó a ideas no lineales más profundas : operadores proximales y operadores monótonos. Una característica especial es la aparición de fórmulas exactas para varios operadores proximales importantes, especialmente la minimización ℓ^1 que llevó a los umbrales suaves y contracción en el Ejemplo 2. Definiremos el operador **Prox** y lo conectaremos con ADMM.

En optimización, lo análogo a esas ecuaciones matriciales (26) corresponde a $\boldsymbol{b} = \boldsymbol{0}$ y $B = \boldsymbol{\nabla F_2}$. Aquí F_2 es la parte $\boldsymbol{\ell^2}$ y su gradiente $\boldsymbol{\nabla F_2}$ (*o su subgradiente* $\boldsymbol{\partial F_2}$) es sin duda lineal. Entonces $\boldsymbol{\partial F_2} + \alpha I$ se corresponde con $B + \alpha I$ y su inversa es un **"operador proximal"** :

$$\boxed{\textbf{Prox}_{\boldsymbol{F}}(\boldsymbol{v}) = \operatorname{argmin}\left(F(\boldsymbol{x}) + \tfrac{1}{2}||\boldsymbol{x}-\boldsymbol{v}||_2^2\right)} \tag{27}$$

Los factores clave para justificar este enfoque proximal (y la división en $F_1 + F_2$) son

1 El problema $\boldsymbol{\ell^2}$ se entiende muy bien y **es rápido de resolver**

2 El problema $\boldsymbol{\ell^1}$ a menudo tiene una **solución cerrada (por contracción).**

La combinación de los dos pasos, análoga a la optimización de las ecuaciones matriciales de dos pasos (26), es la "version proximal" de ADMM con factor de escala α :

ADMM

$$\boldsymbol{x}_{k+1} = (\boldsymbol{\partial F_2} + \boldsymbol{\alpha I})^{-1}(\boldsymbol{\alpha z_k} - \boldsymbol{\alpha u_k}) \tag{28a}$$

$$\boldsymbol{z}_{k+1} = (\boldsymbol{\partial F_1} + \boldsymbol{\alpha I})^{-1}(\boldsymbol{\alpha x_{k+1}} + \boldsymbol{\alpha u_k}) \tag{28b}$$

$$\boldsymbol{u}_{k+1} = \boldsymbol{u_k} + \boldsymbol{x_{k+1}} - \boldsymbol{z_{k+1}} \tag{28c}$$

En términos generales, los algoritmos proximales se aplican a la optimización convexa. Para los problemas adecuados son rápidos. Esto incluye problemas distribuidos: minimizar una suma de términos en paralelo (como para ADMM). Parikh y Boyd observan que **Prox** hace concesiones entre minimizar una función y no desplazarse demasiado. Lo comparan con el descenso por gradiente $\boldsymbol{x} - \alpha\boldsymbol{\nabla F(x)}$ donde α desempeña el papel del tamaño del paso. Los puntos fijos de **Prox** son los minimizadores de F.

He aquí dos referencias excelentes y un sitio web con código fuente :

P. Combettes y J.- C. Pesquet, *Proximal splitting methods in signal processing, en Fixed-Point Algorithms for Inverse Problems*, Springer (2011). arXiv : 0912.3522

N. Parikh y S. Boyd, *Proximal algorithms*, Foundations and Trends in Optimization **1** (2013) 123-321.

Libro y código : **http://stanford.edu/∼boyd/papers/pdf/prox_algs.pdf**

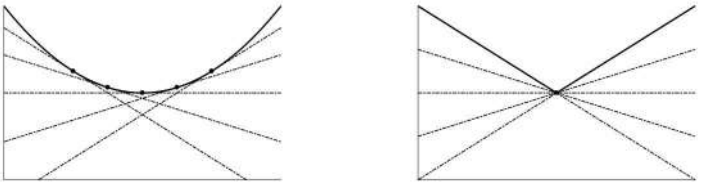


Figura III.3: Función suave $f(x)$: una tangente en cada punto con pendiente ∇f. Función no suave $f(x) = |x|$: muchas tangentes con pendientes ∂f (el subgradiente) en el vértice.

Distancia de Bregman

Aquí mencionamos a Bregman por primera vez en este libro. Pero la "distancia de Bregman" es un concepto cada vez más importante. Su característica inusual es la falta de simetría: $D(u,v) \neq D(v,u)$. Y la distancia de u a v depende de una función f. Tenemos $D \geq 0$ y también $D(w,v) \leq D(u,v)$ a lo largo del segmento $u < w < v$:

Distancia de Bregman D $$D_f(u,v) = f(u) - f(v) - (\nabla f(v), u - v) \tag{29}$$

Siempre se reemplaza el gradiente ∇f por un **subgradiente** ∂f en puntos donde el gráfico de $f(x)$ tiene un vértice. Entonces ∂f puede ser la pendiente de cualquier plano tangente que se mantenga debajo de la función convexa $f(x)$. Los ejemplos estándares en una dimensión son el valor absoluto $|x|$ y la función ReLU $\max(0,x) = \frac{1}{2}(x + |x|)$ con vértices en $x = 0$. Los planos tangentes a $|x|$ pueden tener pendientes $\partial|x|$ entre -1 y 1. Para ∂(ReLU) las pendientes están entre 0 y 1. Estos subgradientes hacen de ∂f un reemplazo efectivo (pero multivaluado) de ∇f.

Partición de Bregman y partición de Kaczmarz

Para minimización con un término de penalización ℓ^1 (como búsqueda de base o *basis pursuit*), necesitamos ver dos algoritmos iterativos. Comencemos por $Ax = b$:

Iteración de Bregman linealizada con parámetro $\lambda \geq 0$

$$y_{k+1} = y_k - s_k A^{\mathrm{T}}(Ax_k - b) \tag{30}$$

$$x_{k+1} = S_\lambda(y_{k+1}) = \text{sign}\,(y_{k+1})\ \text{máx}\,(|y_{k+1}| - \lambda, 0) \tag{31}$$

(30) es un paso normal ajustable para reducir $||Ax - b||^2$. El tamaño del paso es $s > 0$. Luego la *función de umbral suave* S se aplica a cada componente del vector y_{k+1}. Si $y_{k+1} = (1,-3)$ y $\lambda = 2$, el resultado x_{k+1} de esta función no lineal S en (31) será el vector x_{k+1} :

$$x_{k+1} = ((1)\ \text{máx}\,(1-2,0), (-1)\ \text{máx}\,(3-2,0)) = (0,-1).$$

Si $Ax = b$ tiene una solución y si $s\lambda_{\text{máx}}(A^{\mathrm{T}}A) < 1$, los vectores de Bregman x_k convergerán al vector óptimo x^* :

$$x^* \ \text{minimiza} \ \ \lambda||x||_1 + \frac{1}{2}||x||_2^2 \ \text{sujeto a} \ \ Ax = b. \tag{32}$$

La **iteración de Kaczmarz** es efectiva en *big data* con vectores ralos. Los pasos estándares resuelven $A\boldsymbol{x} = \boldsymbol{b}$. Ciclan secuencial o aleatoriamente por esas n ecuaciones $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{b}_i$ (donde $\boldsymbol{a}_i^{\mathrm{T}}$ es la fila i de A). Cada paso ajusta $\boldsymbol{x}_k$ por un múltiplo $c\boldsymbol{a}_i$ de una columna de A^{T}, para satisfacer la i-ésima ecuación $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}_{k+1} = b_i$:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \frac{b_i - \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}_k}{||\boldsymbol{a}_i||^2}\,\boldsymbol{a}_i \quad \text{resuelve} \quad \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}_{k+1} = \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}_k + \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{a}_i\frac{b_i - \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}_k}{||\boldsymbol{a}_i||^2} = b_i. \tag{33}$$

Los pasos de Kaczmarz ralos combinan (33) con el umbral suave de (31) :

$$\boxed{\textbf{Kaczmarz ralo} \quad \boldsymbol{y}_{k+1} = \boldsymbol{x}_k - s_k\boldsymbol{a}_i \quad \text{y} \quad \boldsymbol{x}_{k+1} = S(\boldsymbol{y}_{k+1}).} \tag{34}$$

Siempre tenemos libertad en el tamaño del paso s_k. La elección $s_k = (\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}_k - b_i)/||\boldsymbol{a}_i||^2$ es consistente con Kaczmarz. Lei y Zhou han propuesto un **algoritmo de aprendizaje online** para el mismo problema. Esto significa que el paso $k \to k+1$ se lleva a cabo tan pronto como la nueva observación b_k y la nueva ecuación $\boldsymbol{a}_k^{\mathrm{T}}\boldsymbol{x} = b_k$ se incorporan al cálculo. Luego el algoritmo de aprendizaje (34) calcula $\boldsymbol{y}_{k+1}$ y $\boldsymbol{x}_{k+1}$ de inmediato.

1 T. Goldstein y S. Osher, *The split Bregman method for L^1 regularized problems*, SIAM Journal of Imaging Sciences **2** (2009) 323 – 343.

2 W. Yin, S. Osher, D. Goldfarb, y J. Darbon , *Bregman iterative algorithms for ℓ^1 minimization with applications to compressed sensing*, SIAM J. Imaging Sciences **1** (2008) 143-168.

3 Y. Lei y D.-X. Zhou, *Learning theory of randomized sparse Kaczmarz method*, SIAM J. Imaging Sciences **11** (2018) 547–574.

Variación acotada : norma L^1 para el gradiente

Las imágenes naturales tienen *bordes*. En esos bordes, la función que las define $u(x, y)$ puede tener un salto. Su gradiente $\boldsymbol{\nabla u} = (\partial u/\partial x, \partial u/\partial y)$ puede ser suave en la dirección de los bordes, pero $\boldsymbol{\nabla u}$ **tiene una función delta** en la dirección perpendicular. La norma de la energía de u es infinita pero su norma de variación acotada es finita :

$$\boxed{||\boldsymbol{\nabla u}||_2^2 = \iint (u_x^2 + u_y^2)\,dx\,dy = \infty \quad \text{pero} \quad ||\boldsymbol{u}||_{\mathrm{BV}} = ||\nabla u||_1 = \iint \sqrt{u_x^2 + u_y^2}\,dx\,dy < \infty}$$

En una dimensión $u(x)$ podría ser una *función escalón* unitario. Su derivada es una *función delta* $\delta(x)$. La integral de $\delta(x)^2$ es infinita. Pero la integral de $\delta(x)$ es 1. No podemos trabajar con la norma $\boldsymbol{L^2}$ de la derivada $\delta(x)$, pero la norma $\boldsymbol{L^1}$ de $\delta(x)$ nos sirve.

En aplicaciones de eliminación de ruido en imágenes, esa norma BV es un término de penalización muy exitoso. Ajustamos los datos, y usamos la norma BV para evitar oscilaciones descontroladas. La imagen puede ser suave o suave a trozos, pero no puede tener ruido aleatorio "esporádico".

L. Rudin, S. Osher, y E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D **60** (1992) 259-268. (Este artículo fue una aplicación fundamental de la **BV**.)

Problemas de la Sección III.4

1 ¿Qué vector $\boldsymbol{x}$ minimiza $f(\boldsymbol{x}) = ||\boldsymbol{x}||^2$ sobre la recta $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{v} = 1$?

2 Siguiendo el Ejemplo 1 de esta sección, escriba el Problema 1 como una minimización de $f(\boldsymbol{x}) + g(\boldsymbol{z})$ con $\boldsymbol{x} = \boldsymbol{z}$. Describa esa función indicadora 0–1 $g(\boldsymbol{z})$, y realice un paso ADMM tomando como puntos de partida $\boldsymbol{x} = \boldsymbol{0}, \boldsymbol{z} = \boldsymbol{0}, \boldsymbol{u} = \boldsymbol{0}$.

3 ¿Qué vector $\boldsymbol{x}$ minimiza $\lambda||\boldsymbol{x}||_1$ sobre la recta $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{w} = 1$ para $\boldsymbol{w} = (2, 3)$?

4 Siguiendo el Ejemplo 2, realice un paso ADMM en el Problema 3 partiendo de $\boldsymbol{x} = (1, 1)$.

5 ¿Qué matrices $C \geq 0$ y $R \geq 0$ minimizan $||A - CR||_F^2$ si $A = \begin{bmatrix} 2 & 1 \\ -1 & 2 \end{bmatrix}$?

6 Siguiendo el Ejemplo 3 de esta sección, realice un paso ADMM en el Problema 5:

$$\text{A partir de } A = \begin{bmatrix} 2 & 1 \\ -1 & 2 \end{bmatrix} \quad R_0 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \quad U_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{calcule } X_1, C_1, R_1, U_1.$$

7 Halle el vector LASSO $\boldsymbol{x}$ que minimiza $\frac{1}{2}||A\boldsymbol{x} - \boldsymbol{b}||_2^2 + \lambda||\boldsymbol{x}||_1$ con

$$A = \begin{bmatrix} 4 & 1 \\ 0 & 1 \end{bmatrix} \qquad \boldsymbol{b} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \lambda = 2$$

8 Siguiendo el Ejemplo 4, realice un paso ADMM en el Problema 7 partiendo de $\boldsymbol{u}_0 = (1, 0) = \boldsymbol{z}_0$ con $\rho = 2$.

9 Halle $\mathbf{Prox}_F(\boldsymbol{v}) = \operatorname{argmin}\left(\frac{1}{2}||\boldsymbol{x}||^2 + \frac{1}{2}||\boldsymbol{x} - \boldsymbol{v}||^2\right)$ en la ecuación (27). Este es el operador proximal de $F(\boldsymbol{x}) = \frac{1}{2}||\boldsymbol{x}||^2$. Es una función de $\boldsymbol{v}$.

He aquí tres espacios funcionales (cada uno contenido en el siguiente) y tres ejemplos de $u(x, y)$:

$u(x,y)$ **suave**	**Lipschitz** (la pendiente puede saltar)	**Variación acotada** (u puede saltar)
Bol x^2+y^2	**Base plana** máx$(x^2+y^2-1, 0)$	**Base cilíndrica** (agrega un escalón sobre $r=1$)

Una linda "*fórmula de co-área*" expresa la norma BV de u como la integral de las longitudes de los conjuntos de nivel $L(t)$ donde $u(x, y) = t$. $\quad \iint ||\mathbf{grad}\ \boldsymbol{u}||\, \boldsymbol{dx\, dy} = \int (\mathbf{longitud\ de}\ \boldsymbol{L(t)})\, \boldsymbol{dt}.$

Ejemplo Calcule ambos lados de la fórmula de co-área para un bol: $u(x, y) = x^2 + y^2$.

En el lado izquierdo, $||\text{grad } u|| = ||(2x, 2y)|| = 2r$. El área del bol integra de 0 a R: $||u||_{\mathbf{BV}} = ||\text{grad } u||_{\mathbf{1}} = \iint (2r)\, r\, dr\, d\theta = 4\pi R^3/3$.

En el lado derecho, el conjunto de nivel donde $u = t$ es el círculo $x^2 + y^2 = t$ con longitud $2\pi\sqrt{t}$. La integral de esa longitud $L(t)$ de $t = 0$ a R^2 es $\int 2\pi\sqrt{t}\, dt = 4\pi R^3/3$.

10 ¿Qué es $||u||_{\mathbf{BV}}$ si $u(x, y) = x + y$ en el triángulo con lados $x = 0, y = 0, x + y = 1$? ¿Qué es $||u||_{\mathbf{BV}}$ si $u = 0$ en un cuadrado unitario y $u = 1$ afuera?

III.5. Compressed Sensing y completación de matrices

El principio básico detrás de compressed sensing es que una señal rala puede recuperarse con exactitud a partir de datos incompletos. Es una experiencia impresionante ver cómo emerge una imagen perfecta con muy pocas mediciones. La geometría de un diamante ℓ^1 comparada con la de una esfera ℓ^2 es nos ayuda explicarlo. El análisis matemático ha identificado las condiciones sobre A y el número mínimo de mediciones para lograrlo. Las resumiremos.

El lector entenderá que el teorema de Nyquist-Shannon sigue vigente. Para recuperar una señal ruidosa con exactitud debemos muestrear con al menos el doble de su frecuencia máxima. De lo contrario se pierde parte de esa señal. Pero las señales ralas no son ruidosas. Si una señal está expresada por un número pequeño de sinusoides o wavelets, entonces esa señal rala puede recuperarse (con probabilidad muy cercana a 1) con un número pequeño de mediciones.

Es importante recordar: **la raleza depende de la base $v_1, \ldots, v_n$ en la que la señal está representada**. Usa solo unos pocos v. Y la señal puede percibirse en una base $w_1, \ldots, w_n$ distinta. Los v son las columnas de una matriz representativa V, y los w son las columnas de una matriz adquirida W. Un ejemplo común: los v son una base de Fourier y los w son la base canónica (*spike*). Así V es la matriz de Fourier F y W es la matriz identidad I.

Un requisito clave para compressed sensing es la **"incoherencia" de V y W**: las componentes de $V^{\mathrm{T}}W$ son pequeños. Esos son los productos internos $v_i^{\mathrm{T}} w_j$. Por suerte tenemos baja coherencia entre F e I: todas las componentes de F son del mismo tamaño. Y la suerte no termina ahí: las matrices aleatorias con $||\text{columnas}|| = 1$ son casi seguramente incoherentes con cualquier base fija. Así la aleatoriedad y las probabilidades cercanas a 1 son ideas clave en compressed sensing.

El paso de sensing produce solo $m < n$ coeficientes no nulos y_k de la señal desconocida f: $y = W^{\mathrm{T}} f$. Para reconstruir $f^* = Vx^*$ cercana a f, usamos optimización ℓ^1 para encontrar x^*:

$$\text{Minimizar } ||x||_1 \text{ sujeto a } W^{\mathrm{T}} V x = y \tag{1}$$

Este es un problema de programación lineal (la búsqueda de base). Eso nos dice que podemos esperar una solución rala x^* (en un vértice del conjunto de vectores con $W^{\mathrm{T}} V x = y$). Entonces el método simplex es un algoritmo posible para encontrar x^*. Existen maneras más rápidas de resolver (1).

El teorema básico lo establecieron Candès, Tao y Donoho:

> Supongamos que V y W son incoherentes y x^* es ralo ($\leq S$ no nulos). Si $m > C\, S \log n$, la solución a (1) tiene probabilidad casi uno de reproducir f con exactitud.

Observe Una situación inusual surge cuando la verdadera señal es super rala en la base w. Tal vez es exactamente uno de los w. Entonces podríamos encontrar solo ceros y nos perderíamos por completo ese w en x^*. Esta es una de las razones por las que la probabilidad entra en compressed sensing.

El teorema básico mencionado no es la historia completa. El sistema tendrá ruido, por lo que estaremos recuperando un vector $\boldsymbol{x}$ que resuelve $A\boldsymbol{x} = \boldsymbol{b} + \boldsymbol{z}$, donde $\boldsymbol{z}$ puede ser estocástico y desconocido. El problema ralo es cercano, pero no coincide a la perfección con los datos con ruido. Queremos una **recuperación estable**: resolver (1) con datos con ruido obteniendo un $\boldsymbol{x}^*$ que esté **cerca del $\boldsymbol{x}^{**}$ ralo ($\leq S$ no nulos) como el que podría haberse obtenido de los datos sin ruido.**

Esta conclusión es cierta si A tiene la **"propiedad isométrica restringida"** (RIP) con $\delta < \sqrt{2} - 1$:

$$(\mathbf{RIP}) \qquad (1-\delta)\,||\boldsymbol{x}||_2^2 \leq ||A\boldsymbol{x}||_2^2 \leq (1+\delta)\,||\boldsymbol{x}||_2^2 \text{ si } \boldsymbol{x} \text{ es } S\text{-ralo.} \tag{2}$$

Afortunadamente, la matriz A puede tener columnas elegidas al azar en la esfera unitaria (o elegidas al azar a partir de la distribución normal $\boldsymbol{N}(0, 1/m)$) o con componentes aleatorias independientes iguales a $\pm 1/\sqrt{m}$, o de otras maneras aleatorias. El requisito RIP se cumplirá casi seguramente si $m > C\,S\log(n/S)$.

Esto finalmente conecta la cantidad de mediciones m con la raleza S. Y los datos con ruido nos llevan a reemplazar la búsqueda de base (donde $A\boldsymbol{x}$ es exactamente igual a $\boldsymbol{b}$) con LASSO:

$$\boxed{\textbf{LASSO con ruido} \qquad \text{Minimizar } ||\boldsymbol{x}||_1 \text{ sujeto a } ||A\boldsymbol{x} - \boldsymbol{b}||_2 \leq \epsilon} \tag{3}$$

Esta exposición de compressed sensing se basó en el artículo de Candès y Wakin: *IEEE Signal Processing Magazine* **21** (marzo 2008). Ese artículo se refiere a trabajos previos de Candès, Tao, y Donoho, que lograron un objetivo muy valioso. En lugar de conseguir cantidades masivas de datos y luego comprimirlos (como hace una cámara), solo se consiguen y utilizan $m = O(S \log n/S)$. *Una cámara de un pixel se vuelve posible, sin lente*, como lo demostró Rich Baraniuk. Tal vez las aplicaciones médicas son las más valiosas. Comenzaremos y terminaremos con ellas.

Comenzó con una observación inesperada de Candès en 2004. La imagen de prueba de Logan-Shepp (un modelo abstracto de la cabeza humana) fue corrompida con ruido. Parecía como si una resonancia magnética hubiera sido detenida antes de tiempo. Para mejorar la imagen, Candès probó una idea usando $\boldsymbol{\ell^1}$. Para su sorpresa, *la imagen fantasma quedó perfecta* (aunque los datos estuvieran incompletos). Fue casi como un equivalente al descubrimiento de los rayos X por Roentgen en 1895, un accidente que dio origen a una industria.

Sorprendentemente, la semana en que escribí estas palabras, apareció un ensayo de David Donoho en la versión digital de *Notices of the American Math Society* (enero 2018). Describe cómo compressed sensing ha acelerado las resonancias magnéticas. Los tiempos de escaneo fueron reducidos de 8 minutos a 70 segundos, con imágenes de alta calidad. Las imágenes dinámicas del corazón se hacen viables incluso en niños. Michael Lustig fue un pionero en el éxito de estas resonancias magnéticas, y una gran cantidad de matemáticos contribuyeron a la teoría y a los algoritmos. La FDA ha aprobado un cambio que se irá realizando gradualmente en los Estados Unidos, a medida que miles de equipos sean actualizados.

El objetivo del escrito de Donoho era mostrar que la financiación, la colaboración, la teoría y los experimentos (y un poco de suerte) han producido algo maravilloso.

Completación de matrices con la norma nuclear

El rango de una matriz es como el número de valores no nulos de un vector. De alguna manera el rango mide la raleza. Para un rango pequeño, la matriz Σ de valores singulares es literalmente rala (r no nulos). Así como el número de valores no nulos es una "norma 0" para vectores, **el rango es una "norma 0" para matrices**. Pero $||\boldsymbol{v}||_0$ **y** $||\boldsymbol{A}||_0$ **no son normas verdaderas de vectores y de matrices**, porque $||\boldsymbol{v}||_0 = ||2\boldsymbol{v}||_0$ y rango(A) = rango$(2A)$. Multiplicar por 2 no cambia la cantidad de valores no nulos ni el rango, pero sí duplica cualquier norma verdadera.

Sin embargo a menudo queremos raleza para los vectores y rango bajo para las matrices. El **problema de completación de matrices** comienza con componentes faltantes en una matriz A_0. Queremos completar A_0 a A, manteniendo el rango tan pequeño como sea posible. Estamos introduciendo un mínimo de datos inexplicados. Este problema de **datos faltantes** está muy extendido en todas las áreas de la ciencia observacional. Algunas hipótesis sobre la A completa son necesarias para llenar los espacios en blanco, y el *rango mínimo* es una opción natural. Observe estos ejemplos :

$$A_0 = \begin{bmatrix} \mathbf{1} & \mathbf{2} \\ * & * \end{bmatrix} \qquad B_0 = \begin{bmatrix} \mathbf{1} & * \\ * & \mathbf{4} \end{bmatrix} \qquad C_0 = \begin{bmatrix} \mathbf{1} & \mathbf{2} \\ \mathbf{3} & * \end{bmatrix}$$

Todas pueden completarse con rango 1. A_0 permite cualquier multiplo de $(1, 2)$ en la segunda fila. B_0 permite números cualesquiera b y c con $bc = 4$. C_0 permite solo 6 en su última componente.

Con vectores relajamos la norma $||\boldsymbol{v}||_0$ de raleza a la norma $||\boldsymbol{v}||_1$. Con matrices, **ahora relajamos norma del rango** $||\boldsymbol{A}||_0$ **a la norma nuclear** $||\boldsymbol{A}||_{\boldsymbol{N}}$. Esta norma nuclear es la norma $\boldsymbol{\ell^1}$ de la diagonal de Σ. **Estamos minimizando la suma de los valores singulares** :

$$\boxed{\textbf{Norma nuclear} \qquad ||\boldsymbol{A}||_{\boldsymbol{N}} = \boldsymbol{\sigma_1 + \sigma_2 + \cdots + \sigma_r}} \tag{4}$$

Ahora tenemos una norma convexa, pero no estrictamente convexa: $||B_1 + B_2||_N = ||B_1||_N + ||B_2||_N$ es posible en la "desigualdad" triangular. De hecho, la matriz ejemplo B_0 podría completarse simétricamente con cualquier $b = c$ entre -2 y 2. La norma nuclear $||B||_N$ se mantiene en **5** (valores singulares = autovalores y $||B||_N$ = traza para esta B semidefinida positiva). Las matrices de rango uno son a grandes rasgos análogas a los vértices del diamante $||\boldsymbol{x}||_1 = 1$.

Un ejemplo famoso de completación de matrices fue la *competencia de Netflix*. Las calificaciones de m películas por n espectadores se insertaron en A_0. Pero los clientes no habían visto todas las películas. Faltaban muchas calificaciones. Un *sistema de recomendaciones* las predijo. La norma nuclear da una buena solución que necesitó ajustarse a la psicología humana: la competencia de Netflix no era de matemática pura.

Otra aplicación es el cálculo de la matriz de covarianza a partir de numerosos datos de muestreo. Hallar todas las covarianzas σ_{ij} puede ser costoso (caso típico : las covarianzas de todas las acciones en un período de 365 días). Podemos calcular algunas y estimar el resto vía completación de matrices: Un ejemplo científico : todas las covarianzas al medir la superficie del océano en 10^4 posiciones.

He aquí la "relajación convexa" de la minimización de rango a minimización de norma nuclear :

$$\boxed{\textbf{Completación de matriz} \quad \textbf{Minimizar } ||\boldsymbol{A}||_{\boldsymbol{N}} \textbf{ sujeto a } \boldsymbol{A = A_0} \textbf{ en las componentes conocidas.}}$$

La pregunta matemática es : conociendo K componentes en una matriz de rango r de n por n, ¿ podemos esperar una recuperación perfecta de toda la matriz ? La soprendente respuesta es *sí*, siempre y cuando K sea suficientemente grande. Candès y Recht demostraron que si $K > C\, n^{5/4}\, r \log n$, entonces con probabilidad alta (en un modelo apropiado) la recuperación de A será perfecta.

He aquí algunos comentarios acerca de su prueba de 20 páginas, seguidos de referencias.

1. "Con probabilidad alta" no significa "con certeza". Debemos trabajar con un modelo para matrices A aleatorias. Una opción es tomar U y V como matrices ortogonales aleatorias en $A = U\Sigma V^{\mathrm{T}}$.

2. El análisis y la prueba usan versiones matriciales de desigualdades claves de estadística, desrrolladas en la sección V.3.

3. Encontrar A con la norma nuclear mínima puede expresarse como un **programa semidefinido** :

$$\text{Minimizar la traza de } \begin{bmatrix} W_1 & X \\ X & W_2 \end{bmatrix} \quad \begin{matrix} X \text{ contiene las componentes conocidas} \\ W_1, W_2 \text{ son semidefinidas positivas} \end{matrix}$$

1. D. Donoho, *Compressed sensing*, IEEE Trans. Inform. Th. **52** (2006) 1289-1306.

2. E. Candès, J. Romberg, y T. Tao, *Robust uncertainty principles : Exact signal reconstruction from highly incomplete Fourier information*, IEEE Transactions on Information Theory **52** (2006) 489-509.

3. E. Candès y B. Recht, *Exact matrix completion via convex optimization*, Foundations of Comp. Math. **9** (2009) 717 − 736; arXiv : 0805.4471$v1$, 29 May 2008.

4. T. Hastie, B. Mazumder, J. Lee, y R. Zadeh, Matrix completion and low-rank SVD via fast alternating least squares, arXiv : 1410.2596, 9 octubre 2014.

Algoritmos para completación de matrices

Necesitamos un algoritmo que complete la matriz A, con componentes fijas A_{conocida}. La referencia 4 antes mencionada muestra el desarrollo de tres iteraciones alternantes: cada nuevo método mejora el algoritmo previo. Así es como evoluciona el análisis numérico.

1 (*Mazumder et. al*) Umbral suave de la SVD de A_k (vea la Sección VI.5 para S_λ) :

$$A_k = U_k \Sigma_k V_k^{\mathrm{T}} \text{ y } B_k = U_k S_\lambda(\Sigma_k) V_k^{\mathrm{T}} \text{ con } S_\lambda(\sigma) = \text{máx}\,(\sigma - \lambda, 0)$$

S_λ reduce los valores singulares más pequeños de B_k a cero, reduciendo su rango. Así

$$A_{k+1} \text{ minimiza } \frac{1}{2}||(A - B_k)_{\text{conocida}}||_F^2 + \lambda ||B_k||_N \tag{5}$$

La desventaja está en tener que calcular la SVD de A_k en cada paso. En la competencia de Netflix, A tenía 8×10^9 componentes. Los autores lograron reducir este coste elevado. Y cada SVD previa sirve de pre-calentamiento para la siguiente. Y las ideas nuevas continúan apareciendo.

2 (*Srebro et. al*) Estos autores escriben la matriz solución como $CR^{\mathrm{T}} = (m \times r)(r \times n)$:

$$\begin{matrix} \textbf{Minimizar} \\ C \text{ y } R \end{matrix} \qquad \frac{1}{2}||(A - CR^{\mathrm{T}})_{\text{conocida}}||_F^2 + \frac{\lambda}{2}(||C||_F^2 + ||R||_F^2) \tag{6}$$

Esto es convexo en C y en R por separado (entonces, biconvexo). Un algortimo alternante resulta natural : *actualizar C y luego actualizar R.* Cada problema es una regresión ridge ℓ^2 en cada columna de C y luego en cada columna de R. Esto produce una factorización CR de "margen máximo".

3 (*Hastie et. al*, vea **4** mencionado) El tercer algoritmo se propuso en 2014. Es una variación del primero, pero alterna, como en el segundo, entre C y R. La minimización ahora incluye todas las componentes de $A - CR^{\mathrm{T}}$, no solo aquellos de las posiciones conocidas. Esto simplifica enormemente el problema de mínimos cuadrados, trabajar con columnas completas de componentes conocidas y desconocidas.

Comencemos con el problema de mínimos cuadrados (alternante) con A *completamente conocida* :

$$\begin{array}{c}\textbf{Minimizar}\\ C \text{ y } R\end{array} \qquad \frac{1}{2}||A - CR^{\mathrm{T}}||_F^2 + \frac{\lambda}{2}(||C||_F^2 + ||R||_F^2) \tag{7}$$

Una solución explícita es $C = U_r S_\lambda(\Sigma_r)^{1/2}$ y $R = V_r S_\lambda(\Sigma_r)^{1/2}$. Todas las soluciones, incluida esta, satisfacen $CR^{\mathrm{T}} = U_r S_\lambda(\Sigma_r) V_r^{\mathrm{T}} =$ SVD suave de A. El umbral suave S_λ muestra el efecto del término de penalización en (7). Maravillosamente, ese producto $B = CR^{\mathrm{T}}$ resuelve este problema de norma nuclear de rango r :

$$\begin{array}{c}\textbf{Minimizar}\\ \text{rango}(B) \le r\end{array} \qquad \frac{1}{2}||A - B||_F^2 + \lambda||B||_N \tag{8}$$

Esta conexión con (7) da el siguiente cálculo alternante rápido de $B = CR^{\mathrm{T}}$.

Por el momento suponemos que conocemos todas las componentes de A: no falta ninguna. Comencemos con $C = m \times r$ aleatorio y $D = I_r$. Cada paso actualiza $R, C,$ y D.

1. Minimizar sobre R $(n \times r)$ $\quad ||A - CR^{\mathrm{T}}||_F^2 + \lambda||R||_F^2$	(9)
2. Calcular esta SVD $\quad RD = U\Sigma V^{\mathrm{T}}$ $\quad$ Tomar $D = \sqrt{\Sigma}$ y $R_{\text{nueva}} = VD$	
3. Minimizar sobre C $(m \times r)$ $\quad ||A - CR^{\mathrm{T}}||_F^2 + \lambda||C||_F^2$	(10)
4. Calcular esta SVD $\quad CD = U\Sigma V^{\mathrm{T}}$ $\quad$ Tomar $D = \sqrt{\Sigma}$ y $C_{\text{nueva}} = UD$	

Repetimos esos pasos hasta que CR^{T} converja a la solución B de (8).

Ahora Hastie et. al vuelven al problema real, donde a A le faltan componentes : son desconocidas. En cada iteración, se toman de la matriz CR^{T}. Esto produce una muy eficiente representación rala de rango pequeño de A.

$$A = A_{\text{conocida}} + (CR^{\mathrm{T}})_{\text{desconocida}} = (A - CR^{\mathrm{T}})_{\text{conocida}} + CR^{\mathrm{T}}. \tag{11}$$

El algoritmo final es una pequeña modificación de (9)-(10), con A de la ecuación (11). Las soluciones de (9)-(10) tienen formas muy sencillas, porque esos problemas son en esencia regresiones ridge (mínimos cuadrados) :

$$R^{\mathrm{T}} = (D^2 + \lambda I)^{-1} DU^{\mathrm{T}} A \text{ en (9)} \qquad C = AVD(D^2 + \lambda I)^{-1} \text{ en (10)}$$

En conjunto se describe el algoritmo como **mínimos cuadrados alternantes con imputación suave** (soft-impute alternating least squares en inglés). Hastie et al analizan su convergencia y estabilidad numérica. Ellos prueban con una implementación sobre los datos para la competencia de Netflix con resultados exitosos.

Problemas de la Sección III.5

1 Para uno o más de estos ejemplos, ¿ puede encontrar la completación que minimice $||A||_{\boldsymbol{N}}$?

$$A_0 = \begin{bmatrix} 1 & 2 \\ * & * \end{bmatrix} \qquad B_0 = \begin{bmatrix} 1 & * \\ * & 4 \end{bmatrix} \qquad C_0 = \begin{bmatrix} 1 & 2 \\ 3 & * \end{bmatrix}$$

2 En cuanto a la importante Figura I.16 al comienzo de la Sección I.11, ¿ puede encontrar la matriz con la menor "*norma suma*" $||A||_{\boldsymbol{S}} = |a| + |b| + |c| + |d|$ tal que

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} ?$$

3 Para matrices de 2 por 2, ¿ cómo describiría la bola unitaria $||A||_{\boldsymbol{S}} \leq 1$ en esa norma suma $|a| + |b| + |c| + |d|$?

4 ¿ Puede encontrar las desigualdades que conectan la norma suma con la norma nuclear para matrices de n por n ?

$$||A||_{\boldsymbol{N}} \leq c(n)\,||A||_{\boldsymbol{S}} \quad \text{y} \quad ||A||_{\boldsymbol{S}} \leq d(n)\,||A||_{\boldsymbol{N}}$$

5 Si solo una componente de una matriz A es desconocida, ¿ cómo completaría A para minimizar $||A||_{\boldsymbol{S}}$ o $||A||_{\boldsymbol{N}}$?

He aquí dos lindas fórmulas para la norma nuclear (gracias a Yuji Nakatsukasa).

$$\boxed{||A||_N = \min_{UV=A} ||U||_F\,||V||_F = \min_{UV=A} \frac{1}{2}||U||_F^2 + \frac{1}{2}||V||_F^2} \qquad (*)$$

6 Comience con $A = U\Sigma V^{\mathrm{T}} = (U\Sigma^{1/2})\,(\Sigma^{1/2}V^{\mathrm{T}})$. Si renombra esos dos factores como U^* y V^*, de manera que $A = U^*V^*$, muestre que $||U^*||_F^2 = ||V^*||_F^2 = ||A||_N$: realizan el mínimo en $(*)$.

7 Si A es semidefinida positiva entonces $||A||_N =$ traza de A (¿ Por qué ?). Luego, si $A = UV$, explique cómo la desigualdad de Cauchy-Schwarz da

$$||A||_N = \text{traza}\,(UV) = \Sigma\,\Sigma\,U_{ij}V_{ji} \leq ||U||_F\,||V||_F.$$

Parte IV

Matrices especiales

IV.1 **Transformada de Fourier: discreta y continua**

IV.2 **Matrices de desplazamiento y matrices circulantes**

IV.3 **El producto de Kronecker $A \otimes B$**

IV.4 **Transformadas de senos y cosenos a partir de sumas de Kronecker**

IV.5 **Matrices de Toeplitz y filtros invariantes de desplazamiento**

IV.6 **Grafos y laplacianos y las leyes de Kirchhoff**

IV.7 **Conglomerados con métodos espectrales y k-medias**

IV.8 **Completando matrices de rango uno**

IV.9 **El problema ortogonal de Procrustes**

IV.10 **Matrices de distancia**

Parte IV : Matrices especiales

Este capítulo desarrolla dos grandes temas : las matrices claves para la **transformada de Fourier discreta** y para los **modelos gráficos**. Ambos temas están presentes en aprendizaje automático cuando el problema tiene una estructura especial, que se refleja en la arquitectura de la red neuronal.

Fourier es para problemas que presentan *invariancia con respecto a desplazamientos*. Las operaciones en un píxel de una imagen son las mismas que las operaciones en el siguiente píxel. En ese caso, la operación es una *convolución*. Cada fila de la matriz subyacente es un desplazamiento de la fila anterior. Cada columna es un desplazamiento de la columna anterior. Las redes convolucionales usan los mismos pesos alrededor de cada píxel.

Con esa estructura de invariancia por desplazamientos, una matriz de convolución (un "*filtro*" en procesamiento de imágenes) tiene diagonales constantes. La matriz de N por N en un problema unidimensional está determinada completamente por su primera fila y su primera columna. A menudo la matriz es de banda: el filtro tiene una longitud finita, y la matriz tiene ceros fuera de una banda de diagonales. El ahorro es enorme en problemas bidimensionales con bloques de una dimensión repetidos.

Esto permite el uso de una red neuronal convolucional (**CNN, por su sigla en inglés** o **ConvNet**) cuando sería imposible usar una matriz de N^2 por N^2 llena de pesos independientes. Un gran avance en la historia de la CNN fue este artículo de NIPS de 2012 :

Alex Krizhevsky, Ilya Sutskever, y Geoffrey Hinton : *ImageNet Classification with Deep Convolutional Neural Networks*. Su red neuronal tenía 60 millones de parámetros.

Los grafos tienen una estructura muy diferente, consisten en n nodos conectados por m aristas. La matriz de incidencia A expresa estas conexiones : m filas para las aristas y n columnas para los nodos. Cada grafo está asociado a cuatro matrices importantes :

Matriz de incidencia A	de m por n con -1 y 1 en cada fila
Matriz de adyacencia M	de n por n con $a_{ij} = 1$ cuando los nodos i y j están conectados
Matriz de grado D	matriz diagonal de n por n con las sumas de las filas de M
Matriz Laplaciana $L = A^{\mathrm{T}}A = D - M$	matriz semidefinida positiva

Desde el punto de vista del aprendizaje profundo, los problemas de Fourier están asociados a las CNN, y los modelos gráficos llevan a redes gráficas. La teoría de grafos se ha convertido en nuestra herramienta más valiosa para entender los problemas discretos sobre redes.

IV.1. Transformadas de Fourier : discreta y continua

La transformada de Fourier clásica se aplica a funciones. La transformada de Fourier discreta (DFT) se aplica a vectores :

Serie de Fourier real : funciones periódicas reales $f(x+2\pi) = f(x)$

Serie de Fourier compleja : funciones periódicas complejas

Transformada de Fourier integral : funciones $f(x)$ complejas con $-\infty < x < \infty$

Serie de Fourier discreta : vectores complejos $\boldsymbol{f} = (f_0, f_1, \ldots, f_{N-1})$

Nos enfocaremos en la última: transformar vectores en vectores. Esto se logra con una matriz de N por N. La transformada inversa usa la matriz inversa. Todas estas transformadas comparten **funciones base** del mismo tipo que usó Fourier :

Serie de Fourier real : cosenos $\cos \boldsymbol{nx}$ y senos $\operatorname{sen} \boldsymbol{nx}$

Serie de Fourier compleja : exponenciales complejas $\boldsymbol{e^{inx}}$ con $n = 0, \pm 1, \pm 2, \ldots$

Transformada integral de Fourier : exponenciales complejas $\boldsymbol{e^{ikx}}$ para $-\infty < k < \infty$

Serie de Fourier discreta : N vectores base $\boldsymbol{b}_k$ con $(\boldsymbol{b}_k)_{\boldsymbol{j}} = \boldsymbol{e^{2\pi ijk/N}} = \boldsymbol{(e^{2\pi i/N})^{jk}}$.

Cada función y cada vector se expresan como una combinación de funciones base de Fourier. **¿ Qué es la "transformada" ?** Es la regla que conecta f a sus coeficientes $a_k, b_k, c_k, \widehat{f}(k)$ en estas combinaciones de funciones base :

Serie real	$\boldsymbol{f(x)} = a_0 + a_1 \cos x + b_1 \operatorname{sen} x + a_2 \cos 2x + b_2 \operatorname{sen} 2x + \cdots$
Serie compleja	$\boldsymbol{f(x)} = c_0 + c_1 e^{ix} + c_{-1} e^{-ix} + c_2 e^{2ix} + c_{-2} e^{-2ix} + \cdots$
Integral de Fourier	$\boldsymbol{f(x)} = \displaystyle\int_{-\infty}^{\infty} \widehat{f}(k) e^{ikx}\, dk$
Serie discreta	$\boldsymbol{f} = c_0 \boldsymbol{b}_0 + c_1 \boldsymbol{b}_1 + \cdots + c_{N-1} \boldsymbol{b}_{N-1} =$ **matriz de Fourier** $\boldsymbol{F}$ **por** $\boldsymbol{c}$

Cada **transformada de Fourier** lleva $\boldsymbol{f}$ del "espacio x" a sus coeficientes en el "espacio de frecuencias".

La **transformada inversa** parte de los coeficientes y reconstruye la función original.

Calcular coeficientes es *análisis*. Reconstruir la $\boldsymbol{f}$ original (como se ve en el recuadro) es *síntesis*. Para vectores, los comandos `fft` e `ifft` producen $\boldsymbol{c}$ a partir de $\boldsymbol{f}$ y $\boldsymbol{f}$ a partir de $\boldsymbol{c}$. Esos comandos son ejecutados por la **transformada rápida de Fourier** .

Ortogonalidad

En las cuatro transformadas, los coeficientes tienen lindas fórmulas. Esto es porque **las funciones base son ortogonales**. La ortogonalidad es la clave de todas las transformadas famosas (a menudo la base contiene las autofunciones de un operador simétrico). Esto nos permite hallar cada número c_k por separado. Para vectores, solo tomamos el producto interno (complejo) $(\boldsymbol{b}_k, \boldsymbol{f})$ entre $\boldsymbol{f}$ y cada uno de los vectores base ortogonales $\boldsymbol{b}_k$. He aquí ese paso clave :

$$(\boldsymbol{b}_k, \boldsymbol{f}) = (\boldsymbol{b}_k, c_0\boldsymbol{b}_0 + c_1\boldsymbol{b}_1 + \cdots + c_{N-1}\boldsymbol{b}_{N-1}) = c_k(\boldsymbol{b}_k, \boldsymbol{b}_k). \quad \textbf{Así} \quad \boxed{c_k = \frac{(\boldsymbol{b}_k, \boldsymbol{f})}{(\boldsymbol{b}_k, \boldsymbol{b}_k)}}. \tag{1}$$

Todos los productos internos $(\boldsymbol{b}_k, \boldsymbol{b}_j)$ son cero en la ecuación (1), excepto $(\boldsymbol{b}_k, \boldsymbol{b}_k)$. Es un placer calcular esos denominadores $(\boldsymbol{b}_k, \boldsymbol{b}_k) = ||\boldsymbol{b}_k||^2 = \boldsymbol{\pi}$ o $\boldsymbol{2\pi}$ o $\boldsymbol{N}$ para las funciones base de Fourier :

Serie real : $\int (\cos nx)^2\, dx = \boldsymbol{\pi}$ y $\int (\operatorname{sen} nx)^2\, dx = \boldsymbol{\pi}$ y $\int (1)^2 dx = \boldsymbol{2\pi}$

Serie compleja : $\int e^{ikx} e^{-ikx} dx = \int 1\, dx = \boldsymbol{2\pi}$

Serie discreta : $\overline{\boldsymbol{b}}_k^{\mathrm{T}} \boldsymbol{b}_k = 1 \cdot 1 + e^{2\pi ik/N} \cdot e^{-2\pi ik/N} + e^{4\pi ik/N} \cdot e^{-4\pi ik/N} + \cdots = \boldsymbol{N}$

Si normalizamos los vectores base a $||\boldsymbol{b}_k|| = 1$ entonces las matrices de Fourier son ortogonales.

El caso de la integral de Fourier es sutil y nos detenemos luego de escribir la ecuación (2). Sus integrales infinitas provienen de las series de Fourier cuando el periodo 2π aumenta a $2\pi T$ con $T \to \infty$:

$$\boldsymbol{\widehat{f}(k)} = \int_{x=-\infty}^{\infty} f(x) e^{-ikx}\, dx \quad \text{y} \quad \boldsymbol{f(x)} = \frac{1}{2\pi} \int_{k=-\infty}^{\infty} \widehat{f}(k) e^{ikx}\, dk. \tag{2}$$

Matriz de Fourier F y matriz DFT Ω

Las matrices F_N y Ω_N son de N por N. Ambas son simétricas (pero complejas). Tienen las mismas columnas, pero su orden es distinto. F_N contiene potencias de $\boldsymbol{w = e^{2\pi i/N}}$, y Ω_N contiene potencias del conjugado complejo $\overline{\boldsymbol{w}} = \boldsymbol{\omega = e^{-2\pi i/N}}$.

$\boldsymbol{w}$ **romana y** $\boldsymbol{\omega}$ **griega**. De hecho, las dos matrices son complejas conjugadas : $\boldsymbol{\overline{F}_N = \Omega_N}$. Aquí tenemos a F_4 y Ω_4, que contienen potencias de $\boldsymbol{w} = e^{2\pi i/4} = \boldsymbol{i}$ y $\boldsymbol{\omega} = e^{-2\pi i/4} = \boldsymbol{-i}$. Contamos filas y columnas a partir de cero, por lo tanto F y Ω tienen filas $0, 1, 2, 3$:

$$\boxed{\textbf{matriz de Fourier} \quad \boldsymbol{F_4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & i^2 & i^3 \\ 1 & i^2 & i^4 & i^6 \\ 1 & i^3 & i^6 & i^9 \end{bmatrix} \qquad \textbf{matriz DFT} \quad \boldsymbol{\Omega_4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & (-i)^2 & (-i)^3 \\ 1 & (-i)^2 & (-i)^4 & (-i)^6 \\ 1 & (-i)^3 & (-i)^6 & (-i)^9 \end{bmatrix}} \tag{3}$$

$\boldsymbol{\Omega}$ por $\boldsymbol{f}$ produce los coeficientes de Fourier discretos $\boldsymbol{c}$
$\boldsymbol{F}$ por $\boldsymbol{c}$ trae de vuelta el vector $4\boldsymbol{f} = N\boldsymbol{f}$ porque $F\Omega = NI$

Si multiplicamos F_N por Ω_N, descubriremos una identidad fundamental para las transformadas de Fourier discretas. *Se revela la transformada inversa* :

$$\boxed{\boldsymbol{F_N\,\Omega_N = NI} \quad \text{y por lo tanto} \quad \boldsymbol{F_N^{-1} = \frac{1}{N}\,\Omega_N = \frac{1}{N}\,\overline{F}_N}} \tag{4}$$

Para ver esto, mire de cerca los N números $1, w, \ldots, w^{N-1}$ en la segunda columna de F. Todos esos números tienen módulo 1. Son equidistantes alrededor del círculo unidad en el plano complejo (Figura IV.1). Son las N soluciones de la ecuación de grado N, $z^N = 1$.

Sus complejos conjugados $1, \omega, \ldots, \omega^{N-1}$ son **los mismos N números**. Estas potencias de ω dan la vuelta al círculo unitario en sentido opuesto. La Figura IV.1 muestra 8 potencias de $w = e^{2\pi i/8}$ y $\omega = e^{-2\pi i/8}$. **Sus ángulos son 45 ° y −45 °.**

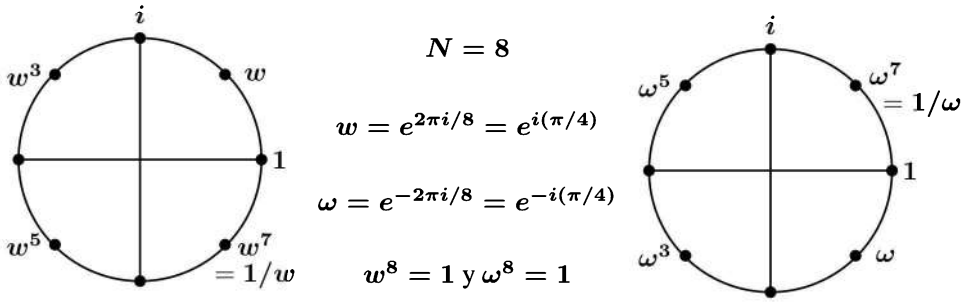


Figura IV.1: las potencias de w también son las potencias de ω. Son las N soluciones de $z^N = 1$.

Para probar que $F_N\Omega_N = NI$, he aquí la propiedad que necesitamos: ¡ los N puntos suman cero !

$$\textbf{Para cada } \boldsymbol{N}\textbf{, la suma } \boldsymbol{S = 1 + w + w^2 + \cdots + w^{N-1}} \textbf{ es cero.} \tag{5}$$

Para probarlo se multiplica S por w. Esto produce $w + \cdots + w^N$. Es lo mismo que S, porque w^N al final es lo mismo que 1 (al inicio). Así $Sw = S$. Entonces S debe ser cero.

Se ve de inmediato que los ocho números alrededor del círculo en la Figura IV.1 suman cero, porque los pares opuestos ya suman cero. Para N impar este apareamiento no funciona.

El hecho que $S = 0$ nos dice que cada elemento por fuera de la diagonal en $F_N\Omega_N$ es cero. Los elementos diagonales (suma de unos) son sin duda N. Así $\boldsymbol{F_N\,\Omega_N = NI}$. Si dividimos F y Ω por $\sqrt{N}$, entonces las dos matrices son inversas y también conjugadas complejas : son *unitarias*.

$$\textbf{Matrices unitarias} \quad \left(\frac{1}{\sqrt{N}}\,\boldsymbol{F_N}\right)\left(\frac{1}{\sqrt{N}}\,\boldsymbol{\Omega_N}\right) = \left(\frac{1}{\sqrt{N}}\,\boldsymbol{F_N}\right)\left(\frac{1}{\sqrt{N}}\,\overline{\boldsymbol{F}}_{\boldsymbol{N}}\right) = I. \tag{6}$$

Las matrices unitarias son la versión compleja de las matrices ortogonales. En lugar de $Q^{\mathrm{T}} = Q^{-1}$ tenemos $\overline{Q}^{\mathrm{T}} = Q^{-1}$. Es importante tomar conjugadas complejas cuando se traspone una matriz compleja (y la mayoría de los códigos de álgebra lineal lo hacen automáticamente). De la misma manera, la versión compleja de una matriz simétrica real es una **matriz hermitiana** con $\overline{S}^{\mathrm{T}} = S$.

La matriz DFT Ω es una permutación de la matriz de Fourier F

Las matrices F y Ω tienen las mismas columnas. Así F y Ω están conectadas por una matriz de permutación. Esa permutación P no afecta la columna correspondiente a la potencia cero : la primera columna de unos en ambas matrices. Luego P intercambia la siguiente columna $(1, w, w^2, \ldots, w^{N-1})$ de F por su última columna $(1, \omega, \omega^2, \ldots, \omega^{N-1})$. Luego del $\mathbf{1}$ en la primera fila y columna, P contiene la matriz identidad reversa J (con 1 en la antidiagonal) :

$$\boldsymbol{P} = \begin{bmatrix} \mathbf{1} & 0 \\ 0 & \boldsymbol{J} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & \mathbf{1} & 0 & 0 \end{bmatrix} \qquad \boldsymbol{P^2 = I} \text{ y } \boldsymbol{\Omega = FP} \text{ y } \boldsymbol{\Omega P = FP^2 = F}$$

He aquí las matrices completas para $\boldsymbol{\Omega = FP}$ cuando $N = 4$:

$$\boldsymbol{\Omega} = \begin{bmatrix} 1 & \mathbf{1} & 1 & 1 \\ 1 & \boldsymbol{-i} & (-i)^2 & (-i)^3 \\ 1 & \boldsymbol{(-i)^2} & (-i)^4 & (-i)^6 \\ 1 & \boldsymbol{(-i)^3} & (-i)^6 & (-i)^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \mathbf{1} \\ 1 & i & i^2 & \boldsymbol{i^3} \\ 1 & i^2 & i^4 & \boldsymbol{i^6} \\ 1 & i^3 & i^6 & \boldsymbol{i^9} \end{bmatrix} \begin{bmatrix} \mathbf{1} & & & \\ & & & \mathbf{1} \\ & & \mathbf{1} & \\ & \mathbf{1} & & \end{bmatrix} \text{ ya que } \begin{array}{rl} 1 & = 1 \\ -i & = i^3 \\ (-i)^2 & = i^2 \\ (-i)^3 & = i \end{array}$$

Estas identidades de matrices llevan al hecho increíble de que $\boldsymbol{F^4 = \Omega^4 = N^2 I}$. Cuatro transformadas devuelven el vector original (multiplicado por N^2). Solo combine $F\Omega = NI$ y $FP = \Omega$ con $P^2 = I$:

$$\boxed{\boldsymbol{F^2 P = F\Omega = NI} \quad \text{entonces} \quad \boldsymbol{PF^2 = NI} \quad \text{y} \quad \boldsymbol{F^4 = F^2 PPF^2 = N^2 I.}}$$

A partir de $F^4 = N^2 I$, se sigue que la matriz de Fourier F y la matriz DFT Ω solo tienen **cuatro autovalores posibles**. Son los números $\lambda = \sqrt{N}, i\sqrt{N}, -\sqrt{N}$ y $-i\sqrt{N}$ que resuelven $\lambda^4 = N^2$. Para tamaño $N > 4$ debe haber y hay λ repetidos. Los autovectores de F no son tan fáciles de encontrar.

La Transformada de Fourier Discreta

Comenzamos con un vector N-dimensional $\boldsymbol{f} = (f_0, \ldots, f_{N-1})$. La Transformada de Fourier Discreta expresa $\boldsymbol{f}$ como una combinación $c_0\boldsymbol{b}_0 + c_1\boldsymbol{b}_1 + \cdots + c_{N-1}\boldsymbol{b}_{N-1}$ de los vectores base de Fourier. Esos vectores base son las columnas $\boldsymbol{b}$ (que contienen potencias de w) en la matriz de Fourier F_N :

$$\boxed{\begin{bmatrix} f_0 \\ \vdots \\ f_{N-1} \end{bmatrix} = \begin{bmatrix} & & \\ \boldsymbol{b}_0 & \cdots & \boldsymbol{b}_{N-1} \\ & & \end{bmatrix} \begin{bmatrix} c_0 \\ \vdots \\ c_{N-1} \end{bmatrix} \qquad \begin{array}{l} \boldsymbol{f} = F_N \boldsymbol{c} \\ \boldsymbol{c} = F_N^{-1} \boldsymbol{f} \\ \boldsymbol{c} = \dfrac{1}{N} \Omega_N \boldsymbol{f} \end{array}} \tag{7}$$

La transformada directa $\boldsymbol{c} = \textbf{fft}\,(\boldsymbol{f})$ multiplica $\boldsymbol{f}$ por la matriz DFT Ω (y divide por N). Ese es el paso de análisis, separar $\boldsymbol{f}$ en N piezas ortogonales. La DFT encuentra los coeficientes en una serie de Fourier finita $\boldsymbol{f} = c_0\boldsymbol{b}_0 + \cdots + c_{N-1}\boldsymbol{b}_{N-1}$.

El paso de síntesis es la transformada inversa $\boldsymbol{f} = \textbf{ifft}\,(\boldsymbol{c}) = F\boldsymbol{c}$. Comienza con esos coeficientes $\boldsymbol{c} = (c_0, \ldots, c_{N-1})$. Realiza la multiplicación matriz-vector $F\boldsymbol{c}$ para recuperar $\boldsymbol{f}$.

Así $\textbf{ifft}\,(\textbf{fft}\,(\boldsymbol{f})) = \boldsymbol{f}$. Los ejemplos esclarecerán quiénes son los dos vectores $\boldsymbol{f}$ y $\boldsymbol{c}$.

Ejemplo 1 La transformada de $\boldsymbol{f} = (\mathbf{1}, \mathbf{0}, \ldots, \mathbf{0})$ es $\boldsymbol{c} = \frac{\mathbf{1}}{\boldsymbol{N}}(\mathbf{1}, \mathbf{1}, \ldots, \mathbf{1})$.

Ese vector $\boldsymbol{f}$ con una punta es una función delta discreta. Está concentrada en un punto. Su transformada $\boldsymbol{c}$ se reparte sobre todas las frecuencias. Multiplicar $\Omega\boldsymbol{f}$ selecciona la primera columna de Ω. Por lo tanto $\boldsymbol{c}$ muestra el mismo coeficiente de Fourier $1/N$ para todas las frecuencias. Aquí $N = 4$:

$$\boldsymbol{c} = \frac{1}{4}\Omega\,\boldsymbol{f} = \frac{1}{4}\begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \frac{\mathbf{1}}{\mathbf{4}}\begin{bmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix} \qquad \boldsymbol{f} = F\boldsymbol{c} = \frac{1}{4}\left(\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ i \\ i^2 \\ i^3 \end{bmatrix} + \begin{bmatrix} 1 \\ i^2 \\ i^4 \\ i^6 \end{bmatrix} + \begin{bmatrix} 1 \\ i^3 \\ i^6 \\ i^9 \end{bmatrix}\right) = \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}.$$

Esta transformada $\boldsymbol{c}$ de un "vector delta" $\boldsymbol{f}$ es como la transformada continua de una "función delta". La función delta concentra todo en $x = 0$. Su transformada de Fourier resulta una función constante:

$$\boldsymbol{\delta(x)} = \sum_{-\infty}^{\infty} c_k e^{ikx} \text{ tiene } \boldsymbol{c_k} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \delta(x) e^{-ikx}dx = \frac{\mathbf{1}}{\boldsymbol{2\pi}} \textbf{ para toda frecuencia } \boldsymbol{k}. \tag{8}$$

Se puede ver cómo 2π en la transformada continua juega el rol de N en la transformada discreta.

Ejemplo 2 $\boldsymbol{f} = (1, 1, \ldots, 1)$ se transformará nuevamente al vector delta $\boldsymbol{c} = (N, 0, \ldots, 0)$.

Ejemplo 3 Un desplazamiento en el vector delta a $\boldsymbol{f} = (0, 1, 0, \ldots, 0)$ produce una "modulación" en su transformada. Este $\boldsymbol{f}$ desplazado elige la siguiente columna $(1, \omega, \ldots, \omega^{N-1})$ de F:

$$\boldsymbol{c} = \frac{1}{N}\,\Omega_N\,\boldsymbol{f} = \frac{1}{N}\begin{bmatrix} 1 \\ \omega \\ \vdots \\ \omega^{N-1} \end{bmatrix} \text{ y } \boldsymbol{f} = F\boldsymbol{c} = \frac{1}{N}\begin{bmatrix} 1 & 1 & \cdot\cdot & 1 \\ 1 & w & \cdot\cdot & w^{N-1} \\ \cdot & \cdot & \cdot\cdot & \cdot \\ 1 & w^{N-1} & \cdot\cdot & w^{(N-1)^2} \end{bmatrix}\begin{bmatrix} 1 \\ \omega \\ \vdots \\ \omega^{N-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}. \tag{9}$$

La serie de Fourier tendrá un desplazamiento de $f(x)$ a $f(x - s)$. Se multiplica cada coeficiente c_k por e^{-iks}. Esto es exactamente como las multiplicaciones de $(1, 1, \ldots, 1)$ en el Ejemplo 1 para producir $(1, \omega, \ldots, \omega^{N-1})$ en el Ejemplo 3.

Regla de desplazamiento	**Un desplazamiento en el espacio x es una multiplicación de los coeficientes en el espacio k.**

Un paso de la transformada rápida de Fourier

Queremos multiplicar F por $\boldsymbol{c}$ lo más rápido posible. Normalmente, multiplicar una matriz por un vector conlleva N^2 multiplicaciones separadas, una por cada elemento de la matriz. Usted creerá que no hay otra manera mejor. (Si la matriz tiene ceros, entonces podemos saltear algunas multiplicaciones. Pero la matriz de Fourier no tiene ceros). Al usar los patrones especiales ω^{jk} y w^{jk} de sus elementos, Ω y F pueden factorizarse de una manera que produce muchos ceros. Esta es la transformada rápida de Fourier **FFT** (Fast Fourier Transform).

La idea clave es conectar $\boldsymbol{F_N}$ con la matriz de Fourier $\boldsymbol{F_{N/2}}$ que tiene la mitad de tamaño. Supongamos que N es una potencia de 2 (por ejemplo: $N = 2^{10} = 1024$). $\boldsymbol{F_{1024}}$ **se conecta con** $\boldsymbol{F_{512}}$.

Cuando $N = 4$, la clave está en la relación entre F_4 y dos copias de F_2 :

$$\boldsymbol{F_4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & i^2 & i^3 \\ 1 & i^2 & i^4 & i^6 \\ 1 & i^3 & i^6 & i^9 \end{bmatrix} \quad \text{y} \quad \begin{bmatrix} \boldsymbol{F_2} & \\ & \boldsymbol{F_2} \end{bmatrix} = \begin{bmatrix} 1 & 1 & & \\ 1 & i^2 & & \\ & & 1 & 1 \\ & & 1 & i^2 \end{bmatrix}.$$

A la izquierda está F_4, sin ceros. A la derecha hay una matriz que es mitad cero. Tenemos la mitad del trabajo. Pero esas matrices no son iguales. Necesitamos dos matrices ralas y simples para completar la factorización FFT:

La FFT tiene tres matrices

$$F_4 = \begin{bmatrix} 1 & & 1 & \\ & 1 & & i \\ 1 & & -1 & \\ & 1 & & -i \end{bmatrix} \begin{bmatrix} 1 & 1 & & \\ 1 & i^2 & & \\ & & 1 & 1 \\ & & 1 & i^2 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & & 1 & \\ & 1 & & \\ & & & 1 \end{bmatrix}. \tag{10}$$

La última matriz es una permutación. Pone los c pares (c_0 y c_2) delante de los c impares (c_1 y c_3). La matriz del medio realiza **transformadas de la mitad de tamaño** F_2 y F_2 en los c pares y en los c impares por separado. La matriz de la izquierda combina los dos resultados de la mitad de tamaño de manera que se obtiene el resultado correcto $\boldsymbol{y} = F_4\boldsymbol{c}$.

La misma idea se aplica cuando $N = 1024$ y $M = \frac{1}{2}N = 512$. El número w es $e^{2\pi i/1024}$. Tiene ángulo $\theta = 2\pi/1024$ del círculo unitario. La matriz de Fourier F_{1024} está llena de potencias de w. La primera fase de la FFT es la gran factorización descubierta por Cooley y Tukey (y presagiada en 1805 por Gauss):

$$\boxed{\boldsymbol{F_{1024}} = \begin{bmatrix} I_{512} & D_{512} \\ I_{512} & -D_{512} \end{bmatrix} \begin{bmatrix} \boldsymbol{F_{512}} & \\ & \boldsymbol{F_{512}} \end{bmatrix} \begin{bmatrix} \text{permutación} \\ \text{par-impar} \end{bmatrix}.} \tag{11}$$

I_{512} es la matriz identidad. D_{512} es la matriz diagonal con elementos $(1, w, \ldots, w^{511})$. Las dos copias de F_{512} son lo que esperábamos. No olvidemos que usan la 512^{a} raíz unitaria (¡ que es w^2 !). La matriz de permutación separa el vector entrante $\boldsymbol{c}$ en sus partes pares e impares $\boldsymbol{c'} = (c_0, c_2, \ldots, c_{1022})$ y $\boldsymbol{c''} = (c_1, c_3, \ldots, c_{1023})$.

He aquí las fórmulas algebraicas que confirman lo mismo que esa factorización de F_{1024}:

(Un paso de la FFT) Sea $M = \frac{1}{2}N$. Las primeras M y las últimas M componentes de $\boldsymbol{y} = F_N \boldsymbol{c}$ combinan las dos medias transformadas $\boldsymbol{y'} = F_M \boldsymbol{c'}$ e $\boldsymbol{y''} = F_M \boldsymbol{c''}$. La Ecuación (11) muestra este paso de N a $M = N/2$ como $I\boldsymbol{y'} + D\boldsymbol{y''}$ e $I\boldsymbol{y'} - D\boldsymbol{y''}$:

$$\begin{aligned} y_j &= y'_j + (w_N)^j y''_j, \quad j = 0, \ldots, M-1 \\ y_{j+M} &= y'_j - (w_N)^j y''_j, \quad j = 0, \ldots, M-1. \end{aligned} \tag{12}$$

Dividamos $\boldsymbol{c}$ en $\boldsymbol{c'}$ y $\boldsymbol{c''}$. Con F_M los transformamos en $\boldsymbol{y'}$ e $\boldsymbol{y''}$. Así (12) reconstruye $\boldsymbol{y}$. Esas fórmulas se obtienen de separar los $c_0 \ldots, c_{N-1}$ en los c_{2k} de índice par y los c_{2k+1} de índice impar en (13).

$$\boldsymbol{y_j} = \sum_{0}^{N-1} w^{jk} c_k = \sum_{0}^{M-1} w^{2jk} c_{2k} + \sum_{0}^{M-1} w^{j(2k+1)} c_{2k+1} \quad \text{con } M = \tfrac{1}{2}N,\ w = w_N. \tag{13}$$

Los c de índice par van en $\boldsymbol{c'} = (c_0, c_2, \ldots)$ y los c de índice impar van en $\boldsymbol{c''} = (c_1, c_3, \ldots)$. Luego vienen las transformadas $F_M \boldsymbol{c'}$ y $F_M \boldsymbol{c''}$. **La clave es $\boldsymbol{w_N^2 = w_M}$.** Esto da $w_N^{2jk} = w_M^{jk}$.

Reescribimos (13) $$\boldsymbol{y_j} = \sum (w_M)^{jk} c'_k + (w_N)^j \sum (w_M)^{jk} c''_k = \boldsymbol{y'_j} + \boldsymbol{(w_N)^j y''_j} \tag{14}$$

Para $j \geq M$, el signo menos en (12) se obtiene de factorizar $(w_N)^M = -1$ a partir de $(w_N)^j$.

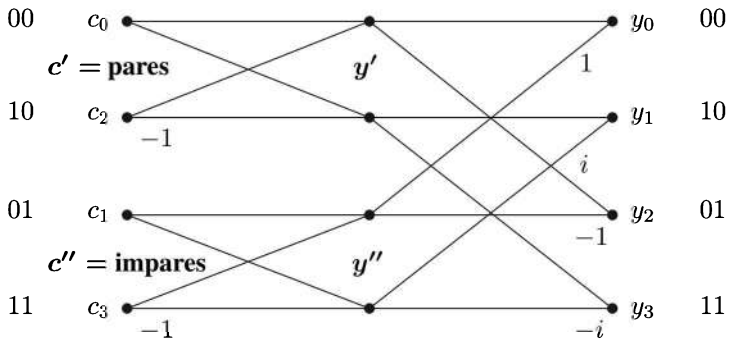
MATLAB separa fácilmente los c pares de los c impares y los multiplica por w_N^j. Usamos $\mathsf{conj}(F)$ o de manera equivalente la transformada inversa de MATLAB: ifft para obtener $\boldsymbol{y} = F\boldsymbol{c}$. Recordemos que fft está basada en $\omega = \overline{w} = e^{-2\pi i/N}$ y produce $\boldsymbol{c}$ a partir de $\boldsymbol{y}$ usando Ω.

Paso FFT de N a $N/2$ en MATLAB		
	Transformar los c pares	$y' =$ ifft $(c(0:2:N-2)) * N/2;$
	Transformar los c impares	$y'' =$ ifft $(c(1:2:N-1)) * N/2;$
	El vector $1, w, \ldots$ es $\boldsymbol{d}$	$d = w.\hat{}(0:N/2-1)';$
	Combinar $\boldsymbol{y'}$ e $\boldsymbol{y''}$	$y = [y' + d.* y''\,;\ y' - d.* y''];$

El grafo de flujo en la siguiente página muestra $\boldsymbol{c'}$ y $\boldsymbol{c''}$ atravesando la F_2 de la mitad de tamaño. Esos pasos son conocidos como "*mariposas*," debido a su forma. Luego los resultados $\boldsymbol{y'}$ e $\boldsymbol{y''}$ se combinan (multiplicando y'' por $1, i$ a partir de D y también por $-1, -i$ a partir de $-D$) para producir $\boldsymbol{y} = F_4 \boldsymbol{c}$.

Esta reducción de F_N en dos F_M casi reduce el trabajo a la mitad, podemos ver los ceros en la factorización de la matriz. Esa reducción del 50 % es buena pero no genial. La **FFT** completa es mucho más poderosa. Nos ahorra mucho más que la mitad del tiempo. La idea clave es **recursión**.

Y la factorización de F se aplica por igual a la matriz conjugada $\Omega = \overline{F}$.



La FFT completa por recursión

Si ha llegado hasta acá, probablemente adivinó qué sigue a continuación. Hemos reducido F_N a $F_{N/2}$. ***Continuemos con*** $\boldsymbol{F_{N/4}}$. Cada F_{512} lleva a F_{256}. Así 256 lleva a 128. *Eso es la recursión.*

La recursión es un principio básico de muchos algoritmos rápidos. He aquí el paso 2 con cuatro copias de F_{256} y D (256 potencias de w_{512}). Los índices pares de los primeros pares $c_0, c_4, c_8, \ldots$ vienen antes que $c_2, c_6, c_{10}, \ldots$

$$\begin{bmatrix} F_{512} & \\ & F_{512} \end{bmatrix} = \begin{bmatrix} I & D & & \\ I & -D & & \\ & & I & D \\ & & I & -D \end{bmatrix} \begin{bmatrix} F & & & \\ & F & & \\ & & F & \\ & & & F \end{bmatrix} \begin{bmatrix} \text{elegir} & 0,4,8,\ldots \\ \text{elegir} & 2,6,10,\ldots \\ \text{elegir} & 1,5,9,\ldots \\ \text{elegir} & 3,7,11,\ldots \end{bmatrix}.$$

Contaremos las multiplicaciones individuales, para ver cuánto ahorramos. Antes de que se inventara la **FFT**, la cuenta era la usual $N^2 = (1024)^2$. Esto es como un millón de multiplicaciones. No digo que cada una requiera mucho tiempo. El costo es grande cuando tenemos que realizar muchas, muchas transformadas, que es lo típico. Así el ahorro con la FFT también es grande:

La cuenta final para el tamaño $\boldsymbol{N = 2^\ell}$ ***se reduce de*** $\boldsymbol{N^2}$ ***a*** $\boldsymbol{\frac{1}{2}N\ell}$.

El número 1024 es 2^{10}, así $\ell = 10$. La cuenta original de $(1024)^2$ se reduce a (5)(1024). El ahorro es un factor de 200. Un millón se reduce a cinco mil. Por eso la FFT revolucionó el procesamiento de señales. He aquí el razonamiento detrás de $\frac{1}{2}N\ell$. Hay ℓ niveles, desde $N = 2^\ell$ hasta $N = 1$. Cada nivel tiene $N/2$ multiplicaciones de la diagonal D, para reconstruir los resultados de mitad de tamaño del nivel más bajo. Esto resulta en una cuenta final de $\frac{1}{2}N\ell$, que es $\frac{1}{2}N \log_2 N$.

Un último comentario sobre este fantástico algoritmo. Hay una regla sorprendente para el orden en que los c ingresan en la FFT, luego de todas las permutaciones par-impar. Escriba los números del 0 al $n-1$ en binario (como $00, 01, 10, 11$ para $n = 4$). Invierta el orden de estos dígitos: $00, 10, 01, 11$. Eso da el **orden inverso en bits 0, 2, 1, 3** con los pares antes que los impares.

La imagen completa muestra los c en orden inverso en bits, los $\ell = \log_2 N$ pasos de la recursión, y el resultado final $y_0, \ldots, y_{N-1}$ que es F_N por $\boldsymbol{c}$.

La recursión FFT para la matriz DFT $\Omega = \overline{F}$ usa exactamente las mismas ideas.

Problemas de la Sección IV.1

1 Luego de $S = 0$ en la ecuación (5), el texto dice que todos los elementos por fuera de la diagonal de $F_n \Omega_N$ son cero. Explique por qué (fila i de F) $\cdot$ (columna j de Ω) = producto punto complejo $= 0$.

¿Por qué es $(1, w^i, w^{2i}, \ldots, w^{(N-1)i}) \cdot (1, \omega^j, \omega^{2j}, \ldots, \omega^{(N-1)j}) = 0$ si $i \neq j$?

Debe usar el producto interno complejo, no $x_1 y_1 + \ldots + x_N y_N$.

2 Si $M = \frac{1}{2}N$, muestre que $(w_N)^M = -1$. Esto se usa en la ecuación FFT (12).

3 ¿Cuáles son las matrices F_3 y Ω_3 (con $w = e^{2\pi i/3}$ y $\omega = \overline{w}$)? ¿Cuál matriz de permutación P de 3 por 3 es la que las conectará de manera que $\Omega = FP$ y $F = \Omega P$?

4 Encuentre la transformada discreta de Fourier $\boldsymbol{c}$ de $\boldsymbol{f} = (0, 1, 0, 0)$. Verifique que la transformada inversa de $\boldsymbol{c}$ sea $\boldsymbol{f}$.

5 Conecte F_6 a dos copias de F_3 mediante una ecuación matricial como (10) y (11).

6 Para $N = 6$, ¿cómo ve que $1 + w + w^2 + w^3 + w^4 + w^5 = 0$?

7 Sea $f(x) = 1$ para $|x| \leq \pi/2$ y $f(x) = 0$ para $\pi/2 < |x| \leq \pi$. Esta función es "par" porque $f(-x) = f(x)$. Las funciones pares pueden expandirse en *series de cosenos*:

$$f(x) = a_0 + a_1 \cos x + a_2 \cos 2x + \cdots$$

Integre ambos lados desde $x = -\pi$ a π, para encontrar a_0. Multiplique ambos lados por $\cos x$ e integre desde $-\pi$ a π, para encontrar a_1.

8 Para toda matriz real A con n columnas se verifica $AA^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{a}_1(\boldsymbol{a}_1^{\mathrm{T}}\boldsymbol{x}) + \cdots + \boldsymbol{a}_n(\boldsymbol{a}_n^{\mathrm{T}}\boldsymbol{x})$.

Sea A una matriz ortogonal Q, ¿qué tienen de especial esos n sumandos?

Para la matriz de Fourier (compleja), ¿qué cambia en esa fórmula?

9 ¿Para qué vector $\boldsymbol{x}$ se verifica que $F_4\boldsymbol{x} = (1, 0, 1, 0)$?

¿Para qué vector y se verifica que $F_4\boldsymbol{y} = (0, 0, 0, 1)$?

IV.2. Matrices de desplazamiento y matrices circulantes

Cuando esta matriz P multiplica un vector $\boldsymbol{x}$, las componentes de $\boldsymbol{x}$ se desplazan hacia arriba :

Desplazamiento hacia arriba
Permutación cíclica

$$P\boldsymbol{x} = \begin{bmatrix} 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \\ \boldsymbol{x_3} \\ \boldsymbol{x_4} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x_2} \\ \boldsymbol{x_3} \\ \boldsymbol{x_4} \\ \boldsymbol{x_1} \end{bmatrix} \tag{1}$$

Las palabras "cíclica" y "circular" se aplican a P porque la primera componente x_1 se corre al final. Si pensamos en los números x_1, x_2, x_3, x_4 alrededor de un círculo, P los corre a todos una posición. Y P^2 gira el círculo dos posiciones :

$$P^2\boldsymbol{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \boldsymbol{x_3} \\ \boldsymbol{x_4} \\ \boldsymbol{x_1} \\ \boldsymbol{x_2} \end{bmatrix} \tag{2}$$

Todo factor P nuevo produce un desplazamiento adicional. **Así $\boldsymbol{P^4}$ da un giro completo de 360 °** : $\boldsymbol{P^4x = x}$ y $\boldsymbol{P^4 = I}$. Las siguientes potencias P^5, P^6, P^7, P^8 repiten el patrón y dan otra vuelta entera. Observe que P^3 es la inversa de P, porque $(P^3)(P) = P^4 = I$.

La siguiente matriz se llama **circulante**. Es tan solo una combinación de P, P^2, P^3 y $P^4 = I$. Tiene **diagonales constantes** :

Matriz circulante

$$\boldsymbol{C = c_0 I + c_1 P + c_2 P^2 + c_3 P^3} = \begin{bmatrix} c_0 & \boldsymbol{c_1} & c_2 & c_3 \\ c_3 & c_0 & \boldsymbol{c_1} & c_2 \\ c_2 & c_3 & c_0 & \boldsymbol{c_1} \\ \boldsymbol{c_1} & c_2 & c_3 & c_0 \end{bmatrix} \tag{3}$$

Cada diagonal de esa matriz se desplaza exactamente como los 1 en P. La diagonal con c_1, c_1, c_1 se completa con el cuarto c_1 al final. Importante : **si multiplicamos dos matrices circulantes $\boldsymbol{C}$ y $\boldsymbol{D}$, su producto $\boldsymbol{CD = DC}$ es a su vez una matriz circulante.**

Cuando multiplicamos CD, estamos multiplicando potencias de P para obtener más potencias de P. Y DC hace lo mismo. El siguiente ejemplo tiene $N = 3$ y $P^3 = I$:

$$CD = \begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} \\ 3 & 1 & 2 \\ 2 & 3 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{5} & \mathbf{0} & \mathbf{4} \\ 4 & 5 & 0 \\ 0 & 4 & 5 \end{bmatrix} = \begin{bmatrix} \mathbf{13} & \mathbf{22} & \mathbf{19} \\ 19 & 13 & 22 \\ 22 & 19 & 13 \end{bmatrix} = \textbf{ circulante} \tag{4}$$

$$\begin{matrix} (I + 2P + 3P^2)(5I + 4P^2) \\ (5I + 4P^2)(I + 2P + 3P^2) \end{matrix} = 5I + 10P + (15 + 4)P^2 + 8P^3 + 12P^4 = \boldsymbol{13I + 22P + 19P^2}$$

En ese último paso necesitamos y usamos las propiedades circulares $P^3 = I$ y $P^4 = P$. Así los vectores $(1,2,3)$ de C y $(5,0,4)$ de D producen el vector $(13,22,19)$ de CD y DC. Esta operación con vectores se llama **convolución cíclica**.

Para resumir: cuando multiplicamos las matrices circulantes C y D de N por N, realizamos la convolución cíclica de los vectores $(c_0, c_1, \ldots, c_{N-1})$ y $(d_0, d_1, \ldots, d_{N-1})$. La convolución ordinaria encuentra los coeficientes cuando multiplicamos $(c_0 I + c_1 P + \cdots + c_{N-1} P^{N-1})$ por $(d_0 I + d_1 P + \cdots + d_{N-1} P^{N-1})$. Luego la convolución cíclica usa la igualdad crucial $P^N = I$.

Convolución	$(1,2,3) * (5,0,4) = (\mathbf{5,10,19,8,12})$	(5)
Convolución cíclica	$(1,2,3) \circledast (5,0,4) = (5+8, 10+12, 19) = (\mathbf{13,22,19})$	(6)

La convolución odinaria es la multiplicación que aprendimos en segundo grado (incluso más fácil pues no hay que "llevarse"nada a la siguiente columna):

$$
\begin{array}{rrrrr}
 & & 1 & 2 & 3 \\
 & & 5 & 0 & 4 \\
\hline
 & & 4 & 8 & 12 \\
 & 0 & 0 & 0 & \\
5 & 10 & 15 & & \\
\hline
\mathbf{5} & \mathbf{10} & \mathbf{19} & \mathbf{8} & \mathbf{12}
\end{array}
\qquad = \boldsymbol{c} * \boldsymbol{d}
$$

El paso cíclico combina $5 + 8$ porque $P^3 = I$. Combina $10 + 12$ porque $P^4 = P$.
El resultado es $(13, 22, 19)$.

Practicar
$$(0,1,0) \circledast (d_0, d_1, d_2) = (d_2, d_0, d_1)$$
$$(1,1,1) \circledast (d_0, d_1, d_2) = (d_0 + d_1 + d_2, d_0 + d_1 + d_2, d_0 + d_1 + d_2)$$
$$(c_0, c_1, c_2) \circledast (d_0, d_1, d_2) = (d_0, d_1, d_2) \circledast (c_0, c_1, c_2)$$

Esa última línea significa que $\boldsymbol{CD = DC}$ para matrices circulantes y $\boldsymbol{c} \circledast \boldsymbol{d} = \boldsymbol{d} \circledast \boldsymbol{c}$ para convoluciones cíclicas. Las potencias de P en C conmutan con las potencias de P en D.

Si luego sumamos $1 + 2 + 3 = 6$ y $5 + 0 + 4 = 9$, tenemos una verificación rápida de la convolución. Multiplicamos 6 por 9 para obtener 54. Así 54 debería ser (y efectivamente es) igual a $5 + 10 + 19 + 8 + 12 = 54$. Y también $13 + 22 + 19 = 54$ para la convolución cíclica.

La suma de los c multiplicada por la suma de los d es igual a la suma de los resultados. Esto ocurre porque todo c multiplica a todo d en $\boldsymbol{c} * \boldsymbol{d}$ y en $\boldsymbol{c} \circledast \boldsymbol{d}$.

Autovalores y autovectores de P

Con $N = 4$, la ecuación $P\boldsymbol{x} = \lambda\boldsymbol{x}$ nos lleva directamente a cuatro autovalores y autovectores:

$$P\boldsymbol{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \\ x_4 \\ x_1 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad \text{es equivalente a} \quad \begin{matrix} x_2 = \lambda x_1 \\ x_3 = \lambda x_2 \\ x_4 = \lambda x_3 \\ x_1 = \lambda x_4 \end{matrix} . \tag{7}$$

Partamos de la última ecuación $x_1 = \lambda x_4$ y avancemos hacia arriba:

$$x_1 = \lambda x_4 = \lambda^2 x_3 = \lambda^3 x_2 = \lambda^4 x_1 \quad \text{implica que} \quad \boldsymbol{\lambda^4 = 1}.$$

Los autovalores de P son las raíces cuartas de 1. Son todas las potencias $i, i^2, i^3, 1$ de $w = i$.

$$\lambda = \boldsymbol{i}, \qquad \lambda = i^2 = \boldsymbol{-1}, \qquad \lambda = i^3 = \boldsymbol{-i}, \qquad \text{y} \quad \lambda = i^4 = \boldsymbol{1}. \tag{8}$$

Estas son las cuatro soluciones de $\det(P - \lambda I) = \lambda^4 - 1 = 0$. Los autovalores $i, -1, -i, 1$ son equidistantes sobre la circunferencia unitaria en el plano complejo (Figura IV.2).

Cuando P es de N por N, el mismo razonamiento nos lleva de $P^N = I$ a $\lambda^N = 1$. Los N autovalores son nuevamente equidistantes sobre la circunferencia, y ahora son las potencias del **número complejo** $\boldsymbol{w}$ en $360/N$ grados $= 2\pi/N$ radianes.

$$\boxed{\textbf{Las soluciones de } \boldsymbol{z^N = 1} \textbf{ son } \boldsymbol{\lambda = w, w^2, \ldots, w^{N-1}, 1} \textbf{ con } \boldsymbol{w = e^{2\pi i/N}}.} \tag{9}$$

En el plano complejo, el primer autovalor w es $e^{i\theta} = \cos\theta + i\,\mathrm{sen}\,\theta$ y el ángulo θ es $2\pi/N$. Los ángulos para los otros autovalores son $2\theta, 3\theta, \ldots, N\theta$. Como θ es $2\pi/N$, ese último ángulo es $N\theta = 2\pi$ y ese autovalor es $\lambda = e^{2\pi i}$ que es $\cos 2\pi + i\,\mathrm{sen}\,2\pi = 1$.

Para las potencias de números complejos, la forma polar con $e^{i\theta}$ es mucho mejor que usar $\cos\theta + i\,\mathrm{sen}\,\theta$.

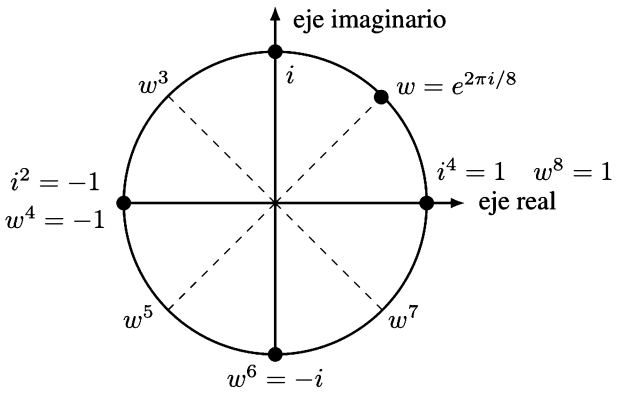


Figura IV.2: autovalores de P_N: las 4 potencias de $\lambda = i$ para $N = 4$ suman cero. Las 8 potencias de $\lambda = w = e^{2\pi i/8}$ cuando $N = 8$ también deben sumar cero.

Conociendo los N autovalores $\lambda = 1, w, \ldots, w^{N-1}$ de P_N, rápidamente encontramos N autovectores :

Fijemos la primera componente de $\boldsymbol{q}$ como 1. Las demás componentes de $\boldsymbol{q}$ son λ, λ^2 y λ^3 :

$$\textbf{Autovectores para } \boldsymbol{\lambda = 1, i, i^2, i^3} \qquad \boldsymbol{q}_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \boldsymbol{q}_1 = \begin{bmatrix} 1 \\ i \\ i^2 \\ i^3 \end{bmatrix} \quad \boldsymbol{q}_2 = \begin{bmatrix} 1 \\ i^2 \\ i^4 \\ i^6 \end{bmatrix} \quad \boldsymbol{q}_3 = \begin{bmatrix} 1 \\ i^3 \\ i^6 \\ i^9 \end{bmatrix} \tag{10}$$

Hemos comenzado la numeración en cero, como suelen hacerlo quienes trabajan con transformadas de Fourier. El autovector cero tiene autovalor $\lambda = 1 = w^0$. La matriz de autovectores tiene columnas $0, 1, 2, 3$ que contienen a $\boldsymbol{q}_0, \boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3$. **Esa matriz de autovectores para $\boldsymbol{P}$ es la matriz de Fourier**.

$$\textbf{Matriz de autovectores } \boldsymbol{N = 4} \textbf{ Matriz de Fourier} \qquad F = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & i^2 & i^3 \\ 1 & i^2 & i^4 & i^6 \\ 1 & i^3 & i^6 & i^9 \end{bmatrix} \quad \text{cumple} \quad \overline{F}^{\mathrm{T}} F = 4I. \tag{11}$$

El patrón se mantiene igual sin importar el tamaño N. El k-ésimo autovalor es $w^k = (e^{2\pi i/N})^k = e^{2\pi ik/N}$. De nuevo la cuenta parte de cero : $\lambda_0 = w^0 = 1, \lambda_1 = w, \ldots, \lambda_{N-1} = w^{N-1}$.

El $\boldsymbol{k}$-ésimo autovector contiene las potencias de $\boldsymbol{w^k}$. La matriz de autovectores contiene todos los N autovectores. Es la matriz de Fourier de N por N con $\overline{F}^{\mathrm{T}} F = NI$.

$$\textbf{Matriz de Fourier} \quad \textbf{Autovectores de } \boldsymbol{P} \qquad \boldsymbol{F_N} = \begin{bmatrix} 1 & 1 & 1 & \cdot & 1 \\ 1 & w & w^2 & \cdot & w^{N-1} \\ 1 & w^2 & w^4 & \cdot & w^{2(N-1)} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & w^{N-1} & w^{2(N-1)} & \cdot & w^{(N-1)\,(N-1)} \end{bmatrix}. \tag{12}$$

Vemos nuevamente que **las columnas de la matriz de Fourier son ortogonales**. Debemos usar el producto interno complejo $(\boldsymbol{x}, \boldsymbol{y}) = \overline{\boldsymbol{x}}^{\mathrm{T}} \boldsymbol{y}$. He aquí la nueva prueba de ortogonalidad :

Las matrices ortogonales como $\boldsymbol{P}$ tienen autovectores ortogonales.

Así $P = F\Lambda F^{\mathrm{T}}/N$ con su matriz de autovalores diagonal $\Lambda = \operatorname{diag}(1, w, \ldots, w^{N-1})$.

La siguiente página pasa de la permutación especial P a una matriz circulante C cualquiera.

Autovalores y autovectores de una C circulante

Los autovectores de una matriz C circulante son muy fáciles. **Esos autovectores son los mismos que los autovectores de la permutación P.** Son las columnas $q_0, q_1, \ldots, q_{N-1}$ de la misma matriz de Fourier F. Aquí $Cq_k = \lambda q_k$ para el k-ésimo autovector y autovalor:

$$(c_0 I + c_1 P + \cdots + c_{N-1} P^{N-1})\, q_k = (c_0 + c_1 \lambda_k + \cdots + c_{N-1} \lambda_k^{N-1})\, q_k. \tag{13}$$

Recordemos que $\lambda_k = w^k = e^{2\pi i k/N}$ es el k-ésimo autovalor de P. Esos números están en la matriz de Fourier F. Luego los autovalores de C en la ecuación (13) tienen una fórmula casi mágica: **multiplicar F por el vector c en la fila superior de C para encontrar los autovalores.**

$$\begin{bmatrix} \lambda_0(C) \\ \lambda_1(C) \\ \lambda_2(C) \\ \cdot \\ \lambda_{N-1}(C) \end{bmatrix} = \begin{bmatrix} c_0 + c_1 + \cdots + c_{N-1} \\ c_0 + c_1 w + \cdots + c_{N-1} w^{N-1} \\ c_0 + c_1 w^2 + \cdots + c_{N-1} w^{2(N-1)} \\ \cdot\ \cdot\ \cdot \\ c_0 + c_1 w^{N-1} + \cdots + c_{N-1} w^{(N-1)\,(N-1)} \end{bmatrix} = F \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \cdot \\ c_{N-1} \end{bmatrix} = \boldsymbol{Fc}. \tag{14}$$

Los autovalores de C son las componentes de Fc = transformada inversa de Fourier de c.

Ejemplo con $N = 2$ para $w = e^{2\pi i/2} = -1$ en la matriz de Fourier F

$$P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{y} \quad C = \begin{bmatrix} c_0 & c_1 \\ c_1 & c_0 \end{bmatrix} \quad \text{y} \quad \boldsymbol{c} = \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} \quad \text{y} \quad F = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Los autovectores de P y C son las columnas de F. Los autovalores de P son ± 1. Los autovalores de $C = c_0 I + c_1 P$ son $c_0 + c_1$ y $c_0 - c_1$. Estos autovalores de C son las componentes de F por $\boldsymbol{c}$:

$$\textbf{Autovalores de } \boldsymbol{C} \qquad F\boldsymbol{c} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{c_0 + c_1} \\ \boldsymbol{c_0 - c_1} \end{bmatrix}. \tag{15}$$

Para cualquier N, la permutación P es una matriz circulante C con $\boldsymbol{c} = (0, 1, 0, \ldots, 0)$.
Los autovalores de P están en el vector columna $F\boldsymbol{c}$ con este $\boldsymbol{c}$.
Esa es la columna $(1, w, w^2, \ldots, w^{N-1})$ de la matriz de Fourier F.
Esto coincide con los autovalores $1, i, i^2, i^3$ de P en la ecuación (8), para $N = 4$.

La regla de convolución

Esta regla compara **convolución** con **multiplicación**. Entienda que son muy diferentes, pero tienen una bella conexión gracias a la matriz de Fourier F.

Comenzamos con dos matrices circulantes C y D. Sus filas superiores son los vectores $\boldsymbol{c}$ y $\boldsymbol{d}$. La ecuación (4) al inicio de esta sección muestra un ejemplo de la primera fila de CD:

$$\textbf{Primera fila de } \boldsymbol{CD} = \textbf{ Convolución cíclica } = \boldsymbol{c} \circledast \boldsymbol{d}. \tag{16}$$

Así los autovalores de CD según la ecuación (14) están en el vector $\boldsymbol{F(c \circledast d)}$.

Ahora encontremos esos autovalores de CD de otra manera. Los autovalores $\lambda(C)$ están en el vector $F\boldsymbol{c}$. Los autovalores $\lambda(D)$ están en el vector $F\boldsymbol{d}$. ¡ Los autovectores $\boldsymbol{q}_k$ son los mismos para C y D ! Son las columnas de F. **Así cada autovalor $\boldsymbol{\lambda_k(CD)}$ es $\boldsymbol{\lambda_k(C)}$ por $\boldsymbol{\lambda_k(D)}$. Este "producto de Hadamard" realizado término a término se escribe en** MATLAB **con .***

$$\begin{bmatrix} \lambda_0(CD) \\ \vdots \\ \lambda_{N-1}(CD) \end{bmatrix} = \begin{bmatrix} \lambda_0(C)\lambda_0(D) \\ \vdots \\ \lambda_{N-1}(C)\lambda_{N-1}(D) \end{bmatrix} = \begin{bmatrix} \lambda_0(C) \\ \vdots \\ \lambda_{N-1}(C) \end{bmatrix} \boldsymbol{.*} \begin{bmatrix} \lambda_0(D) \\ \vdots \\ \lambda_{N-1}(D) \end{bmatrix} = \boldsymbol{Fc \;.*\; Fd}$$

Esa notación $\boldsymbol{.*}$ indica una multiplicación componente a componente de dos vectores.

La regla de convolución compara nuestras dos fórmulas para los autovalores de CD:

Convolucionar vectores
Multiplicar transformadas

$$\boxed{\textbf{Regla de convolución} \qquad \boldsymbol{F\,(c \circledast d) = (Fc).*(Fd).}} \tag{17}$$

A la izquierda Convolucionar $\boldsymbol{c}$ y $\boldsymbol{d}$ primero, luego transformar con F.
A la derecha Transformar con F primero, luego multiplicar $F\boldsymbol{c}$ por $F\boldsymbol{d}$ componente a componente.

¡ Esta es la identidad fundamental del procesamiento de señales !
Las transformadas son rápidas usando la FFT.

Otra manera de ver la regla de convolución es multiplicando las matrices diagonales $\Lambda(C)$ y $\Lambda(D)$ que contienen los autovalores de C y de D. C se diagonaliza por $F^{-1}CF = \Lambda(C)$:

$$(F^{-1}CF)\,(F^{-1}DF) = F^{-1}(CD)F \text{ es exactamente } \boldsymbol{\Lambda(C)\,\Lambda(D) = \Lambda(CD)}. \tag{18}$$

Esto se logra exitosamente porque todas las matrices circulantes tienen los mismos autovectores (las columnas de F).

La regla de convolución puede verificarse directamente (Problema 1). Está bueno verlo para $N = 2$:

$$F = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \boldsymbol{c} \circledast \boldsymbol{d} = \begin{bmatrix} c_0d_0 + c_1d_1 \\ c_0d_1 + c_1d_0 \end{bmatrix} \qquad F\boldsymbol{c} = \begin{bmatrix} c_0 + c_1 \\ c_0 - c_1 \end{bmatrix} \qquad F\boldsymbol{d} = \begin{bmatrix} d_0 + d_1 \\ d_0 - d_1 \end{bmatrix}$$

La regla de convolución (17) establece que $\boldsymbol{F(c \circledast d)}$ es el producto que se obtiene de multiplicar componente a componente.

$$F(\boldsymbol{c} \circledast \boldsymbol{d}) = \begin{bmatrix} c_0d_0 + c_1d_1 + c_0d_1 + c_1d_0 \\ c_0d_0 + c_1d_1 - c_0d_1 - c_1d_0 \end{bmatrix} = \begin{bmatrix} (c_0 + c_1)(d_0 + d_1) \\ (c_0 - c_1)(d_0 - d_1) \end{bmatrix} = (F\boldsymbol{c}) \boldsymbol{.*} (F\boldsymbol{d})$$

Multiplicación y convolución de funciones

Si $f(x) = \Sigma\, c_k e^{ikx}$ y $g(x) = \Sigma\, d_m e^{imx}$, ¿qué coeficientes de Fourier tiene $f(x)\, g(x)$? Estamos multiplicando las funciones 2π-periódicas f y g.

La multiplicación fg está en el "espacio x". Por la regla de convolución, esperamos convolucionar $\boldsymbol{c} * \boldsymbol{d}$ en el "espacio de frecuencias". Estas son las funciones periódicas f y g $(-\pi \le x \le \pi)$ y sus series de Fourier son infinitas $(k = 0, \pm 1, \pm 2, \ldots)$. Así la convolución cíclica desaparece y multiplicamos dos series de Fourier infinitas:

$$f(x)\, g(x) = \left(\sum_{k=-\infty}^{\infty} c_k e^{ikx} \right) \left(\sum_{m=-\infty}^{\infty} d_m e^{imx} \right) = \sum_{n=-\infty}^{\infty} h_n e^{inx}. \tag{19}$$

¿ Cuándo e^{ikx} por e^{imx} produce e^{inx} ? El requisito es $\boldsymbol{k + m = n}$.

El coeficiente h_n combina todos los productos $c_k d_m$ con $k + m = n$. Así $\boldsymbol{m = n - k}$:

$$\boldsymbol{h_n = \sum_{k=-\infty}^{\infty} c_k d_{n-k}} \text{ es la convolución } \boldsymbol{h = c * d} \text{ para vectores infinitos.} \tag{20}$$

A continuación multiplicamos los coeficientes $c_k d_k$ en el espacio de la variable k. ¡ Entonces convolucionamos $f * g$ en el espacio de la variable x !

Convolución de funciones 2π-periódicas
$$(\boldsymbol{f * g})(x) = \int_{t=-\pi}^{\pi} \boldsymbol{f(t)\, g(x - t)}\, dt \tag{21}$$

Regla de convolución para funciones periódicas
Los coeficientes de Fourier de $\boldsymbol{f * g}$ son $\boldsymbol{2\pi c_k d_k}$ (22)

Para ver que $\boldsymbol{f * g = g * f}$ en la ecuación (21), cambiamos las variables a $T = x - t$ y $t = x - T$.

La función delta $\delta(x)$ es la identidad para la convolución, como I para la multiplicación:

$$(\boldsymbol{\delta * g})(x) = \int \delta(t)\, g(x - t)\, dt = \boldsymbol{g(x)} \quad \text{y} \quad (1,0,0) \circledast (a,b,c) = (a,b,c).$$

Correlación cruzada y autocorrelación

La correlación cruzada es como la convolución, pero es diferente y la indicamos como $**$

$$[\textbf{No } \boldsymbol{n - k}] \qquad h_n = \sum_k c_k d_{\boldsymbol{n+k}} \text{ es } \textbf{la correlación cruzada } \boldsymbol{h = c ** d} \text{ para vectores} \tag{23}$$

$$(\boldsymbol{f ** g})(x) = \int f(t)\, g(\boldsymbol{x + t})\, dt = \boldsymbol{f(x) * g(-x)} \text{ es la correlación cruzada para funciones} \tag{24}$$

Estamos desplazando el vector $\boldsymbol{d}$ y tomando su producto punto con $\boldsymbol{c}$. Estamos desplazando la función g a lo largo del eje x y tomando su producto interno con f. Estos productos punto serán máximo cuando los vectores $\boldsymbol{c}, \boldsymbol{d}$ y las funciones f, g estén lo más alineadas posible.

Es valioso tener esta sencilla manera de encontrar la mejor alineación.

Los casos especiales $\boldsymbol{c} = \boldsymbol{d}$ y $f = g$ son los más importantes y sin duda los mejores alineados. En esos casos, las correlaciones cruzadas $\boldsymbol{c} * * \boldsymbol{c}$ y $f * * f$ se llaman **autocorrelaciones**.

¡ La convolución y la correlación cruzada se corresponden a la perfección con la multiplicación de matrices !

$\boldsymbol{c} * \boldsymbol{d}$ da los elementos en CD (matrices infinitas de diagonal constante)

$\boldsymbol{c} \circledast \boldsymbol{d}$ da los elementos en CD (matrices circulantes finitas)

$\boldsymbol{c} * * \boldsymbol{d}$ da los elementos en $C^{\mathrm{T}}D$ y $\boldsymbol{a} * * \boldsymbol{a}$ da los elementos en $A^{\mathrm{T}}A$

Problemas de la Sección IV.2

1 Encuentre $\boldsymbol{c} * \boldsymbol{d}$ y $\boldsymbol{c} \circledast \boldsymbol{d}$ para $\boldsymbol{c} = (2, 1, 3)$ y $\boldsymbol{d} = (3, 1, 2)$.

2 Demuestre la regla de convolución para $N = 3$: la k-ésima componente de $F(\boldsymbol{c} \circledast \boldsymbol{d})$ es igual a $(F\boldsymbol{c})_k$ por $(F\boldsymbol{d})_k$. Comience a partir de $(\boldsymbol{c} \circledast \boldsymbol{d})_p = c_0 d_p + c_1 d_{p-1} + c_2 d_{p-2}$.

$$\text{Demuestre que} \quad \sum_{p=0}^{2} w^{kp} (\boldsymbol{c} \circledast \boldsymbol{d})_p = \left(\sum_{m=0}^{2} w^{km} c_m \right) \left(\sum_{n=0}^{2} w^{kn} d_n \right) \quad \text{donde} \quad w^3 = 1$$

3 Si $\boldsymbol{c} * \boldsymbol{d} = \boldsymbol{e}$, ¿ por qué $(\sum c_i)(\sum d_i) = (\sum e_i)$? ¿ Por qué tuvimos éxito en nuestra verificación ? $(1 + 2 + 3)\,(5 + 0 + 4) = (\mathbf{6})\,(\mathbf{9}) = \mathbf{54} = 5 + 10 + 19 + 8 + 12$.

4 Dos matrices circulantes del mismo tamaño conmutan : $CD = DC$. Tienen los mismos autovectores $\boldsymbol{q}_k$ (las columnas de la matriz de Fourier F). Muestre que los autovalores $\lambda_k(CD)$ son iguales a $\lambda_k(C)$ por $\lambda_k(D)$.

5 ¿ Cuáles son los autovalores de la matriz circulante $C = I + P + P^2 + P^3$ de 4 por 4 ? Conecte esos autovalores a la transformada discreta $F\boldsymbol{c}$ con $\boldsymbol{c} = (1, 1, 1, 1)$. ¿ Para cuáles tres números reales o complejos z es $1 + z + z^2 + z^3 = 0$?

6 "Una matriz circulante C es invertible cuando el vector $F\boldsymbol{c}$ no tiene ceros." Conecte esta afirmación correcta con la siguiente verificación en el espacio de frecuencias :

$$C(e^{i\theta}) = \sum_{0}^{N-1} c_j e^{ij\theta} \neq 0 \ \text{ en los } \ N \ \text{ puntos } \ \theta = 2\pi/N, 4\pi/N, \ldots, 2\pi.$$

7 ¿ Cómo encontraría $\boldsymbol{d}$ en una ecuación de convolución $\boldsymbol{c} * \boldsymbol{d} = \boldsymbol{e}$ o en $\boldsymbol{c} \circledast \boldsymbol{d} = \boldsymbol{e}$? Con matrices esto es $CD = E$ y entonces $D = C^{-1}E$. Pero la deconvolución es a menudo más rápida con la regla de convolución $(F\boldsymbol{c}).*(F\boldsymbol{d}) = (F\boldsymbol{e})$. Así $F\boldsymbol{d} =$ ¿ ?

8 La n-ésima componente de la autocorrelación $\boldsymbol{c} * * \boldsymbol{c}$ es el producto punto entre los vectores $\boldsymbol{c}$ y $S^n\boldsymbol{c}$ (el vector $\boldsymbol{c}$ es desplazado n lugares). ¿ *Por qué* $\boldsymbol{c}^{\mathrm{T}} S^n \boldsymbol{c} \leq \boldsymbol{c}^{\mathrm{T}} \boldsymbol{c}$? Así la componente más grande de $\boldsymbol{c} * * \boldsymbol{c}$ es la componente cero de $\boldsymbol{c}^{\mathrm{T}} \boldsymbol{c}$ (sin desplazamiento).

IV.3. El producto de Kronecker $A \otimes B$

La Sección IV.1 describe la transformada discreta de Fourier para una señal unidimensional $\boldsymbol{f}$. Esta sección describe la DFT bidimensional, que se necesita para el procesamiento de imágenes. Cuando la transformada unidimensional usa una matriz de tamaño N, la transformada bidimensional necesita una matriz de tamaño N^2 (y el vídeo introducirá una tercera dimensión). Las matrices 2D serán grandes. Esperamos construirlas con facilidad a partir de las matrices de Fourier F y Ω correspondientes al caso unidimensional.

Esta construcción usa los **productos de Kronecker $\boldsymbol{F \otimes F}$ y $\boldsymbol{\Omega \otimes \Omega}$.** Antes se denominaba *producto tensorial.* Los comandos MATLAB son **kron**(F, F) y **kron**(Ω, Ω).

Esta operación es extremadamente conveniente en muchos contextos. Por eso esta sección desarrolla las ideas y operaciones claves de $K = A \otimes B$: para invertir K, para resolver $(A \otimes B)\,\boldsymbol{x} = \boldsymbol{y}$, y para encontrar los autovalores y los autovectores y la SVD de $A \otimes B$.

Lo primero que debemos saber sobre los productos de Kronecker es el tamaño de $A \otimes B = \mathsf{kron}(A, B)$:

1 Si A y B son de n por n, entonces **$\boldsymbol{A \otimes B}$ es de $\boldsymbol{n^2}$ por $\boldsymbol{n^2}$.**

2 Si A es de m por n y B es de M por N, entonces **$\boldsymbol{A \otimes B}$ tiene $\boldsymbol{mM}$ filas y $\boldsymbol{nN}$ columnas.**

Los elementos de $A \otimes B$ son (todos los mn elementos de A) por (todos los MN elementos de B).

Lo siguiente es la posición de esos productos en la matriz grande. La regla es **multiplicar cada elemento de $\boldsymbol{A}$ por toda la matriz $\boldsymbol{B}$.** Así $A \otimes B$ es una matriz por bloques. Todo bloque es un múltiplo de B:

$$\textbf{Productos de Kronecker} \qquad \boldsymbol{A \otimes B} = \begin{bmatrix} \boldsymbol{a_{11}}B & \cdots & \boldsymbol{a_{1n}}B \\ \vdots & & \vdots \\ \boldsymbol{a_{m1}}B & \cdots & \boldsymbol{a_{mn}}B \end{bmatrix}. \tag{1}$$

El caso más simple se tiene con A y B matrices identidad: (de 2 por 2) $\otimes$ (de 3 por 3) = de 6 por 6.

$$\boldsymbol{I_2 \otimes I_3 = I_6} \qquad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{1}\,I_3 & \mathbf{0}\,I_3 \\ \mathbf{0}\,I_3 & \mathbf{1}\,I_3 \end{bmatrix} = \boldsymbol{I_6}. \tag{2}$$

Un caso más desafiante (pero no imposible) se obtiene de multiplicar dos productos de Kronecker:

$$\boldsymbol{A \otimes B} \quad \text{por} \quad \boldsymbol{C \otimes D} \quad \text{es igual a} \quad \boldsymbol{AC \otimes BD} \tag{3}$$

$$\boldsymbol{A \otimes B} \quad \text{por} \quad \boldsymbol{A^{-1} \otimes B^{-1}} \quad \text{es igual a} \quad \boldsymbol{I \otimes I} \tag{4}$$

La ecuación (3) permite matrices rectangulares, pero (4) es para matrices cuadradas invertibles. $I \otimes I$ es la matriz identidad tamaño nN. **Así la inversa de $\boldsymbol{A \otimes B}$ es $\boldsymbol{A^{-1} \otimes B^{-1}}$.**

La demostración de la ecuación (3) se obtiene de la multiplicación en bloques de los productos de Kronecker :

$$\begin{bmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{bmatrix}\begin{bmatrix} c_{11}D & c_{12}D \\ c_{21}D & c_{22}D \end{bmatrix}=\begin{bmatrix} (a_{11}c_{11}+a_{12}c_{21})\,BD & (a_{11}c_{12}+a_{12}c_{22})\,BD \\ (a_{21}c_{11}+a_{22}c_{21})\,BD & (a_{21}c_{12}+a_{22}c_{22})\,BD \end{bmatrix}. \quad (5)$$

Eso es exactamente $AC \otimes BD$. Una matriz de tamaño N^2 por una matriz de tamaño N^2 sigue siendo una matriz de tamaño N^2. La identidad básica final corresponde a la traspuesta de $A \otimes B$:

$$(\boldsymbol{A} \otimes \boldsymbol{B})^{\mathbf{T}} = \boldsymbol{A}^{\mathbf{T}} \otimes \boldsymbol{B}^{\mathbf{T}} \qquad \begin{bmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} a_{11}B^{\mathrm{T}} & a_{21}B^{\mathrm{T}} \\ a_{12}B^{\mathrm{T}} & a_{22}B^{\mathrm{T}} \end{bmatrix} \quad (6)$$

Las transformadas de Fourier discretas bidimensionales

Comencemos con una imagen de N por N. Podría ser una fotografía con N^2 píxeles pequeños que hayamos tomado con nuestro teléfono. Podría ser un cuadro de Chuck Close (quien descubrió que nuestros ojos fusionan los pequeños cuadrados en una imagen continua, y así también es como funcionan las cámaras de los teléfonos). Tenemos N^2 números.

Y tenemos que decidir. Podemos desplegar esos números en un largo vector de longitud N^2. O podemos guardarlos en una matriz de N por N, para que los píxeles que están cerca en la imagen se mantengan cerca. El despliegue en un vector (con un operador llamado vec) se describirá más adelante en esta sección. Por ahora pensamos en un vector $\boldsymbol{f}$ de $\mathbf{R}^{n^2}$ con sus n^2 componentes en un cuadrado.

Queremos aplicar una transformada discreta de Fourier $2D$ a $\boldsymbol{f}$. El resultado es un vector $2D$ al que llamamos $\boldsymbol{c}$. Este proceso se divide en dos pasos :

Fila por fila	Aplicar la DTF $1D$ a cada fila de píxeles por separado.
Columna por columna	Reacomodar el resultado por columnas y transformar cada columna.

La matriz para cada paso es de N^2 por N^2. Pensemos primero en los N^2 píxeles una fila a la vez y multipliquemos cada fila de ese largo vector por la matriz DFT Ω_N de una dimensión :

$$\boldsymbol{\Omega}_{\textbf{fila}}\boldsymbol{f} = \begin{bmatrix} \Omega_N & & & \\ & \Omega_N & & \\ & & . & \\ & & & \Omega_N \end{bmatrix} \begin{bmatrix} \text{fila } 1 \\ \text{fila } 2 \\ . \\ \text{fila } N \end{bmatrix} \qquad (\boldsymbol{f} \text{ y } \boldsymbol{\Omega}_{\textbf{fila}}\boldsymbol{f} \text{ tienen longitud } N^2) \quad (7)$$

Esa matriz es $\Omega_{\textbf{fila}} = I_N \otimes \Omega_N$. Es un producto de Kronecker de tamaño N^2.

Ahora el resultado $\boldsymbol{\Omega}_{\textbf{fila}}\boldsymbol{f}$ se reacomoda en columnas (mentalmente, no electrónicamente). El segundo paso de la transformada $2D$ multiplica cada columna de esa imagen "a medio camino" $\boldsymbol{\Omega}_{\textbf{fila}}\boldsymbol{f}$ por Ω_N. Una vez más estamos multiplicando por una matriz $\boldsymbol{\Omega}_{\textbf{columna}}$ de tamaño N^2. La tranformada 2D completa es $\Omega_N \otimes \Omega_N$.

Esa matriz $\boldsymbol{\Omega}_{\textbf{columna}}$ es el producto de Kronecker $\boldsymbol{\Omega_N \otimes I_N}$.

La transformada $\boldsymbol{2D}$ junta los pasos fila y columna en $\boldsymbol{\Omega_{N\times N}}$.

$$\boldsymbol{\Omega_{N\times N} = \Omega_{\textbf{columna}}\,\Omega_{\textbf{fila}} = (\Omega_N \otimes I_N)(I_N \otimes \Omega_N) = \Omega_N \otimes \Omega_N.} \quad (8)$$

Ejemplo 1 Con $N = 4$ hay $N^2 = 16$ píxeles en un cuadrado. La matriz de bloques $\Omega_{4\times 4}$ para la transformada $2D$ es de 16 por 16 :

$$\boldsymbol{\Omega_{4\times 4} = \Omega_4 \otimes \Omega_4} = \begin{bmatrix} \Omega_4 & \Omega_4 & \Omega_4 & \Omega_4 \\ \Omega_4 & -i\,\Omega_4 & (-i)^2\,\Omega_4 & (-i)^3\,\Omega_4 \\ \Omega_4 & (-i)^2\,\Omega_4 & (-i)^4\,\Omega_4 & (-i)^6\,\Omega_4 \\ \Omega_4 & (-i)^3\,\Omega_4 & (-i)^6\,\Omega_4 & (-i)^9\,\Omega_4 \end{bmatrix}$$

Esa matriz debe dividirse entre $4 \times 4 = 16$ para coincidir correctamente con la transformada $\Omega_4/4$ correspondiente a $1D$. Así la inversa de esta matriz de 16 por 16 da la DFT inversa bidimensional, que es $F_4 \otimes F_4$. Aplicamos la fórmula inversa (4) para productos de Kronecker :

$$\textbf{La inversa } 2D \textbf{ es } \left(\frac{\mathbf{1}}{\mathbf{16}}\boldsymbol{\Omega_4 \otimes \Omega_4}\right)^{\mathbf{-1}} \boldsymbol{= F_4 \otimes F_4} \textbf{ = producto de Kronecker de inversas } 1D.$$

La suma de Kronecker $A \oplus B$

La transformada de Fourier $2D$ es el **producto** de dos pasos : transformar por filas y transformar por columnas el resultado. Esos pasos fueron $I \otimes \Omega$ y $\Omega \otimes I$. En otros problemas queremos la **suma** en lugar del producto. Eso resulta en una matriz diferente. Su tamaño aún es N^2 o MN.

$$\boxed{\begin{array}{l} \boldsymbol{A} \textbf{ de } \boldsymbol{M} \textbf{ por } \boldsymbol{M} \\ \boldsymbol{B} \textbf{ de } \boldsymbol{N} \textbf{ por } \boldsymbol{N} \end{array} \quad \textbf{La suma de Kronecker } \boldsymbol{A \oplus B = A \otimes I_N + I_M \otimes B} \textbf{ es de } \boldsymbol{MN} \textbf{ por } \boldsymbol{MN}.} \tag{9}$$

Esta construcción es natural para la ecuación de Laplace (o la ecuación de Poisson) en 2D :

$$\textbf{Ecuación de Laplace en un cuadrado} \quad -\frac{\boldsymbol{\partial^2 u}}{\boldsymbol{\partial x^2}} - \frac{\boldsymbol{\partial^2 u}}{\boldsymbol{\partial y^2}} = \boldsymbol{F(x,y)} \quad \text{para } 0 \le x \le 1, 0 \le y \le 1$$

Dividamos ese cuadrado unitario en N^2 cuadrados pequeños de lado $h = 1/N$. Coloquemos nodos en los vértices de esos cuadrados. Habrá $(N+1)^2$ nodos. Luego reemplacemos la ecuación diferencial de segundo orden de Laplace por una ecuación en diferencias de segundo orden que conecte los valores U_{jk} correspondientes a los nodos :

En una dimensión $-\dfrac{\partial^2 u}{\partial x^2}$ se convierte en una diferencia segunda $\dfrac{-u(x+h) + 2u(x) - u(x-h)}{h^2}$

Para una recta de $N+1$ nodos, esas diferencias segundas van en una matriz Δ de tamaño $N+1$:

$$\begin{array}{c} N = 4 \\ h = 1/4 \end{array} \qquad \boldsymbol{\Delta_5} = \frac{1}{(1/4)^2} \begin{bmatrix} \mathbf{1} & -1 & & & \\ -1 & \mathbf{2} & -1 & & \\ & \mathbf{-1} & \mathbf{2} & \mathbf{-1} & \\ & & -1 & \mathbf{2} & -1 \\ & & & \mathbf{-1} & \mathbf{1} \end{bmatrix} \tag{10}$$

Observemos el primer y último elemento de la diagonal : **1 y no 2**. Esto refleja las condiciones de frontera en los extremos del intervalo. Estamos considerando la condición libre $\partial u/\partial x = 0$ en las filas 1 y 5. (Las condiciones fijas $u = 0$ llevarían a todos 2 sobre la diagonal.)

La matriz de diferencias segundas Δ_5 es **semidefinida positiva**. Su espacio nulo contiene al vector columna $\mathbf{1} = (1,1,1,1,1)$.

Esta matriz Δ_5 reemplaza a $-\partial^2/\partial x^2$ en cada fila y a $-\partial^2/\partial y^2$ en cada columna. Queremos la matriz $\boldsymbol{\Delta_{\text{fila}}}$ de 25×25 que determina diferencias de segundo orden en *todas las filas simultáneamente*, y también la $\boldsymbol{\Delta_{\text{columna}}}$ de las diferencias de segundo orden en *todas las columnas simultáneamente* :

$$\boldsymbol{\Delta_{\text{row}}} = \begin{bmatrix} \Delta_5 & & \\ & \ddots & \\ & & \Delta_5 \end{bmatrix} = \boldsymbol{I_5 \otimes \Delta_5}$$

$$\boldsymbol{\Delta_{\text{column}}} = \text{ (una columna a la vez)} = \boldsymbol{\Delta_5 \otimes I_5}$$

Por ahora es como la transformada discreta de Fourier : todas las operaciones de fila por un lado y las de columna por otro. La diferencia es que *sumamos* estas matrices. Queremos aproximar $-\partial^2/\partial x^2$ *más* $-\partial^2/\partial y^2$. La matriz de diferencias finitas $2D$ general es la **suma de Kronecker** de 25×25 :

$$\boldsymbol{\Delta_{5\,\times\,5}} = I_5 \otimes \Delta_5 + \Delta_5 \otimes I_5 = \boldsymbol{\Delta_{\text{row}} + \Delta_{\text{column}}}. \tag{11}$$

La misma matriz $\Delta_{5\times5}$ se obtiene con el método de elementos finitos (para elementos finitos lineales).

¡ Esta matriz también es un laplaciano de grafo ! El grafo es un cuadrado de 5 por 5 con 25 nodos. Tiene 20 aristas horizontales y 20 aristas verticales. Su matriz de incidencia A es de 40×25. Así su laplaciano de grafo $A^{\mathrm{T}}A$ es la misma matriz $\Delta_{5\times5}$ de 25×25. Esta matriz es semidefinida positiva pero no es invertible. Estudiaremos los laplacianos de grafos en la Sección IV.6.

El espacio nulo de $\Delta_{5\times5}$ contiene al vector de 25 unos. Ese vector es $\mathbf{1} \otimes \mathbf{1}$.

Autovectores y autovalores de $A \otimes B$ y $A \oplus B$

Sea $\boldsymbol{x}$ un autovector de A : $A\boldsymbol{x} = \lambda\boldsymbol{x}$. Sea $\boldsymbol{y}$ un autovector de B : $B\boldsymbol{y} = \mu\boldsymbol{y}$. Así el producto de Kronecker de $\boldsymbol{x}$ e $\boldsymbol{y}$ es un autovector de $A \otimes B$. **El autovalor es $\boldsymbol{\lambda\mu}$:**

$$(A \otimes B)(\boldsymbol{x} \otimes \boldsymbol{y}) = (A\boldsymbol{x}) \otimes (B\boldsymbol{y}) = (\lambda\boldsymbol{x}) \otimes (\mu\boldsymbol{y}) = \lambda\mu(\boldsymbol{x} \otimes \boldsymbol{y}). \tag{12}$$

Aquí A es de n por n y B es de N por N. Por lo tanto $\boldsymbol{x}$ es de n por 1 e $\boldsymbol{y}$ es de N por 1. Así $A \otimes B$ es una matriz cuadrada de tamaño nN y $\boldsymbol{x} \otimes \boldsymbol{y}$ es un vector con nN componentes. El mismo patrón también es cierto para las sumas de Kronecker cuando el mismo autovector $\boldsymbol{x} \otimes \boldsymbol{y}$:

$$(A \oplus B)(\boldsymbol{x} \otimes \boldsymbol{y}) = (A \otimes I_N)(\boldsymbol{x} \otimes \boldsymbol{y}) + (I_n \otimes B)(\boldsymbol{x} \otimes \boldsymbol{y}) = (\lambda + \mu)(\boldsymbol{x} \otimes \boldsymbol{y}). \tag{13}$$

El autovalor de $\boldsymbol{A \oplus B}$ es $\boldsymbol{\lambda + \mu}$. El vector $\boldsymbol{y}$ es sin duda un autovector de la matriz identidad I_N (con autovalor 1). El vector $\boldsymbol{x}$ es sin duda un autovector de I_n (con autovalor 1). Así la ecuación (13) para la suma de Kronecker se obtiene de dos aplicaciones de la ecuación (12) para el producto de Kronecker. $A \otimes I_N$ tiene autovalor λ por 1, y $I_n \otimes B$ tiene autovalor 1 por μ. El autovector en ambos casos es $\boldsymbol{x} \otimes \boldsymbol{y}$. Entonces sumamos para obtener la ecuación (13) con autovalor $\lambda + \mu$.

Separación de variables

La última página fue muy formal, pero la idea es simple e importante. Es el equivalente matricial del truco más útil para autofunciones de la ecuación de Laplace, y de todas las ecuaciones similares que tienen derivadas en x sumadas a derivadas en y.

$$\textbf{Autovalores } \alpha \textbf{ del Laplaciano} \qquad \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \alpha\, u(x,y). \tag{14}$$

El truco es buscar las funciones u separables como producto $\boldsymbol{u(x,y) = v(x)\, w(y)}$. Sustituimos vw por u en la ecuación de autovalores del laplaciano (14):

$$\textbf{Separación de variables } x \textbf{ e } y \qquad \left(\frac{d^2 v}{dx^2}\right) w(y) + v(x)\frac{d^2 w}{dy^2} = \alpha\, v(x)\, w(y). \tag{15}$$

Derivadas ordinarias en lugar de parciales, v solo depende de x, y w solo depende de y.

Divididamos la ecuación (15) por $\boldsymbol{vw}$. El lado izquierdo es una función solo de x más una función solo de y.

$$\frac{d^2 v/dx^2}{v(x)} + \frac{d^2 w/dy^2}{w(y)} = \alpha = \text{constante}. \tag{16}$$

Si esto es cierto para todo x y todo y entonces cada término es una constante (¡ *piense esto* !):

$$\boxed{\textbf{Ecuaciones separadas} \quad \frac{d^2 v}{dx^2} = \lambda\, v \;\text{ y }\; \frac{d^2 w}{dy^2} = \mu\, w \;\text{ con }\; \lambda + \mu = \alpha.} \tag{17}$$

Así $\lambda + \mu$ es un autovalor de la ecuación de Laplace (14). El laplaciano en el lado izquierdo de (14) es tan solo la **suma de Kronecker** de $A = \partial^2/\partial x^2$ y $B = \partial^2/\partial y^2$. El punto clave es que la autofunción $u(x,y)$ es el producto de dos autofunciones 1D $v(x)$ y $w(y)$, que es continuo en lugar de discreto, con derivadas en lugar de matrices de diferencias, y autofunciones en lugar de autovectores.

La ecuación diferencial parcial se reduce a dos ecuaciones diferenciales ordinarias (17).

La operación matriz a vector: vec(A)

A menudo queremos *vectorizar una matriz*. Comenzando con una matriz A de m por n, apilamos sus n columnas para obtener un vector columna **vec**(A) con mn componentes:

$$\textbf{vec}(A) = \begin{bmatrix} \text{columna } 1 \\ \vdots \\ \text{columna } n \end{bmatrix} \qquad \textbf{vec}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = \begin{bmatrix} a \\ c \\ b \\ d \end{bmatrix}. \tag{18}$$

Este vector de N^2 componentes se multiplica por la matriz de Fourier $2D$ (un producto de Kronecker de tamaño N^2) para producir el vector de N^2 coeficientes de Fourier. Y este vector de N^2 componentes se multiplica por la matriz del laplaciano de grafo (una suma de Kronecker) en la aproximación por diferencias finitas de la ecuación de Laplace en un cuadrado.

Entonces debemos ver cómo esta sencilla operación **vec** interactúa con la multiplicación de matrices. He aquí la identidad clave, cuando una matriz B se multiplica a la izquierda por una matriz A y a la derecha por una matriz C:

$$\boxed{\textbf{vec}(ABC) = (C^{\mathrm{T}} \otimes A)\,\textbf{vec}(B).} \tag{19}$$

Primero verifiquemos las dimensiones cuando las tres matrices A, B, C son de n por n. Así ABC es de n por n y su **vec** es de de n^2 por 1. El producto de Kronecker es de n^2 por n^2 y multiplica al vector **vec**(B) de n^2 por 1. Así el lado derecho de (19) también tiene n^2 componentes. Y si ABC es de m por p, entonces la matriz $C^{\mathrm{T}} \otimes A$ tiene mp filas como debería. *Bien.*

Observemos esta gran diferencia en la cantidad de operaciones. En el caso de n por n, las dos multiplicaciones en ABC necesitan $2n^3$ multiplicaciones y sumas. ¡ El producto de Kronecker a la derecha tiene n^4 elementos ! Entonces si no notamos que es un producto de Kronecker, esta multiplicación matriz-vector necesita n^4 multiplicaciones y sumas. Reformatear (reshape) es esencial para aprovechar la estructura de Kronecker:

A es de $m \times n$ y $mn = MN$ $\quad \boldsymbol{A'} = \textbf{reshape}\,(A, M, N)$ es de $M \times N$ $\quad \textbf{vec}\,(A') = \textbf{vec}\,(A)$

Si A es de 3×2, entonces $\boldsymbol{A'} = \textbf{reshape}\,(\boldsymbol{A}, \mathbf{1}, \mathbf{6})$ produce $A' = [a_{11} \quad a_{21} \quad a_{31} \quad a_{12} \quad a_{22} \quad a_{32}]$.

Necesitamos entender la identidad **vec** (19). Comencemos con el caso en que $B = [\boldsymbol{x}_1 \;\; \boldsymbol{x}_2]$ tiene 2 columnas y C es de 2 por 2. El lado derecho de (19) es simplemente una multiplicación matriz-vector $K\boldsymbol{x}$ donde $\boldsymbol{x} = \textbf{vec}\,(B)$ y reconocemos que K es el producto de Kronecker $C^{\mathrm{T}} \otimes A$. Esa multiplicación produce un vector $\boldsymbol{y}$:

$$\boldsymbol{y} = K\boldsymbol{x} = \left(C^{\mathrm{T}} \otimes A\right) \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} = \begin{bmatrix} c_{11}A & c_{21}A \\ c_{12}A & c_{22}A \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} = \begin{bmatrix} c_{11}A\boldsymbol{x}_1 + c_{21}A\boldsymbol{x}_2 \\ c_{12}A\boldsymbol{x}_1 + c_{22}A\boldsymbol{x}_2 \end{bmatrix} \tag{20}$$

Este vector $\boldsymbol{y}$ es exactamente el lado izquierdo de la identidad (19) : **vec** (ABC) con $B = [\boldsymbol{x}_1 \;\; \boldsymbol{x}_2]$ es

$$\textbf{vec}\left(\begin{bmatrix} A\boldsymbol{x}_1 & A\boldsymbol{x}_2 \end{bmatrix} \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}\right) = \textbf{vec}\,\begin{bmatrix} c_{11}A\boldsymbol{x}_1 + c_{21}A\boldsymbol{x}_2 & c_{12}A\boldsymbol{x}_1 + c_{22}A\boldsymbol{x}_2 \end{bmatrix}$$

Así si tenemos un sistema lineal $K\boldsymbol{x} = \boldsymbol{b}$ con una matriz de Kronecker K, (19) reducirá su tamaño.

Los valores de los píxeles en una imagen $2D$ (de n por n) son apilados por **vec** en un vector columna (de longitud n^2 o $3n^2$ con color RGB). Un vídeo con T cuadros (frames) tiene una secuencia de imágenes en T momentos diferentes. Así **vec** apila el vídeo en un vector columna de Tn^2 componentes.

Referencia

C. Van Loan, *The ubiquitous Kronecker product*, J. Comp. Appl. Math. **123** (2000) 85–100.
También *The Kronecker Product*, https://www.cs.cornell.edu/cv/ResearchPDF/KPhist.pdf

Problemas de la Sección IV.3

1 Una persona con una mirada matricial puede preferir ver todos los n autovectores de A en una matriz de autovectores: $AX = X\Lambda_A$. De la misma forma $BY = Y\Lambda_B$. Así el producto de Kronecker $X \otimes Y$ de tamaño nN es la matriz de autovectores tanto para $A \otimes B$ como para $A \oplus B$:

$$(A \otimes B)(X \otimes Y) = (X \otimes Y)(\Lambda_A \otimes \Lambda_B) \text{ y } (A \oplus B)(X \otimes Y) = (X \otimes Y)(\Lambda_A \oplus \Lambda_B).$$

Los autovalores de $\boldsymbol{A \otimes B}$ son los nN productos $\lambda_i \mu_j$ (todo λ *multiplicado por* todo μ). Los autovalores de $\boldsymbol{A \oplus B}$ son las nN sumas $\boldsymbol{\lambda_i + \mu_j}$ (todo λ *sumado a* todo μ).

Si A y B son de n por n, ¿cuándo es $A \oplus B$ invertible? ¿Y si (autovalor de A) $=$ $-$ (autovalor de B)? Halle matrices de 2×2 tales que $A \oplus B$ tenga rango 3 y no rango 4.

2 Demuestre: si A y B son simétricas definidas positivas, también lo son $A \otimes B$ y $A \oplus B$.

3 Halle una permutación P que verifique $P(A \otimes B) = (B \otimes A)P$. ¿Tienen $A \otimes B$ y $B \otimes A$ los mismos autovalores?

4 Supongamos que se quiere calcular $\boldsymbol{y} = (F \otimes G)\,\boldsymbol{x}$ donde $\boldsymbol{x} = \mathsf{vec}(X)$. La matriz F es de m por n, y G es de p por q. La matriz $F \otimes G$ es de mp por nq, la matriz X es de q por n, el vector $\boldsymbol{x}$ es de nq por 1. Muestre que este código encuentra la $\boldsymbol{y} = (F \otimes G)\,\boldsymbol{x}$ correcta:

$$\begin{aligned} Y &= G \otimes X \otimes F^{\mathrm{T}} \\ \boldsymbol{y} &= \mathsf{reshape}(\boldsymbol{Y}, \boldsymbol{mp}, \mathbf{1}) \end{aligned}$$

5 Supongamos que se quiere resolver $(F \otimes G)\,\boldsymbol{x} = \boldsymbol{b}$ con F y G matrices invertibles (de n por n). Así $\boldsymbol{b}$ y $\boldsymbol{x}$ son de n^2 por 1. Muestre que esto es equivalente a calcular

$$X = G^{-1}B(F^{-1})^{\mathrm{T}} \text{ con } \boldsymbol{x} = \mathbf{vec}(X) \text{ y } \boldsymbol{b} = \mathbf{vec}(B).$$

En realidad esas matrices inversas no se calculan nunca. En su lugar, resolvemos dos sistemas:

Hallar Z a partir de $GZ = B$

Hallar X a partir de $XF^{\mathrm{T}} = Z$ o $FX^{\mathrm{T}} = Z^{\mathrm{T}}$

Muestre que el costo ahora es $O(n^3)$. El sistema más grande $(F \otimes G)\,\boldsymbol{x} = \boldsymbol{b}$ cuesta $O(n^6)$.

6 ¿Cómo se vería una imagen si sus píxeles provinieran de un producto de Kronecker $A \otimes B$?

7 ¿Cómo crearía una FFT bidimensional? Para una imagen de n por n, ¿cuántas operaciones necesita la transformada rápida de Fourier 2D?

IV.4. Transformadas de senos y cosenos a partir de sumas de Kronecker

Esta sección del libro presenta un ejemplo sobresaliente de una suma de Kronecker de la forma $\boldsymbol{K = I \otimes D + D \otimes I}$. La matriz D con $1, -2, 1$ aproxima la derivada segunda d^2/dx^2 en una dimensión (a lo largo de una recta). La suma de Kronecker la lleva a dos dimensiones (en un cuadrado). Esta gran matriz K de tamaño N^2 aproxima al Laplaciano $\partial^2/\partial x^2 + \partial^2/\partial y^2$.

Las sumas de Kronecker permiten crear una matriz K importante de manera muy conveniente. Los siguientes pasos hacen práctico trabajar con esta matriz grande. Aproximamos la ecuación diferencial parcial de Laplace con $(N+1)^2$ ecuaciones en diferencias :

$$\textbf{Laplaciano} \quad \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y) \qquad \textbf{Laplaciano discreto} \quad \frac{1}{h^2} KU = F \tag{1}$$

Las N^2 componentes de $\boldsymbol{U}$ estarán cerca del verdadero $u(x,y)$ en los N^2 puntos $x = ph$, $y = qh$ dentro de una grilla cuadrada. Aquí p va de 1 a N a lo largo de una fila de la grilla y q va de 1 a N a lo largo de una columna, con $h = \Delta x = \Delta y =$ paso de la grilla.

La dificultad está en el tamaño de la matriz K. **La solución a esta dificultad está en conocer y utilizar los $\boldsymbol{N^2}$ autovectores de esta matriz.**

En general los autovectores no se conocen. Este caso particular es una excepción (este es prácticamente el único ejemplo importante que hemos visto), y gracias a que los conocemos las ecuaciones lineales $KU = h^2 F$ se resuelven escribiendo $h^2 F$ y U como combinaciones de los autovectores v_i de K :

$$\boxed{h^2 F = b_1 v_1 + b_2 v_2 + \cdots \quad U = \frac{b_1}{\lambda_1} v_1 + \frac{b_2}{\lambda_2} v_2 + \cdots \quad \text{Entonces } KU = h^2 F.} \tag{2}$$

Cuando K multiplica U, los λ se cancelan ya que cada $Kv_i = \lambda_i v_i$. Así KU coincide con $h^2 F$.

Los autovectores u_j de la matriz tridiagonal D con $1, -2, 1$ en sus diagonales, que corresponde a la ecuación diferencial en una dimensión, resultan ser **vectores senos** discretos. Son puntos muestrales tomados de las autofunciones $\operatorname{sen} j\pi x$ of d^2/dx^2 :

$$\boxed{\textbf{Continuo y discreto} \quad \frac{d^2}{dx^2} \operatorname{sen} j\pi x = -j^2\pi^2 \operatorname{sen} j\pi x \quad Du_j = D \begin{bmatrix} \operatorname{sen} j\pi h/(N+1) \\ \vdots \\ \operatorname{sen} j\pi N h/(N+1) \end{bmatrix} = \Lambda_j u_j} \tag{3}$$

En la Sección IV.3 que trata sobre el producto de Kronecker, hemos hallado los autovectores v de la gran matriz $\boldsymbol{K = I \otimes D + D \otimes I}$. **Los autovectores de $\boldsymbol{K}$ son productos de Kronecker $\boldsymbol{v_{jk} = u_j \otimes u_k}$ de los autovectores senos de $\boldsymbol{D}$.** Esto es cierto en el caso continuo (para autofunciones). También es cierto en el caso discreto (para autovectores). Las componentes de v_{jk} son componentes de u_j multiplicadas por componentes de u_k :

Autofunciones continuas $v_{jk}(x,y) = (\operatorname{sen} j\pi x)(\operatorname{sen} k\pi y)$ **Autovectores discretos** $v_{jk}(p,q) = \left(\operatorname{sen} \dfrac{\pi jp}{N+1}\right)\left(\operatorname{sen} \dfrac{\pi kq}{N+1}\right)$

Nuestro éxito depende del paso final, el más crucial. **Usando la transformada rápida de Fourier, podemos llevarlo a cabo con estos autovectores.** La matriz de Fourier en la Sección IV.1 tiene exponenciales complejas. Su parte real tiene cosenos (que aparecen en la transformada discreta de cosenos). La parte imaginaria de F produce la transformada discreta de seno (DST, por su sigla en inglés). La matriz DST contiene los autovectores $\boldsymbol{u}_j$ de D. La matriz DST de dos dimensiones contiene los autovectores $\boldsymbol{v}_{jk}$ de la matriz de diferencias K del laplaciano.

La FFT ejecuta los cálculos de la ecuación (2) realizando $O(N^2 \log_2 N)$ operaciones :

(1) Una transformada rápida 2D FFT $\otimes$ FFT para hallar los coeficientes b_{jk} de los senos de $h^2\boldsymbol{F}$

(2) Una división de cada b_{jk} por el autovalor $\lambda_{jk} = \lambda_j(D)\ \lambda_k(D)$

(3) Una transformada inversa rápida 2D para encontrar $\boldsymbol{U}$ a partir de sus coeficientes b_{jk}/λ_{jk} de los senos.

Este algoritmo resuelve la ecuación de Laplace en un cuadrado. Los valores de borde están prescritos en los cuatro lados. Está codificado en FISHPACK. El nombre es un mal juego de palabras de la traducción de Poisson, quien agregó un término fuente a la ecuación de Laplace : $\partial^2 u/\partial x^2 + \partial^2 u/\partial y^2 = f(x,y)$. FISHPACK también permite la condición de borde "libre" o "natural" o de "Neumann" $\partial u/\partial n = 0$ con autovectores cosenos, como así también la condición de borde "fija" o "esencial" o de "Dirichlet" $u = u_0$ con autovectores senos.

La diferencia entre fija y libre aparece en la primera y última fila de D en una dimensión, y luego en las filas correspondientes a nodos del borde en la suma de Kronecker $K = D \oplus D$ de dos dimensiones :

$$\boldsymbol{D}_{\textbf{fija}} = \begin{bmatrix} \mathbf{-2} & 1 & & \\ 1 & -2 & 1 & \\ & \cdot & \cdot & \cdot \\ & & 1 & \mathbf{-2} \end{bmatrix} \qquad \boldsymbol{D}_{\textbf{libre}} = \begin{bmatrix} \mathbf{-1} & 1 & & \\ 1 & -2 & 1 & \\ & \cdot & \cdot & \cdot \\ & & 1 & \mathbf{-1} \end{bmatrix} \tag{4}$$

Los autovectores de $D_{\textbf{fija}}$ nos llevan a la transformada discreta de senos. Los autovectores de $D_{\textbf{libre}}$ nos llevan a la transformada discreta de cosenos. Pero una aplicación diferente es demasiado interesante como para perdérsela. En lugar de resolver la ecuación en diferencias del laplaciano con condiciones de borde libres, aplicaremos la transformada discreta de cosenos (DCT, por su sigla en inglés) bidimensional para la compresión de imágenes.

En ese mundo el algoritmo basado en la DCT se conoce como JPEG.

La transformada discreta de cosenos en JPEG

Las letras JPEG se refieren a Joint Photographic Experts Group, el grupo que estableció una familia de algoritmos que comienzan con valores de píxeles (escala de grises del 0 a 255, o las intensidades de rojo, azul y verde para cada píxel). El código produce un archivo de imagen comprimido en el formato .jpg. El algoritmo puede verse en dos etapas :

El paso 1 es una transformación lineal de la matriz de los valores de cada píxel. Al comienzo los valores de la escala de grises están muy correlacionados ya que los píxeles adyacentes suelen tener valores similares. La transformada produce números con información más independiente.

Ejemplo : tomamos el promedio y la diferencia entre dos valores vecinos. Cuando la diferencia es pequeña, podemos transmitir menos bits y el ojo humano nunca se dará cuenta.

El paso 2 es una compresión no lineal y la cuantización de la señal transformada. La compresión mantiene solo los valores que son significativos visualmente. La cuantización convierte la señal comprimida en una secuencia de bits, listos para la transmisión rápida. Luego el receptor usa esos bits para reconstruir una imagen que es muy similar a la original.

El paso 1 a menudo usa una transformada discreta de cosenos. Trabaja por separado en bloques de 8×8 de la imagen. (JPEG2000 ofrece una transformada de ondículas pero esta opción no se ha adoptado de forma generalizada.) **Cada bloque de 64 valores en escala de grises lleva a un bloque de 64 coeficientes de coseno**: una transformada que no pierde nada. La transformada inversa recupera los bloques originales de la imagen.

Pero antes de invertir, comprimiremos y cuantizaremos esos 64 números. En este paso se pierde información que nuestros ojos no pueden percibir.

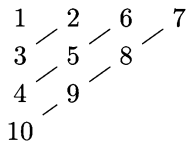
Las transformadas discretas de cosenos están desarrolladas en *SIAM Review* de 1999 (volumen **41**, páginas 135-147). Se diferencian en sus condiciones de borde. La elección más popular es DCT-2 con estos 8 vectores base ortogonales en cada dimensión :

DCT-2 La j-ésima componente del k-ésimo vector es $\cos\left(j+\frac{1}{2}\right)k\frac{\pi}{8} \quad j,k=1,\ldots,8$

Esos 64 números van a una matriz C de 8 por 8. Luego, la matriz correspondiente a la transformada de coseno $2D$ es el producto Kronecker $C \otimes C$ de tamaño 8^2. Sus columnas son ortogonales. Actúa sobre cada bloque de 8×8 de la imagen para dar un bloque de coeficientes de Fourier de cosenos de 8×8, que nos da la combinación correcta de vectores base de cosenos para reconstruir el bloque original en la imagen.

Pero no buscamos la reconstrucción perfecta. El paso 2 descarta información que no necesitamos. El paso 1 produce coeficientes c_{jk} de cosenos, de tamaños muy variados, a menudo números c más pequeños para frecuencias j, k más grandes. El archivo **https://cs.stanford.edu/people/eroberts/courses/soco/projects/data-compression/lossy/jpeg/dct.htm** muestra un bloque de 8×8 típico antes y después del paso DCT. Los coeficientes de cosenos están listos para el paso 2: compresión y cuantización.

Primero, cada bloque de 64 coeficientes va en un vector de tamaño 64×1 (pero no con el comando vec). Una secuencia zigzagueante es mejor para mantener al principio los coeficientes grandes. Si terminamos con una secuencia de casi nulos, pueden comprimirse a cero y solo transmitimos la longitud de la secuencia: la cantidad de ceros. He aquí el comienzo del orden zigzagueante para los 64 coeficientes :



Las frecuencias mayores vienen después en el orden zigzagueante y generalmente con coeficientes más pequeños. A menudo podemos reescalar con seguridad esos números antes de que el paso de redondeo dé q_{jk} :

Ejemplo de cuantización $\quad q_{jk} = \dfrac{c_{jk}}{j+k+3}$ redondeado al entero más cercano

Ahora cada bloque se representa por 64 enteros q_{jk}. Para mayor eficiencia transmitimos esos números después de codificar. El receptor reconstruye cada bloque de la imagen de manera aproximada.

En una imagen a color, 64 se vuelve 192. Cada píxel combina tres colores. Pero es probable que rojo, verde y azul no sean las mejores coordenadas. Una mejor primera coordenada nos indica el brillo. Las otras dos coordenadas nos indican la diferencia colorimétrica (*"chrominance"* en inglés). El objetivo es tener tres números estadísticamente independientes.

La reconstrucción completa, al ensamblar los bloques, la menudo muestra dos efectos no deseados. Uno es el **cuadriculado** (*blocking* en inglés) : los bloques no se unen suavemente. (Puede ser que ya haya visto esto en alguna imagen impresa excesivamente comprimida.) El segundo efecto es el **anillado** (*ringing* en inglés) : oscilaciones rápidas en los bordes de la imagen. Estas oscilaciones son famosas en el fenómeno de Gibbs cuando ocho cosenos aproximan una función salto.

Para una imagen de alta calidad luego de la compresión DCT, podemos cancelar los efectos cuadriculado y anillado. Para la televisión de alta definición, la compresión es esencial, ya que si no se utilizara, habría que estar pendientes de demasiados bits.

El estándar de DCT fue impuesto por el grupo de expertos JPEG. Luego se construyeron los equipos y el software para que el sistema de procesamiento de imágenes fuera efectivo.

Problemas de la Sección IV.4

1 ¿ Cuáles son los autovalores Λ_j de D en la ecuación (3) ?

2 ¿ Cuáles son los autovalores λ_i y autovectores $\boldsymbol{v}_i$ de $K = I \otimes D + D \otimes I = D \oplus D$?

3 ¿ Cuál sería el operador de Laplace $K3$ en una grilla cúbica 3D ?

4 ¿ Cuál sería la matriz de Fourier $F3$ de N^3 por N^3 en 3D ? En 2D es $F \otimes F$.

IV.5. Matrices de Toeplitz y filtros invariantes respecto a desplazamiento

Una matriz de Toeplitz tiene diagonales constantes. La primera fila y columna definen el resto de la matriz porque contienen el primer elemento de cada diagonal. Las **matrices circulantes** son matrices de Toeplitz que satisfacen la condición "envolvente" adicional que las hace periódicas. Para matrices circulantes de 4×4, c_{-3} es idéntico a c_1 :

$$\textbf{Matriz de Toeplitz} \quad \boldsymbol{A} = \begin{bmatrix} a_0 & a_{-1} & a_{-2} & a_{-3} \\ \boldsymbol{a_1} & a_0 & a_{-1} & a_{-2} \\ a_2 & \boldsymbol{a_1} & a_0 & a_{-1} \\ a_3 & a_2 & \boldsymbol{a_1} & a_0 \end{bmatrix} \qquad \textbf{Matriz circulante} \quad \boldsymbol{C} = \begin{bmatrix} c_0 & c_3 & c_2 & \boldsymbol{c_1} \\ \boldsymbol{c_1} & c_0 & c_3 & c_2 \\ c_2 & \boldsymbol{c_1} & c_0 & c_3 \\ c_3 & c_2 & \boldsymbol{c_1} & c_0 \end{bmatrix}$$

Las matrices circulantes son perfectas para la transformada discreta de Fourier. Siempre $CD = DC$. Sus autovectores son *exactamente* las columnas de la matriz de Fourier de la Sección IV.2. Sus autovalores son exactamente los valores de $C(\theta) = \sum c_k e^{ik\theta}$ en los n ángulos equidistantes $\theta = 0, 2\pi/n, 4\pi/n, \ldots$ (donde $e^{in\theta} = e^{2\pi i} = 1$).

Las matrices de Toeplitz son casi perfectas. Son las que usamos para el procesamiento de señales y en las redes neurales convolucionales (**las CNN, por su sigla en inglés**). No son envolventes, por lo tanto el análisis de A está basado en el polinomio $A(\theta)$ con coeficientes $a_{1-n} \ldots a_0 \ldots a_{n-1}$:

Respuesta de frecuencia = símbolo de A	$\boldsymbol{A(\theta) = \sum a_k\, e^{ik\theta}}$
$A(\theta)$ es real cuando A es simétrica	$a_k\, e^{ik\theta} + a_k\, e^{-ik\theta} = 2a_k \cos k\theta$
$C(\theta)$ no es nulo cuando C es invertible	El símbolo para C^{-1} es $1/C(\theta)$

¡ $A(\theta) \neq 0$ no sirve para probar la invertibilidad de A ! Y A^{-1} no es de Toeplitz (las matrices triangulares son la excepción). Las matrices C circulantes son convoluciones cíclicas. Pero las matrices de Toeplitz son convoluciones no cíclicas con $\boldsymbol{a} = (a_{1-n} \ldots a_{n-1})$, seguidas de proyecciones :

Espacio $\boldsymbol{x}$ $\quad A\boldsymbol{x}$ $\quad$ = convolucionar $\boldsymbol{a} * \boldsymbol{x}$, luego mantener los componentes de 0 a $n-1$

Espacio $\boldsymbol{\theta}$ $\quad A\boldsymbol{x}(\theta)$ = multiplicar $A(\theta)\boldsymbol{x}(\theta)$, luego proyectar nuevamente a n coeficientes

Queremos usar el polinomio simple $A(\theta)$ para aprender sobre la matriz de Toeplitz A.

En muchos problemas la matriz de Toeplitz es **de banda**. La matriz solo tiene w diagonales por arriba y por debajo de la diagonal principal. Solo los coeficientes de a_{-w} a a_w pueden ser no nulos. Entonces el "ancho de banda" es w, con un total de $\boldsymbol{2w+1}$ **diagonales distintas de cero** :

$$\textbf{Tridiagonal de Toeplitz} \quad \textbf{Ancho de banda } \boldsymbol{w = 1} \qquad A = \begin{bmatrix} a_0 & a_{-1} & & \\ a_1 & a_0 & a_{-1} & \\ & a_1 & a_0 & a_{-1} \\ & & a_1 & a_0 \end{bmatrix}$$

Para entender las matrices tridiagonales de Toeplitz (y sus autovalores) de tamaño grande n solo necesitamos estudiar el símbolo $A(\theta) = a_{-1}e^{-i\theta} + a_0 + a_1 e^{i\theta}$. Se construye a partir de a_{-1}, a_0, a_1.

Matrices de Toeplitz : conceptos básicos

En procesamiento de señales una matriz de Toeplitz es un **filtro**. La multiplicación de matrices $A\boldsymbol{x}$ en el dominio temporal se traduce en multiplicación ordinaria $A(\theta)\boldsymbol{x}(\theta)$ en el dominio de frecuencia. Esa es la idea fundamental, *pero no es exactamente cierta*. La teoría de Toeplitz está muy cerca de un análisis de Fourier simple (una frecuencia a la vez), pero los bordes interfieren.

Una señal de longitud finita $\boldsymbol{x} = (x_0, \ldots, x_n)$ tiene bordes en 0 y en n. Esos bordes no permien tener una respuesta simple para cada frecuencia por separado. No podemos tan solo multiplicar por $A(\theta)$. En muchas aplicaciones, pero no en todas, este hecho inconveniente puede suprimirse. Así una matriz de Toeplitz produce esencialmente una convolución. Si queremos un **filtro pasobanda** (*bandpass filter*) que preserve las frecuencias $a \le \theta \le b$ y quite todas las otras frecuencias, entonces construimos $A(\theta)$ que esté cerca de 1 en esa banda y cerca de cero por fuera de la banda. (Una vez más, un filtro ideal con A infinitos unos y ceros es imposible.)

Las ecuaciones en diferencias finitas lineales con *coeficientes constantes* producen matrices de Toeplitz. Las ecuaciones no cambian a medida que transcurre el tiempo (**LTI** = Invarianza lineal con respecto al tiempo, por su sigla en inglés). No cambian en el espacio (**LSI** = Invarianza lineal con respecto al desplazamiento, por su sigla en inglés). Un ejemplo importante es la conocida matriz de diferencias segundas $-1, 2, -1$:

$$\text{Matriz } -1, 2, -1 \text{ tridiagonal con símbolo } \boldsymbol{A(\theta)} = -e^{-i\theta} + 2 - e^{i\theta} = \mathbf{2 - 2\cos\theta}.$$

El hecho de que $A(\theta) \ge 0$ nos dice que A es simétrica definida o semidefinida positiva. El hecho de que $A = 2 - 2 = 0$ cuando $\theta = 0$ nos dice que $\lambda_{\text{mín}}(A)$ se acercará a cero a medida que n aumente. La matriz de Toeplitz finita A es apenas definida positiva y la matriz de Toeplitz infinita es singular, ya que el símbolo satisface $A(\theta) = 0$ en $\theta = 0$.

La inversa de una matriz de Toeplitz A no suele ser de Toeplitz. Por ejemplo,

$$A^{-1} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}^{-1} = \frac{1}{4}\begin{bmatrix} 3 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 3 \end{bmatrix} \quad \text{no es de Toeplitz}$$

Pero Levinson encontró una manera de usar el patrón de Toeplitz en una recursión, reduciendo los típicos $O(n^3)$ pasos para la resolución de $A\boldsymbol{x} = \boldsymbol{b}$ a $O(n^2)$. Algoritmos superrápidos fueron propuestos luego, pero la "recursión de Levinson-Durbin" es mejor para n moderados. Los algoritmos superrápidos dan respuestas precisas (no exactas) para n grande, por ejemplo usando un precondicionador circulante.

Con frecuencia ocurre que la primera y la última fila no encajan en el patrón de Toeplitz de "invarianza respecto a desplazamiento". Los elementos en esas filas correspondientes a los extremos pueden ser distintos de a_0, a_1 y a_{-1}. Este cambio puede provenir de las *condiciones de borde* en una ecuación diferencial o en un filtro. Cuando A multiplica un vector $\boldsymbol{x} = (x_1, \ldots, x_n)$, la matriz de Toeplitz (corte pronunciado) supone que $x_0 = 0$ y $x_{n+1} = 0$ (*zero-padding*).

La completación con ceros (zero padding) puede dar una aproximación pobre en los extremos. Para una mejor aproximación, a menudo cambiamos las filas correspondientes a los extremos. Ya que se perturba el patrón diagonal constante en la matriz de Toeplitz, un análisis completo de esos cambios puede ser difícil.

Multiplicación rápida usando la FFT

Para matrices tridiagonales no necesitamos un algoritmo especial para multiplicar $A\boldsymbol{x}$. Esto requiere solo $3n$ multiplicaciones separadas. Pero para una matriz densa y completa, las propiedades de Toeplitz y circulantes permiten, con la transformada rápida de Fourier, una gran aceleración en el espacio θ.

Las matrices circulantes son convoluciones cíclicas. En el espacio θ, la multiplicación matriz-vector $C\boldsymbol{x}$ se convierte en $(\sum c_k e^{ik\theta})(\sum x_k e^{ik\theta})$. Este producto dará potencias de $e^{i\theta}$ que van por fuera del rango de $k = 0$ a $k = n-1$. **Esas frecuencias más altas y las frecuencias negativas son "solapas" (*aliases*) de frecuencias estándares**: son iguales en todos los $\theta = p2\pi/n$:

Solapamiento $\quad e^{in\theta} = 1$ y $e^{i(n+1)\theta} = e^{i\theta}$ y $e^{i(n+p)\theta} = e^{ip\theta}$ en $\theta = \dfrac{2\pi}{n}, \dfrac{4\pi}{n}, \ldots$

La convolución cíclica $\boldsymbol{c} \circledast \boldsymbol{x}$ lleva cada término de $(\sum c_k e^{ik\theta})(\sum x_k e^{ik\theta})$ de nuevo a un término con $0 \le k < n$. Por la transformada rápida de Fourier, una multiplicación circulante $C\boldsymbol{x}$ solo necesita $O(n \log_2 n)$ pasos debido a la convolución.

Las convoluciones cíclicas $\quad \boldsymbol{c} \circledast \boldsymbol{x}$ **y** $\boldsymbol{c} \circledast \boldsymbol{d}$ **dan los elementos de** $\boldsymbol{Cx}$ **y de** $\boldsymbol{CD}$**.**

Una multiplicación con una matriz de Toeplitz $A\boldsymbol{x}$ no es cíclica. Las frecuencias más altas en $A(\theta)x(\theta)$ no se ajustan a la perfección con las frecuencias más bajas. Pero podemos usar una multiplicación cíclica y una circulante con un truco de duplicación : **insertar $\boldsymbol{A}$ en una matriz circulante $\boldsymbol{C}$.**

$$\boldsymbol{A} = \begin{bmatrix} a_0 & a_{-1} & \cdots & a_{1-n} \\ a_1 & & & \vdots \\ \vdots & & & a_{-1} \\ a_{n-1} & \cdots & a_1 & a_0 \end{bmatrix} \qquad \boldsymbol{C} = \begin{bmatrix} \boldsymbol{A} & * \\ * & * \end{bmatrix} = \begin{bmatrix} a_0 & a_{-1} & \cdots & a_{1-n} & a_{n-1} & \cdots & a_1 \\ a_1 & & & \vdots & & & \vdots \\ \vdots & & & a_{-1} & & & a_{n-1} \\ a_{n-1} & \cdots & a_1 & a_0 & & & a_{1-n} \\ a_{1-n} & & & & & & \vdots \\ \vdots & & & & & & a_{-1} \\ a_{-1} & \cdots & a_{1-n} & a_{n-1} & \cdots & a_1 & a_0 \end{bmatrix}$$

Para calcular $A\boldsymbol{x}$ de tamaño n, podemos usar esta matriz circulante C de tamaño $2n-1$.

1. Agregar $n-1$ ceros para extender $\boldsymbol{x}$ a un vector $\boldsymbol{X}$ de tamaño $2n-1$.
2. Multiplicar $\boldsymbol{CX}$ con la transformada rápida de Fourier (convolución cíclica).
3. Las primeras n componentes de $\boldsymbol{CX}$ producen el $\boldsymbol{Ax}$ deseado.

Si se prefiere el tamaño $2n$ para $\boldsymbol{C}$, una diagonal de a_0 puede ir entre a_{1-n} y a_{n-1}.

Autovalores de Toeplitz y el teorema de Szegö

Encontramos los autovalores exactos de las matrices circulantes en la Sección IV.2. Conocemos los autovectores de antemano. Son siempre los mismos, las columnas de la matriz de Fourier F. Los autovalores de C son las componentes del vector $F\boldsymbol{c}$. Son los valores de $C(\theta)$ en n puntos equidistantes. En otras palabras, los autovalores son la transformada de Fourier discreta de la columna 0 de la matriz circulante C.

Como siempre, las fórmulas circulantes son exactas y las fórmulas de Toeplitz son aproximadas. Veremos integrales en lugar de valores puntuales de $A(\theta)$. *Las fórmulas solo se vuelven exactas a medida que la matriz de Toeplitz se vuelve infinita.* Así tenemos un límite cuando $n \to \infty$ incluido en el teorema de Szegö sobre los autovalores de A. Dos casos especiales y luego el teorema completo.

Szegö Cuando $n \to \infty$, la traza y el logaritmo del determinante de una matriz de Toeplitz A satisfacen

$$\lim_{n\to\infty} \frac{1}{n} \text{traza}\,(A) \quad = \frac{1}{2\pi}\int_0^{2\pi} A(\theta)\, d\theta = a_0 \tag{1}$$

$$\lim_{n\to\infty} \frac{1}{n} \log(\det A) = \frac{1}{2\pi}\int_0^{2\pi} \log(A(\theta))\, d\theta \tag{2}$$

La traza y el determinante de una matriz son la suma y producto de los autovalores. Vemos su media aritmética en (1) y su media geométrica en (2) : logaritmo del determinante = suma de $\log \lambda_k$. Esos dos límites son los casos más importantes $F(\lambda) = \lambda$ y $F(\lambda) = \log \lambda$ del teorema completo, que permite cualquier función continua F de los autovalores de A :

Teorema de Szegö $\lim_{n\to\infty} \dfrac{1}{n}\displaystyle\sum_{k=0}^{n-1} F(\lambda_k) = \dfrac{1}{2\pi}\displaystyle\int_0^{2\pi} F(A(\theta))\, d\theta.$ (3)

El control sobre $A(\theta)$ proviene de la condición de Wiener $\sum |a_k| < \infty$. En el sitio web de Stanford podemos leer notas excelentes al respecto: **ee.stanford.edu/~gray/toeplitz.pdf**.

Las notas del Profesor Gray también desarrollan una importante aplicación para procesos aleatorios de tiempo discreto. Cuando el proceso es débilmente estacionario, los estadísticos se mantienen iguales en todo paso temporal. *La matriz de covarianza entre resultados a tiempos t y T solo depende de la diferencia $T - t$.* **Así la matriz de covarianza es de Toeplitz = invariante respecto a desplazamientos**.

Antes de aplicarlo, mencionaremos tres temas clave más de la teoría de matrices de Toeplitz :

- La fórmula de Gohberg-Semencul para A^{-1} (Gohberg fue realmente extraordinario)
- La factorización de Wiener-Hopf de sistemas infinitos $A_\infty \boldsymbol{x} = \boldsymbol{b}$ (¡ Wiener también !)
- La prueba de invertibilidad de A_∞ es $A_\infty(\theta) \neq 0$ y el índice (*winding number*) $= 0$

Un **desplazamiento de un paso infinito** modela las dificultades asociadas a $n = \infty$. Tiene una diagonal de unos por encima o debajo de la diagonal principal. **Su símbolo es tan solo $e^{i\theta}$ o $e^{-i\theta}$** (nunca cero). Pero $e^{i\theta}$ da vueltas alrededor de alrededor de cero a medida que θ va de 0 a 2π, y la matriz de desplazamiento infinito $S(x_0, x_1, \ldots) = (x_1, x_2, \ldots)$ o $S\boldsymbol{x} = (0, x_0, x_1, \ldots)$ no es invertible.

Filtros pasobajo en el procesamiento de señales

Un filtro es una convolución: multiplicar por una matriz de Toeplitz. Atravesando un "filtro pasobajo", el vector constante $\boldsymbol{x} = (\ldots, 1, 1, 1, 1, \ldots)$ puede salir inalterado: $A\boldsymbol{x} = \boldsymbol{x}$. Pero una señal oscilante a alta frecuencia como $\boldsymbol{y} = (\ldots, -1, 1, -1, 1, \ldots)$ tiene $A\boldsymbol{y} \approx \boldsymbol{0}$. Eso es lo que se obtiene si $A(0) = \Sigma a_k = 1$ y $A(\pi) \approx 0$. He aquí un ejemplo de filtro pasobajo.

Filtro pasobajo que promedia $(A\boldsymbol{x})_n = \dfrac{\mathbf{1}}{\mathbf{4}}\, x_{n+1} + \dfrac{\mathbf{1}}{\mathbf{2}}\, x_n + \dfrac{\mathbf{1}}{\mathbf{4}}\, x_{n-1}.$

La matriz de Toeplitz A es simétrica. Sus tres diagonales tienen los elementos $\frac{1}{4}, \frac{2}{4}, \frac{1}{4}$. Su símbolo $A(\theta)$ (respuesta de frecuencia) es real y $A(\theta) = 1$ en la frecuencia $\theta = 0$: filtro pasobajo.

$$\textbf{respuesta de frecuencia} \qquad A(\theta) = \frac{1}{4}\left(e^{-i\theta} + 2 + e^{i\theta}\right) = \frac{1}{2}\left(1 + \cos\theta\right) \geq 0.$$

La mayor frecuencia $\theta = \pi$ da la señal más-menos infinitamente larga $\boldsymbol{y} = (\ldots, -1, 1, -1, 1, \ldots)$. Esa señal tiene $A\boldsymbol{y} = \boldsymbol{0}$. Es filtrada. Una componente típica de $A\boldsymbol{y}$ es $-\frac{1}{4} + \frac{1}{2} - \frac{1}{4} = 0$. Y también vemos esto a partir del símbolo: $\boldsymbol{A(\theta) = \frac{1}{2}(1 + \cos\theta)}$ **es cero en** $\boldsymbol{\theta = \pi}$.

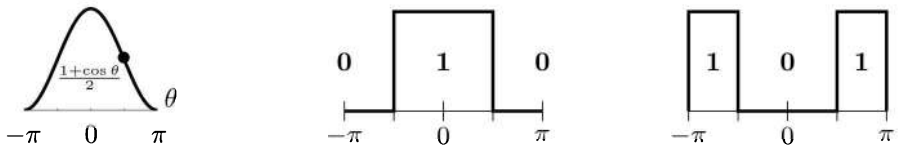


Figura IV.3: Respuestas de frecuencia $A(\theta)$: filtro pasobajo corto, pasobajo ideal, pasoalto ideal.

Vemos dos *filtros ideales* en la Figura IV.3. Solo se logran con infinitas diagonales no nulas en la matriz A ya que un polinomio ordinario no puede mantenerse constante. Los números a_k en las diagonales de A son los coeficientes de Fourier de $A(\theta)$.

En la práctica, los filtros son un compromiso entre cortos e ideales. Los *filtros equiripple* son un favorito natural, oscilan alrededor de 1 y alrededor de 0. Todas las ondas (ripples) (oscilaciones alrededor del ideal) tienen la misma altura. La altura decrece a medida que usamos más coeficientes a_k. El filtro es más afilado, la caída de 1 a 0 es más pronunciada, pero calcular $A\boldsymbol{x}$ lleva más tiempo.

Promedios, diferencias y ondículas

Los filtros pasobajo son *promedios móviles*. Los filtros pasoalto son *diferencias móviles*. La palabra "móvil" significa que una ventana se mueve a lo largo del vector $\boldsymbol{x}$. Suponga que la ventana solo permite ver 3 componentes de la señal $\boldsymbol{x}$. El filtro pasobajo las multiplica por $\frac{1}{4}, \frac{1}{2}, \frac{1}{4}$ y suma, para encontrar la señal promediada $A\boldsymbol{x}$. Un filtro pasoalto podría alternar esas señales para producir $-\frac{1}{4}, \frac{1}{2}, -\frac{1}{4}$. Así D toma diferencias segundas en lugar de promedios de promedios.

La ventana móvil crea una convolución = matriz de Toeplitz A o D:

$$(A\boldsymbol{x})_n = \frac{1}{4}x_{n-1} + \frac{1}{2}x_n + \frac{1}{4}x_{n+1} \qquad (D\boldsymbol{x})_n = -\frac{1}{4}x_{n-1} + \frac{1}{2}x_n - \frac{1}{4}x_{n+1}$$

La idea de las "ondículas" es usar ambos filtros: promedios A y diferencias D.
Reducir la resolución: eliminamos $(A\boldsymbol{x})_n$ y $(D\boldsymbol{x})_n$ para n impar. Mantenemos semi señales.
Ahora pasamos la señal $(A\boldsymbol{x})_{2n}$, cuya longitud es la mitad, a través de ambos filtros A, D. **Reducimos la resolución de nuevo.**

Transformada de ondículas con reducción de resolución A **para frecuencias bajas,** D **para frecuencias altas**

$$\begin{array}{ccccc} (\boldsymbol{x})_n & \longrightarrow & (A\boldsymbol{x})_{2n} & \longrightarrow & (AA\boldsymbol{x})_{4n} \\ & \searrow & & \searrow & (DA\boldsymbol{x})_{4n} \\ & & (D\boldsymbol{x})_{2n} & & \end{array}$$

El largo total de la señal no es afectado por su transformada de ondícula $(A^2\boldsymbol{x})_{4n}, (DA\boldsymbol{x})_{4n}, (D\boldsymbol{x})_{2n}$. Pero la señal se separa en bloques de frecuencias bajo-bajo, alto-bajo y alto. Comprimimos esas piezas por separado, y las frecuencias altas son las más comprimidas.

Para invertir la transformada de ondículas recorremos inversamente el diagrama de flujo. Subimos la resolución de cada pieza de la transformada agregando ceros. Luego invertimos las flechas y ensamblamos la señal original $(\boldsymbol{x})_n$ a partir de los bloques $AA\boldsymbol{x}, DA\boldsymbol{x}, D\boldsymbol{x}$ de su transformada:

Transformada de ondícula inversa

$$\begin{array}{ccccc} (AA\boldsymbol{x})_{4n} & \longrightarrow & (A\boldsymbol{x})_{2n} & \longrightarrow & (\boldsymbol{x})_n \\ (DA\boldsymbol{x})_{4n} & \nearrow & (D\boldsymbol{x})_{2n} & \nearrow & \end{array}$$

Este proceso usa los filtros cuidadosamente elegidos A y D para reducir la longitud de una señal sin perder la información que queremos ver y oir. Las mejores transformadas de ondículas se ajustan para que A y D produzcan matrices ortogonales o simétricas. Los filtros de cuatro coeficientes de Daubechies son un favorito, con estas diagonales en A y en D:

$$\boldsymbol{a_k} = 1+\sqrt{3} \quad 3+\sqrt{3} \quad 3-\sqrt{3} \quad 1-\sqrt{3} \quad \boldsymbol{d_k} = \sqrt{3}-1 \quad 3-\sqrt{3} \quad -3-\sqrt{3} \quad \sqrt{3}+1$$

Como siempre, las matrices de Toeplitz finitas con estos elementos en sus diagonales, deben ser ajustadas en los bordes. Toeplitz tiene *ceros* por fuera de la matriz, pero las ondículas útiles a menudo usan *reflexiones*.

Problemas de la Sección IV.5

1 Muestre que las ecuaciones (2) y (3) son la misma cuando $F(\lambda) = \log \lambda$.

2 Tomando $F(\lambda) = \lambda^2$ el teorema de Szegö (3) da el límite del autovalor promedio de A^2.

(a) Muestre, tomando el cuadrado de A con diagonales a_{-1}, a_0, a_1, que el símbolo de A^2 es $(A(\theta))^2$.

(b) Integrar ese polinomio $(A(\theta))^2$ de 0 a 2π produce su término constante. ¿Cuál es ese término para la matriz $-1, 2, -1$?

La factorización $A = LU$ de la matriz simétrica ($n = 4$) de diferencias segundas $-1, 2, -1$ es

$$\begin{bmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ & & -1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ -1/2 & 1 & & \\ & -2/3 & 1 & \\ & & -3/4 & 1 \end{bmatrix} \begin{bmatrix} 2/1 & -1 & & \\ & 3/2 & -1 & \\ & & 4/3 & -1 \\ & & & 5/4 \end{bmatrix}$$

3 Verifique que $(LU)_{44} = 2$ y $\det A = 5 = n + 1$.

4 Estos factores de la matriz de Toeplitz $-1, 2, -1$ no son matrices de Toeplitz. Verifique que los símbolos límites $(-e^{-i\theta} + 1)$ y $(-e^{i\theta} + 1)$ multiplicados dan el símbolo correcto de A.

5 El símbolo $5 - 2e^{i\theta} - 2e^{-i\theta}$ se factoriza en $2 - e^{i\theta}$ por $2 - e^{-i\theta}$. Cuando la matriz simétrica de Toeplitz S con diagonales $-2, 5, -2$ se factoriza en $S = A^{\mathrm{T}}A$ con A triangular superior, ¿a qué esperaría que converja la última columna cuando $n \to \infty$? (A solo tendrá dos diagonales no nulas.)

6 Use el comando de Cholesky $\mathsf{A} = \mathsf{chol(S)}$ con la matriz S que tiene $-2, 5 - 2$ en el Problema 5, para verificar que la última fila y columna de A^{T} y A se aproximan a los límites previstos.

IV.6. Grafos y laplacianos y las leyes de Kirchhoff

Un grafo consiste de un conjunto de nodos y un conjunto de aristas entre esos nodos. Este es el modelo más importante de la matemática aplicada discreta : simple, útil y general. Esa palabra *discreta* se usa en contraste con *continua* : tenemos vectores en lugar de funciones, tomamos diferencias y sumas en lugar de derivadas e integrales, dependemos del álgebra lineal en lugar del cálculo.

Comencemos con la **matriz de incidencia** del grafo. Con m aristas y n nodos, la matriz de incidencia A es de m por n. La fila i de A corresponde a la arista i del grafo. Si esa arista va del nodo j al nodo k, entonces la *fila i de A tiene -1 en su columna j y $+1$ en la columna k*. Así cada fila de A suma cero y $(1, 1, \dots, 1)$ pertenece al espacio nulo.

El espacio nulo de A contiene todos los vectores constantes $x = (c, c, \dots, c)$.

Suponemos que el grafo es conexo, si no hay una arista del nodo j al nodo k, entonces al menos hay un camino de aristas conectándolos. He aquí las dimensiones de los cuatro subespacios :

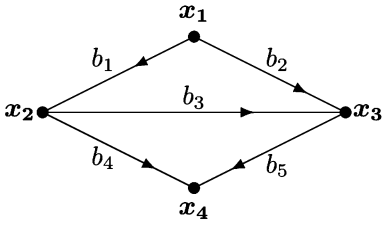
$$\textbf{dim } \mathbf{N}(A) = \mathbf{1} \quad \textbf{dim } \mathbf{C}(A) = \textbf{dim } \mathbf{C}(A^{\mathrm{T}}) = \boldsymbol{n - 1} \quad \textbf{dim } \mathbf{N}(A^{\mathrm{T}}) = \boldsymbol{m - n + 1}$$

El vector constante $\mathbf{1} = (1, 1, \dots, 1)$ es la opción más sencilla para la base del espacio nulo. Entonces el espacio fila contiene todos los vectores $\boldsymbol{x}$ con $x_1 + x_2 + \cdots + x_n = 0$ (así $\boldsymbol{x}$ es ortogonal a $\mathbf{1}$). Para encontrar **bases** para los cuatro subespacios podemos usar árboles y bucles :

$\mathbf{C}(A^{\mathrm{T}})$	$n - 1$ filas de A que producen un árbol en el grafo (un árbol no tiene bucles)
$\mathbf{C}(A)$	las primeras $n - 1$ columnas de A (o cualesquiera $n - 1$ columnas de A)
$\mathbf{N}(A^{\mathrm{T}})$	fluye alrededor de los $m - n + 1$ bucles pequeños en el grafo : vea la ecuación (3)

Si se desean bases ortogonales, hay que elegir los vectores singulares a derecha y a izquierda en $A = U\Sigma V^{\mathrm{T}}$.

Ejemplo de la Sección I.3 $m = 5$ aristas y $n = 4$ nodos



$$\boldsymbol{A} = \begin{array}{c} \\ \textbf{aristas} \\ \textbf{nodos} \end{array} \begin{array}{c} \begin{bmatrix} \mathbf{-1} & \mathbf{1} & 0 & 0 \\ \mathbf{-1} & 0 & \mathbf{1} & 0 \\ 0 & \mathbf{-1} & \mathbf{1} & 0 \\ 0 & \mathbf{-1} & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{-1} & \mathbf{1} \end{bmatrix} \\ \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \end{array} \begin{array}{c} \\ \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}$$

La **matriz laplaciana del grafo** $\boldsymbol{L = A^{\mathrm{T}}A}$ es cuadrada y simétrica *semidefinida positiva*: $A^{\mathrm{T}}A$ tiene $n - 1$ autovalores positivos $\lambda = \sigma^2$ y un autovalor nulo (porque $A\mathbf{1} = \mathbf{0}$). La forma especial $\boldsymbol{A^{\mathrm{T}}A = D - B}$ se destaca en este ejemplo con 5 aristas y 4 nodos :

Laplaciana
Una arista faltante

$$\boldsymbol{A^{\mathrm{T}}A} = \begin{bmatrix} \mathbf{2} & -1 & -1 & \mathbf{0} \\ -1 & \mathbf{3} & -1 & -1 \\ -1 & -1 & \mathbf{3} & -1 \\ \mathbf{0} & -1 & -1 & \mathbf{2} \end{bmatrix} = \boldsymbol{D - B}$$

La **matriz de grados es** $D = \text{diag}\,(2,3,3,2)$. Cuenta las aristas en los nodos $1,2,3,4$.
La **matriz de adyacencias** B tiene elementos 0 y 1. Una arista de j a k produce $b_{jk} = 1$.
Un **grafo completo** (todas las aristas presentes) tiene $D = (n-1)\,I$ y $B =$ todos unos menos I.

Si todo par de nodos está conectado por una arista, el grafo es **completo**. Tendrá $m = (n-1) + (n-2) + \cdots + 1 = \frac{1}{2}n(n-1)$ aristas. B es una matriz de unos (menos I). Todos los grados son $n-1$, así que $D = (n-1)I$. En el otro extremo, solo tenemos $n-1$ aristas. En este caso *no tenemos bucles* en el grafo conectado: el grafo es un *árbol*. El número $\boldsymbol{m}$ de aristas en un grafo conectado está entre $m = n-1$ y $m = \frac{1}{2}n(n-1)$.

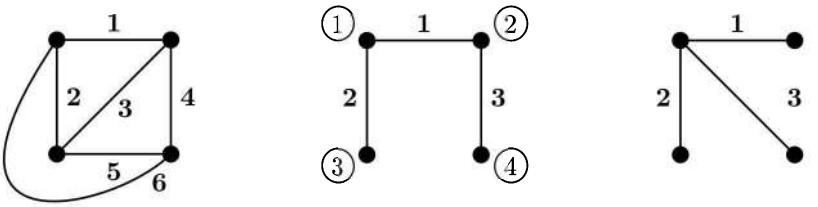


Figura IV.4: un grafo completo y dos árboles con $n = 4$ nodos: $m = 6$ o 3 aristas. La figura central muestra la numeración de los cuatro nodos en los tres grafos.

Esos grafos están conectados con el álgebra lineal mediante sus matrices de incidencia A_1, A_2, A_3 de m por n.

$$\begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \\ -1 & 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}$$

primer árbol **segundo árbol**
matriz de incidencia A_2 **matriz de incidencia A_3**

Nuestra convención es que -1 viene antes de $+1$ en cada fila. Pero estos no son grafos dirigidos. Los flujos en las aristas (las *corrientes* y_1 *a* y_m) pueden ser positivos o negativos. Y toda la información sobre un grafo (sus nodos y aristas) son revelados por su matriz de incidencia.

La matriz de incidencia tiene n columnas cuando el grafo tiene n nodos. La suma de estos vectores columna es el vector nulo. Otra manera de decirlo: el vector de unos $\boldsymbol{x} = (1,1,1,1)$ está en el espacio nulo de las tres matrices de incidencia. El espacio nulo de A es una sola recta con dirección ese vector de unos. $A\boldsymbol{x} = \mathbf{0}$ requiere de $x_1 = x_2 = x_3 = x_4$ de modo que $\boldsymbol{x} = (c,c,c,c)$.

$$\boxed{A\boldsymbol{x} = \mathbf{0}} \qquad \begin{aligned} -x_1 + x_2 &= 0 \quad (\textbf{es decir } \boldsymbol{x_1 = x_2}) \\ -x_1 + x_3 &= 0 \quad (\textbf{es decir } \boldsymbol{x_1 = x_3}) \\ -x_2 + x_3 &= 0 \\ -x_2 + x_4 &= 0 \quad (\textbf{es decir } \boldsymbol{x_2 = x_4}) \\ -x_3 + x_4 &= 0 \\ -x_1 + x_4 &= 0 \end{aligned} \tag{1}$$

Ley de corrientes de Kirchhoff

La ecuación $A\boldsymbol{x} = \mathbf{0}$ no es muy interesante. Todas sus soluciones son vectores constantes. La ecuación $A^{\mathrm{T}}\boldsymbol{y} = \mathbf{0}$ sí es muy interesante : una ecuación central de la matemática aplicada. Necesitamos ver qué significa y encontrar un conjunto completo de $\boldsymbol{m - n + 1}$ **soluciones independientes**.

Ley de corrientes de Kirchhoff
(KCL por su sigla en inglés)
La KCL es $\boldsymbol{A^{\mathrm{T}}y = 0}$

$$\begin{bmatrix} -1 & -1 & 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 1 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{2}$$

Primero contamos las soluciones. Con $m = 6$ incógnitas y y $r = 3$ ecuaciones independientes, hay $6 - 3$ **soluciones independientes**. El espacio nulo de A^{T} tiene dimensión $m - r = \mathbf{3}$. Queremos identificar una base para ese subespacio de $\mathbf{R}^6$.

¿ Cuál es el significado de las cuatro ecuaciones de la ley de corrientes de Kirchhoff $A^{\mathrm{T}}\boldsymbol{y} = \mathbf{0}$? Cada ecuación es una ley de balance de corrientes entrando o saliendo de un nodo :

KCL = Balance de corrientes:
el flujo entrante a cada nodo equivale al flujo saliente de ese nodo.

En el nodo 4, la última ecuación en (2) es $y_4 + y_5 + y_6 = 0$. El flujo total neto entrando al nodo 4 es cero (o los electrones se acumularían). Este balance de corrientes o fuerzas o dinero se ve en todo momento en la ingeniería, la ciencia y la economía. Es la ecuación de balance del equilibrio.

La clave para resolver $\boldsymbol{A^{\mathrm{T}}y = 0}$ es mirar los pequeños bucles en el grafo. Un bucle es un "ciclo" de aristas, un camino que vuelve al principio. El primer grafo en la Figura IV.4 tiene tres pequeños bucles. Recorriendo esos bucles encontramos las siguientes aristas :

bucle 1 : hacia adelante por la arista 2, hacia atrás por las aristas 3 y 1
bucle 2 : hacia adelante por la arista 3 and 5, hacia atrás por la arista 4
bucle 3 : hacia adelante por la arista 6, hacia atrás por las aristas 5 y 2

El flujo alrededor de un bucle automáticamente satisface la ley de corrientes de Kirchhoff. En cada nodo del bucle, el flujo que ingresa en el nodo sale hacia el siguiente. Los tres bucles del grafo producen tres soluciones independientes de $A^{\mathrm{T}}\boldsymbol{y} = \mathbf{0}$. Cada $\boldsymbol{y}$ da seis corrientes por las aristas alrededor de un bucle :

$$A^{\mathrm{T}}\boldsymbol{y} = \mathbf{0} \quad \text{para} \quad \boldsymbol{y}_1 = \begin{bmatrix} -1 \\ 1 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{e} \quad \boldsymbol{y}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \\ 1 \\ 0 \end{bmatrix} \quad \text{e} \quad \boldsymbol{y}_3 = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ -1 \\ 1 \end{bmatrix}. \tag{3}$$

¡ No habría más soluciones independientes incluso si hubiera más bucles (de mayor tamaño) ! El bucle más grande alrededor de todo el grafo es exactamente la suma de los tres bucles pequeños. Así la solución $\boldsymbol{y} = (-1, 0, 0, -1, 0, 1)$ para el bucle exterior es exactamente la suma $\boldsymbol{y}_1 + \boldsymbol{y}_2 + \boldsymbol{y}_3$.

Las dimensiones de los subespacios llevan a una identidad fundamental de la topología (descubierta por Euler):

$$\boxed{\begin{array}{c}(\textbf{Número de nodos}) - (\textbf{Número de aristas}) + (\textbf{Número de bucles}) = \\ (n) - (m) + (m - n + 1) = 1.\end{array}} \tag{4}$$

El lector ya sabe que **un árbol no tiene bucles**. Nuestros segundo y tercer grafo son árboles con 4 nodos y 3 aristas. Entonces la cuenta de Euler es $(4) - (3) + (0) = 1$. Y $A^{\mathrm{T}}\boldsymbol{y} = \mathbf{0}$ solo tiene la solución $\boldsymbol{y} = \mathbf{0}$. En todo árbol las filas de A son independientes.

$$A_2 = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \quad \text{y} \quad A_3 = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}.$$

La matriz $A^{\mathrm{T}}CA$ en matemática aplicada

Los grafos son ejemplos perfectos para las tres ecuaciones que aparecen por todas partes en la ingeniería, la ciencia y la economía. Esas ecuaciones describen un sistema en *equilibrio de estado estacionario*. Para flujos en una red eléctrica (corrientes a lo largo de las seis aristas de nuestro primer grafo) las tres ecuaciones conectan los *voltajes* $\boldsymbol{x} = (x_1, x_2, x_3, x_4)$ *en los cuatro nodos* y las *corrientes directas* $\boldsymbol{y} = (y_1, y_2, y_3, y_4, y_5, y_6)$ *a lo largo de las seis aristas.*

Diferencias de voltaje a lo largo de las aristas	$\boldsymbol{e = Ax}$	e_1 = voltaje en el nodo final 2 − voltaje en el nodo inicial 1
Ley de Ohm en cada arista	$\boldsymbol{y = Ce}$	corriente $y_1 = c_1$ por e_1 = (conductancia) (voltaje)
Ley de Kirchhoff con fuentes de corriente	$\boldsymbol{f = A^{\mathrm{T}}y}$	fuentes de corriente $\boldsymbol{f}$ en nodos equilibran las corrientes internas $\boldsymbol{y}$

Esas tres ecuaciones $\boldsymbol{e = Ax}$, $\boldsymbol{y = Ce}$ y $\boldsymbol{f = A^{\mathrm{T}}y}$ se combinan en una ecuación de equilibrio $\boldsymbol{A^{\mathrm{T}}CAx = f}$. Esta es la forma de tantas leyes fundamentales. La belleza se encuentra en que aparecen tanto A como A^{T}. El resultado es que la matriz gobernante $A^{\mathrm{T}}CA$ es simétrica. $\boldsymbol{A^{\mathrm{T}}CA}$ **es semidefinida positiva** porque el vector con todos unos $\boldsymbol{x} = (1, \ldots, 1)$ es solución de $A\boldsymbol{x} = \mathbf{0}$.

Con una condición de borde como $x_4 = 0$ (que fija el nodo 4 y quita la última columna de A) **la matriz reducida $\boldsymbol{A^{\mathrm{T}}CA}$ se vuelve simétrica definida positiva**.

La red fijada tiene $n - 1 = 3$ voltajes desconocidos (se conoce $x_4 = 0$)

La matriz de incidencia reducida A es ahora de 6 por 3 : rango completo 3.

La matriz del sistema $A^{\mathrm{T}}CA$ es $(3 \times 6)(6 \times 6)(6 \times 3) = \mathbf{3 \times 3}$

La energía es positiva : $\boldsymbol{x}^{\mathrm{T}}A^{\mathrm{T}}CA\boldsymbol{x} = (A\boldsymbol{x})^{\mathrm{T}}C(A\boldsymbol{x}) > 0$ si $\boldsymbol{x} \neq \mathbf{0}$

Ahora $\boldsymbol{A^{\mathrm{T}}CA}$ es simétrica e invertible y definida positiva.

Esta matriz $A^{\mathrm{T}}CA$ es la base de mi curso del MIT 18.085 sobre ciencia e ingeniería computacional. Es el punto donde el álgebra lineal tiene un mensaje importante para los cálculos a gran escala (como el método de elementos finitos). Las clases en vídeo y el libro de texto destacan las aplicaciones de $A^{\mathrm{T}}A$ y $A^{\mathrm{T}}CA$.

Al preservar la estructura simétrica definida positiva de las ecuaciones gobernantes, que a menudo son ecuaciones diferenciales parciales, el formato $A^{\mathrm{T}}CA$ encaja con las leyes de la ciencia. La ley de corrientes de Kirchhoff $A^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{0}$ se vuelve un modelo para todas las leyes de equilibrio: conservación de carga, equilibrio de fuerzas, ingreso neto cero en economía, conservación de masa y energía, continuidad de todo tipo.

La misma matriz $A^{\mathrm{T}}CA$ aparece en regresión lineal (cuadrados mínimos aplicados a $A\boldsymbol{x} = \boldsymbol{b}$).

$\boldsymbol{A^{\mathrm{T}}A\widehat{x} = A^{\mathrm{T}}b}$ Ecuación normal para el vector $\widehat{\boldsymbol{x}}$ que mejor ajusta a los datos $\boldsymbol{b}$

$\boldsymbol{A^{\mathrm{T}}CA\widehat{x} = A^{\mathrm{T}}Cb}$ Cuadrados mínimos ponderados por la matriz de covarianza inversa $C = V^{-1}$

$\boldsymbol{\min ||b - Ax||_C^2}$ Error cuadrático mínimo $(\boldsymbol{b} - A\boldsymbol{x})^{\mathrm{T}}C(\boldsymbol{b} - A\boldsymbol{x})$

El **aprendizaje produndo** en el capítulo final será una vez más un problema de optimización. Encontrar los pesos entre cada capa de neuronas para que la función de aprendizaje F clasifique correctamente los datos de entrenamiento. En el siglo 20, cuando F era lineal, esto no tenía tanto éxito. En este siglo, en cada neurona también se aplica una función de activación no lineal como $\mathbf{ReLU}(\boldsymbol{x})$ (el máximo entre 0 y x). El aprendizaje profundo se ha vuelto ahora increíblemente poderoso.

La función general F que clasifica los datos es *continua y lineal a trozos*. Su gráfico tiene una cantidad increíble de pequeños segmentos de línea recta. Cada aplicación de $\mathbf{ReLU}(\boldsymbol{x})$ suma un pliegue al gráfico de F. Ese pliegue cruza los otros pliegues para dividir el espacio de características en muchas muchas piezas. **Vea la Sección VII.1 sobre redes neuronales profundas.**

Construir todos estos segmentos nos da el poder matemático del aprendizaje profundo.

La matriz laplaciana de grafo

$K = A^{\mathrm{T}}CA$ es una laplaciana de grafo ponderado donde los pesos están en C. La matriz laplaciana estándar es $G = A^{\mathrm{T}}A$, con pesos unitarios ($C = I$). Ambas laplacianas son matrices cruciales para la teoría y las aplicaciones. Las propiedades principales de $A^{\mathrm{T}}A$ y $A^{\mathrm{T}}CA$ se trasladan a todo grafo conexo. K puede ser una matriz de rigidez o una matriz de conductancias en ingeniería.

1 Cada fila y columna de G y K suma cero porque $\boldsymbol{x} = (1, \ldots, 1)$ satisface $A\boldsymbol{x} = \boldsymbol{0}$.

2 $G = A^{\mathrm{T}}A$ es simétrica porque las aristas son bidireccionales (**grafo no dirigido**).

3 El elemento diagonal $(A^{\mathrm{T}}A)_{ii}$ cuenta las aristas que se encuentran en el nodo i : **el grado**.

4 El elemento por fuera de la diagonal es $(A^{\mathrm{T}}A)_{ij} = -1$ cuando una arista conecta los nodos i y j.

5 G y K son **semidefinidas positivas** pero no definidas positivas (porque $A\boldsymbol{x} = \boldsymbol{0}$ en **1**).

$A^{\mathrm{T}}A =$ diagonal $+$ fuera de la diagonal $=$ **matriz de grado** $-$ **matriz de adyacencias** $= \boldsymbol{D - B}$.

Problemas de la Sección IV.6

1 ¿Cuáles son las matrices laplacianas $A^{\mathrm{T}}A$ de un grafo triangular y un grafo cuadrado? La matriz de incidencia A invierte el signo si todas las flechas son invertidas, pero los signos en $A^{\mathrm{T}}A$ no dependen de las flechas.

2 ¿Qué es $A^{\mathrm{T}}A$ para un grafo completo con $n = 5$ nodos? Un grafo completo de $n = 5$ nodos contiene 10 aristas.

3 Para un grafo triangular con pesos c_1, c_2, c_3 en las aristas $1 \to 2$, $1 \to 3$, $2 \to 3$, muestre con una multiplicación matricial que

$$K = A^{\mathrm{T}}CA = \begin{bmatrix} c_1 + c_2 & -c_1 & -c_2 \\ -c_1 & c_1 + c_3 & -c_3 \\ -c_2 & -c_3 & c_2 + c_3 \end{bmatrix}$$

4 Esa matriz $K = A^{\mathrm{T}}CA$ es la suma de $m = 3$ **"matrices elementales"**:

$$K = c_1 \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + c_2 \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} + c_3 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

Muestre que esas matrices de rango 1 provienen de $K = A^{\mathrm{T}}(CA) =$ **columnas por filas**.

5 Dibuje un árbol con $n = 4$ nodos y $m = 3$ aristas. Debería haber $m - n + 1 = 0$ soluciones para la ley de corrientes $A^{\mathrm{T}}\boldsymbol{w} = \mathbf{0}$. Explique la siguiente conclusión: las filas de A que corresponden a un árbol en el grafo son *independientes*.

6 Un grafo completo con $n = 5$ nodos y $m = 10$ aristas aparentemente no puede dibujarse en un plano. ¿Puede demostrar (luego de experimentar) que las aristas se cruzarán?

7 (a) Para ese grafo completo con 4 nodos y 6 aristas, encuentre la matriz $A^{\mathrm{T}}A$.

(b) Encuentre también $6 - 4 + 1$ soluciones (a partir de bucles) para la ley de Kirchhoff $A^{\mathrm{T}}\boldsymbol{w} = \mathbf{0}$.

8 Explique la fórmula de Euler (el inicio de la topología) para un grafo cualquiera en un plano:

$$(\text{número de nodos}) - (\text{número de aristas}) + (\text{número de bucles pequeños}) = \mathbf{1}$$

9 Para un grafo triangular, encuentre los autovalores y autovectores de $G = A^{\mathrm{T}}A$. Los autovectores no están completamente determinados porque G tiene un _____ repetido. Encuentre una SVD: $A = U\Sigma V^{\mathrm{T}}$.

IV.7. Conglomerados usando métodos espectrales y k-medias

¿Cómo se interpreta un grafo con muchos nodos (y sin todas las aristas posibles)? Un punto de partida importante es separar los nodos en dos o más **conglomerados**[1], como grupos de amigos. Las aristas son más probables dentro de los conglomerados que entre los distintos conglomerados. Esperamos que los conglomerados sean de tamaños similares para no estar agrupando algunos elementos solitarios. Identificar estos conglomerados nos dará un primer panorama para comprender el grafo a grandes rasgos.

Un paso clave para decodificar información genética es *agrupar genes* que tienen niveles de expresión muy correlacionados (y a veces anticorrelacionados). Los conglomerados de genes suelen encontrarse en la misma ruta metabólica celular. El gran logro del Proyecto Genoma Humano fue decirnos cuáles son las piezas del rompecabezas de la vida: las filas de G. Ahora enfrentamos el mayor problema de encastrar las piezas juntas para producir una *función*: como la creación de proteínas.

Un ejemplo con dos conglomerados

La figura debajo muestra $n = 5$ nodos. Esos nodos están separados en $k = 2$ conglomerados. Los puntos marcados con $*$ son los centroides $(2, 1)$ y $(-1, 2/3)$ de ambos conglomerados. El primer centroide es el promedio $\frac{1}{2}(1,1) + \frac{1}{2}(3,1)$ de los dos puntos, y el segundo centroide es el promedio $\frac{1}{3}[(0,0)+(-3,0)+(0,2)]$. Esos centroides $\boldsymbol{c}_1$ y $\boldsymbol{c}_2$ minimizan la suma de las distancias al cuadrado $||\boldsymbol{c} - \boldsymbol{a}_j||^2$ a los puntos $\boldsymbol{a}_j$ en los conglomerados.

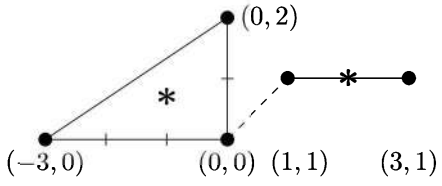
Estos conglomerados fueron producidos por el famoso algoritmo de k-medias con $k = 2$. Esta es una manera sencilla de agrupar los nodos, pero no es la única. (Y probablemente no sea la más rápida ni la mejor para un conjunto grande de nodos). Antes de usar autovalores y cortes para producir conglomerados, mostraremos cómo las k-medias son un ejemplo excelente de un tema central de este libro:

$$\textbf{Aproximar una matriz } \boldsymbol{A} \textbf{ de tamaño } \boldsymbol{m \times n} \textbf{ por } \boldsymbol{CR = (m \times k)(k \times n)} \tag{1}$$

El rango de CR es bajo porque C solo tiene k columnas y R tiene k filas. En la aproximación de k-medias, *las columnas de C son los centroides de los conglomerados.* ¿Qué es R?

Cada columna de R tiene un solo 1 y $k - 1$ ceros. Para ser más precisos, $R_{ij} = 1$ (o 0) si el centroide i es el más cercano (o no) al punto $\boldsymbol{x}_j$. Entonces los 1 en la fila i de R nos dicen el conglomerado de nodos alrededor del centroide (marcado por $*$) en la columna i de C.

Para 5 nodos y 2 conglomerados, R tiene solo dos columnas diferentes (centroides de los conglomerados) en $A \approx CR$.



$$\begin{bmatrix} 0 & 1 & 3 & 0 & -3 \\ 0 & 1 & 1 & 2 & 0 \end{bmatrix} \approx \begin{bmatrix} -1 & 2 & 2 & -1 & -1 \\ 2/3 & 1 & 1 & 2/3 & 2/3 \end{bmatrix}$$

$$\boldsymbol{A \approx CR} = \begin{bmatrix} \boldsymbol{-1} & \boldsymbol{2} \\ \boldsymbol{2/3} & \boldsymbol{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{1} \\ \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$$

[1] En algunos países de habla hispana se los llama también agrupamientos, *clusters* en inglés.

Cuatro métodos para conglomerar

De las muchas aplicaciones, comenzamos por: **dividir un grafo en dos partes**. Esas partes son conglomerados de nodos. La mayoría de las aristas deberían estar dentro de algún conglomerado.

1. Cada conglomerado debería contener aproximadamente la mitad de los nodos.
2. El número de aristas entre los conglomerados debería ser relativamente pequeño.

Para equilibrar la carga en el cálculo de alto rendimiento, asignamos igual trabajo a dos procesadores (con poca comunicación entre ellos). En redes sociales identificamos dos grupos distintos. En una imagen realizamos segmentación. Estamos reordenando filas y columnas de una matriz para que los bloques por fuera de la diagonal sean ralos.

Muchos algoritmos han sido y serán inventados para particionar un grafo. Nos enfocaremos en cuatro métodos exitosos que se extienden a problemas más difíciles : **conglomerado espectral** (con el laplaciano del grafo o la matriz de modularidad), **corte mínimo y k-medias ponderadas**. He aquí esos cuatro métodos :

I. Encontrar el **vector de Fiedler** $\boldsymbol{z}$ que resuelve $A^{\mathrm{T}}CA\boldsymbol{z} = \lambda D\boldsymbol{z}$. La matriz $A^{\mathrm{T}}CA$ es la matriz laplaciana del grafo. Su diagonal D contiene los pesos totales de las aristas de cada uno de los nodos. *D normaliza el laplaciano.*

El autovector asociado a $\lambda_1 = 0$ es $(1, \ldots, 1)$. Le sigue el autovalor de Fiedler : $\lambda = \lambda_2$.

Componentes positivas y negativas de su autovector indican los dos conglomerados de nodos.

II. Reemplazar la matriz laplaciana del grafo $A^{\mathrm{T}}CA$ por la **matriz de modularidad** $\boldsymbol{M}$. Elegir el autovector asociado al mayor autovalor de M. Una vez más serán las componentes positivas y negativas las que indiquen los dos conglomerados :

$$\boxed{\textbf{Matriz de modularidad} \qquad \boldsymbol{M} = \text{ (matriz de adyacencia) } - \frac{1}{2m}\,\boldsymbol{d}\,\boldsymbol{d}^{\mathrm{T}}}$$

El vector $\boldsymbol{d}$ da los grados de los n nodos (el número de aristas adyacentes a los nodos). Cada fila y cada columna de $M = M^{\mathrm{T}}$ suma cero, así uno de los autovectores de M es de nuevo $(1, 1, \ldots, 1)$. Si su autovalor $\lambda = 0$ es el mayor (es decir M no tiene autovalores positivos) entonces todos los nodos irán al mismo conglomerado.

El artículo de Mark Newman en PNAS **103** (2006) 8577-8582 presenta un fuerte argumento a favor de usar la matriz de modularidad para conglomerar los nodos.

III. Encontrar el **corte mínimo normalizado** que separa los nodos en dos conglomerados P y Q. La medida no normalizada de un corte es la **suma de los pesos de las aristas** $\boldsymbol{w_{ij}}$ sobre ese corte. Esas aristas conectan un nodo en P a un nodo por fuera de P:

$$\textbf{peso a lo largo del corte} \quad \boldsymbol{enlaces(P)} = \sum \boldsymbol{w_{ij}} \quad \textbf{para} \quad \boldsymbol{i} \textbf{ en } \boldsymbol{P} \quad \textbf{y} \quad \boldsymbol{j} \textbf{ no en } \boldsymbol{P}. \tag{2}$$

Con esta medida un corte mínimo podría no tener nodos en P. *Entonces normalizamos por los tamaños de P y Q.* Estas son sumas de pesos dentro de conglomerados :

$$\textbf{Tamañano del conglomerado} \qquad \boldsymbol{tamaño(P)} = \sum \boldsymbol{w_{ij}} \quad \textbf{para} \quad \boldsymbol{i} \textbf{ en } \boldsymbol{P}. \tag{3}$$

Observe que una arista dentro de P se cuenta dos veces, como w_{ij} y como w_{ji}. El tamaño sin pesos contaría solo los nodos y nos llevaría a un "corte por proporción." Aquí repartimos el peso a ambos lados del corte por los tamaños de P y Q, para normalizar la cantidad clave *Ncortes*:

$$\textbf{Peso de corte normalizado} \qquad \textbf{Ncortes}(\boldsymbol{P,Q}) = \frac{\textit{enlaces}(P)}{\textit{tamaño}(P)} + \frac{\textit{enlaces}(Q)}{\textit{tamaño}(Q)}. \tag{4}$$

Shi y Malik encontraron que minimizar $Ncortes(P, Q)$ da una buena partición del grafo. Esa aplicación se utilizó en la segmentación de imágenes. Descubrieron la conexión con la laplaciana L.

La definición de *Ncortes* se extiende de dos conglomerados P y Q a k conglomerados $P_1, \ldots, P_k$:

$$N\textbf{-cortes normalizados} \qquad Ncortes(P_1, \ldots, P_k) = \sum_{i=1}^{K} \frac{\textit{enlaces}(P_i)}{\textit{tamaño}(P_i)} \tag{5}$$

Nos estamos acercando al conglomerado de **k-medias**. Comencemos con $k = 2$ conglomerados (P y Q).

IV. k-medias Representar los nodos en el grafo como vectores $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$. Los conglomerados P y Q tienen centros $\boldsymbol{c_P}$ y $\boldsymbol{c_Q}$. Minimizamos la distancia total al cuadrado desde los nodos a esos "centroides".

$$\textbf{conglomerado de 2-medias } \boldsymbol{c_P, c_Q} \textbf{ = centroides} \qquad \textbf{Minimizar } E = \sum_{i \text{ en } P} ||\boldsymbol{a_i} - \boldsymbol{c_P}||^2 + \sum_{i \text{ en } Q} ||\boldsymbol{a_i} - \boldsymbol{c_Q}||^2 \tag{6}$$

El centroide es el *promedio* $\boldsymbol{c_P} = (\sum \boldsymbol{a_i})/|P|$ de los vectores en el conglomerado P.

El vector $\boldsymbol{a_i}$ puede o no representar la ubicación física del nodo i. Entonces el objetivo del conglomerado E no está restringido a la distancia euclideana. El algoritmo más general de **k-medias vía núcleos** funciona con una matriz núcleo (*kernel* en inglés) K que asigna productos internos $K_{ij} = \boldsymbol{a_i^{\mathrm{T}} a_j}$. Las distancias y las medias se calculan a partir de un peso K.

La distancia E también será ponderada, para mejorar los conglomerados P y Q.

La matriz laplaciana del grafo

El primer paso hacia L es $A^{\mathrm{T}}A$. Esta A es la matriz de incidencia de m por n del grafo. Por fuera de la diagonal, si una arista conecta los nodos i y j, el elemento i, j de $A^{\mathrm{T}}A$ es -1. Los elementos diagonales hacen que todas las filas sumen cero. Así $(A^{\mathrm{T}}A)_{ii} =$ número de aristas en el nodo $i =$ grado de nodo i. Con todos los pesos iguales a uno, $A^{\mathrm{T}}A =$ *matriz de grado − matriz de adyacencia.*

Los pesos de las aristas en C pueden ser conductancias, constantes de resortes o longitudes de aristas. Aparecen sobre y por fuera de la diagonal de $\boldsymbol{A^{\mathrm{T}}CA = D - W}$ **= matriz de pesos de nodos − matriz de pesos de aristas**. Por fuera de la diagonal, los elementos de $-W$ son menos

los pesos w_{ij}. Los elementos diagonales d_i todavía hacen que todas las filas sumen cero: $\boldsymbol{D} = \mathbf{diag}(\mathbf{sum}(\boldsymbol{W}))$.

El vector de unos $\mathbf{1} = \mathbf{unos}(n, 1)$ está en el espacio nulo de $A^{\mathrm{T}}CA$, porque $A\mathbf{1} = \mathbf{0}$. Cada fila de A tiene 1 y -1. Equivalentemente, $D\mathbf{1}$ cancela $W\mathbf{1}$ (la suma de cada fila da cero). El siguiente autovector es como el modo de vibración más bajo de un tambor, con $\lambda_2 > 0$.

Para la **laplaciana ponderada normalizada**, multiplicamos $A^{\mathrm{T}}CA$ a izquierda y derecha por $D^{-1/2}$, preservando la simetría. La fila i y la columna j se dividen por $\sqrt{d_i}$ y $\sqrt{d_j}$, de manera que el elemento i, j de $A^{\mathrm{T}}CA$ queda dividido por $\sqrt{d_i d_j}$. Así L tiene $d_i/d_i = \mathbf{1}$ a lo largo de su diagonal principal.

$$\boxed{\begin{array}{l}\textbf{Laplaciana normalizada } \boldsymbol{L} \\ \textbf{Pesos normalizados } \boldsymbol{n_{ij}}\end{array} \quad \boldsymbol{L = D^{-1/2}\, A^{\mathrm{T}}CA\, D^{-1/2} = I - N} \quad \boldsymbol{n_{ij} = \dfrac{w_{ij}}{\sqrt{d_i d_j}}}} \tag{7}$$

Un grafo triangular tiene $n = 3$ nodos y $m = 3$ pesos en las aristas $c_1, c_2, c_3 = w_{12}, w_{13}, w_{23}$:

$$\begin{bmatrix} w_{12}+w_{13} & -w_{12} & -w_{13} \\ -w_{21} & w_{21}+w_{23} & -w_{23} \\ -w_{31} & -w_{32} & w_{31}+w_{32} \end{bmatrix} \qquad \boldsymbol{L} = \begin{bmatrix} \mathbf{1} & -n_{12} & -n_{13} \\ -n_{21} & \mathbf{1} & -n_{23} \\ -n_{31} & -n_{32} & \mathbf{1} \end{bmatrix} \tag{8}$$

$$\boldsymbol{A^{\mathrm{T}}CA = D - W} \qquad\qquad \boldsymbol{L = D^{-1/2} A^{\mathrm{T}}CA\, D^{-1/2}}$$

La laplaciana normalizada $L = I - N$ es como una *matriz de correlación* en Estadística, con diagonal unitaria. Tres de sus propiedades son cruciales para el conglomerado:

1. L es simétrica semidefinida positiva: autovectores ortogonales, autovalores $\lambda \geq 0$.
2. El autovector asociado a $\lambda = 0$ es $\mathbf{u} = (\sqrt{d_1}, \ldots, \sqrt{d_n})$. Así $L\mathbf{u} = D^{-1/2}A^{\mathrm{T}}CA\mathbf{1} = 0$.
3. El segundo autovector $\boldsymbol{v}$ de L minimiza el **cociente de Rayleigh** en un subespacio:

$$\boxed{\begin{array}{l}\boldsymbol{\lambda_2} \textbf{ = menor autovalor no nulo de } \boldsymbol{L} \\ \textbf{Minimizar sujeto a } \boldsymbol{x^{\mathrm{T}}u = 0}\end{array} \quad \textbf{mín}\ \frac{\boldsymbol{x^{\mathrm{T}}Lx}}{\boldsymbol{x^{\mathrm{T}}x}} = \frac{\boldsymbol{v^{\mathrm{T}}Lv}}{\boldsymbol{v^{\mathrm{T}}v}} = \boldsymbol{\lambda_2} \ \text{ en } \ \boldsymbol{x = v}} \tag{9}$$

El cociente $\boldsymbol{x^{\mathrm{T}}Lx/x^{\mathrm{T}}x}$ da una cota superior para λ_2, para todo vector $\boldsymbol{x}$ ortogonal al primer autovector $D^{1/2}\mathbf{1}$. Una buena cota inferior para λ_2 es más difícil de encontrar.

Normalizado versus no normalizado

Los algoritmos de conglomerado podrían usar la matriz sin normalizar $A^{\mathrm{T}}CA$. Pero L generalmente da mejores resultados. La conexión entre ellas es $Lv = D^{-1/2}A^{\mathrm{T}}CAD^{-1/2}v = \lambda v$. Con $z = D^{-1/2}v$ tiene la forma simple e importante $A^{\mathrm{T}}CAz = \lambda Dz$:

$$\boxed{\textbf{Vector de Fiedler normalizado } \boldsymbol{z} \qquad \boldsymbol{A^{\mathrm{T}}CAz = \lambda Dz} \quad \textbf{con} \quad \boldsymbol{1^{\mathrm{T}}Dz = 0.}} \tag{10}$$

Para este problema de autovalores "generalizado" el autovector correspondiente a $\lambda = 0$ sigue siendo el vector de unos $\mathbf{1} = (1, \ldots, 1)$. El siguiente autovector $\mathbf{z}$ es D-ortogonal a $\mathbf{1}$, lo que

significa $\mathbf{1}^{\mathrm{T}} D\,\boldsymbol{z} = 0$ (Sección I.10). Al cambiar $\boldsymbol{x}$ a $D^{1/2}\boldsymbol{y}$, el cociente de Rayleigh encontrará ese segundo autovector $\mathbf{z}$:

$$\boxed{\begin{array}{l}\textbf{Mismo autovalor } \boldsymbol{\lambda_2} \\ \textbf{Fiedler } \boldsymbol{z = D^{-1/2}\,v}\end{array} \qquad \min_{\mathbf{1}^{\mathrm{T}}D\boldsymbol{y}\,=\,0} \frac{\boldsymbol{y}^{\mathrm{T}}A^{\mathrm{T}}CA\boldsymbol{y}}{\boldsymbol{y}^{\mathrm{T}}D\boldsymbol{y}} = \frac{\sum\sum w_{ij}(y_i - y_j)^2}{\sum d_i\, y_i^2} = \lambda_2 \text{ en } \boldsymbol{y} = \boldsymbol{z}.} \tag{11}$$

En $\boldsymbol{Ay}$, la matriz de incidencia A da las diferencias $y_i - y_j$. C las multiplica por w_{ij}.

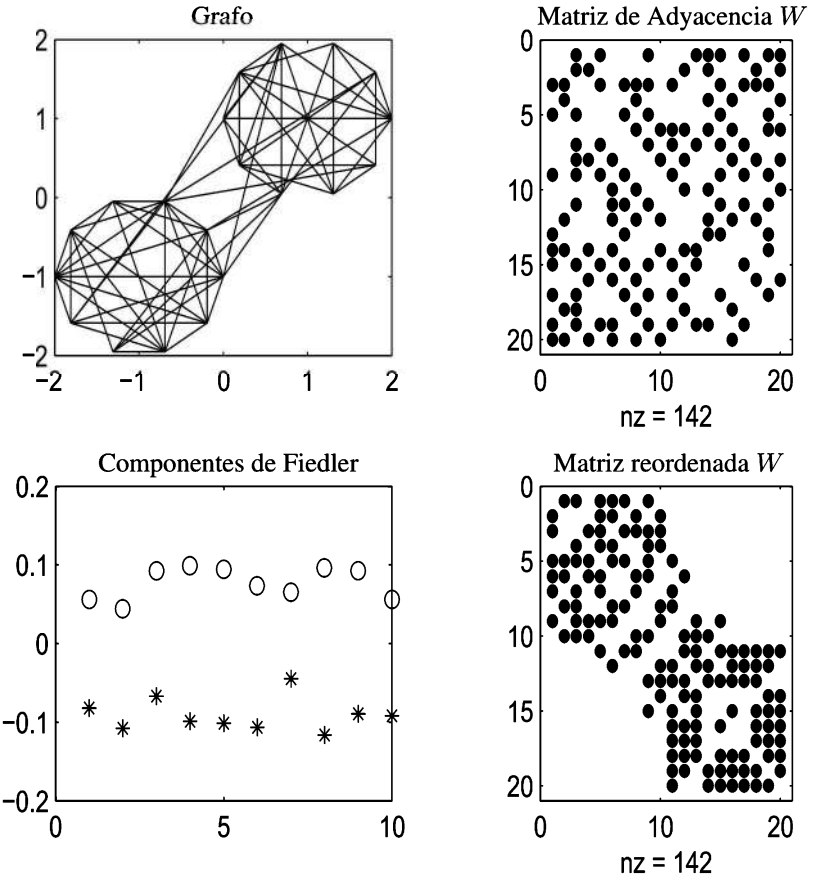
Observación *Para algunos autores, el vector de Fiedler* $\boldsymbol{v}$ es un autovector de $A^{\mathrm{T}}CA$. Preferimos $\boldsymbol{z} = D^{-1/2}\boldsymbol{v}$. Así $A^{\mathrm{T}}CA\boldsymbol{z} = \lambda_2 D\boldsymbol{z}$. Los experimentos parecen dar conglomerados similares a partir de $\boldsymbol{v}$ y de $\boldsymbol{z}$. Los grados ponderados d_i (la suma de los pesos de las aristas en el nodo i) normalizan el problema ordinario de autovalores $A^{\mathrm{T}}CA$ para mejorar el conglomerado.

¿Por qué resolveríamos un problema de autovalores $L\boldsymbol{v} = \lambda\boldsymbol{v}$ (a menudo costoso) como un primer paso para reordenar un sistema lineal $A\boldsymbol{x} = \boldsymbol{b}$? Una respuesta es que no necesitamos un autovector preciso $\boldsymbol{v}$. Un *método multinivel* "jerárquico" combina nodos para obtener una L más pequeña y un $\boldsymbol{v}$ satisfactorio. Los algoritmos de k-medias más rápidos de-refinan (*coarsen* en inglés) el grafo nivel a nivel y luego se ajusta el conglomerado durante la fase de refinamiento.

Ejemplo 1 Un grafo de 20 nodos contiene dos conglomerados P y Q (para encontrar a partir de $\boldsymbol{z}$). El código MATLAB genera aristas dentro de P y dentro de Q, con probabilidad 0.7. Las aristas entre los nodos de P y Q tienen probabilidad más pequeña: 0.1. Todas las aristas tienen pesos $w_{ij} = 1$, es decir $C = I$. P y Q son obvios si observamos el grafo pero no si miramos su matriz de adyacencia W.

Con $G = A^{\mathrm{T}}A$, el comando para calcular autovalores $[V, E] = \mathbf{eig}(G, D)$ resuelve $A^{\mathrm{T}}A\boldsymbol{x} = \lambda D\boldsymbol{x}$. Ordenar los λ determina λ_2 y su vector de Fiedler $\boldsymbol{z}$. El tercer grafo de Des Higham muestra cómo los componentes de $\boldsymbol{z}$ se reparten entre dos conglomerados (más y menos), indicando un buen reordenamiento. Des Higham produjo este código MATLAB.

```
N = 10; W = zeros(2*N, 2*N);        % Genera 2N nodos en dos conglomerados
rand('state', 100)                  % rand se obtiene siempre el mismo grafo
for i = 1:2*N-1
    for j = i+1:2*N
        p = 0.7-0.6 * mod(j-i, 2);  % p = 0.1 cuando j - i es impar, 0.7 en los demás casos
        W(i, j) = rand < p;         % Inserta aristas con probabilidad p
    end                             % Los pesos son w_ij = 1 (o cero)
end                                 % W es una matriz estrictamente triangular superior
W = W + W'; D = diag(sum(W));       % Matriz de adyacencia W, grados en D
G = D-W; [V, E] = eig(G, D);        % Autovalores de Gx = λDx en E
[a, b] = sort(diag(E)); z = V(:, b(2)); % Autovector de Fiedler z para λ2
plot(sort(z), '.-');                % Muestra grupos + − de componentes de Fiedler
```



Aplicación a datos de microarray

Los datos de microarray se organizan en forma de matriz M a partir de m genes y n muestras. Sus elementos m_{ij} registran la actividad (nivel de expresión) del gen i en la muestra j. La matriz $M^{\mathrm{T}}M$ de pesos de n por n mide la similitud entre las muestras (los nodos en un grafo completo).

Los elementos por fuera de la diagonal de $M^{\mathrm{T}}M$ van a W. Las sumas de cada fila de W van en D. Así $D - W$ es la matriz laplaciana ponderada $A^{\mathrm{T}}CA$. Resolvemos $A^{\mathrm{T}}CAz = \lambda Dz$.

Higham, Kalna, y Kibble reportaron resultados para tres conjuntos de datos. Son datos de leucemia ($m = 5000$ genes, $n = 38$ pacientes), tumores cerebrales ($m = 7129$, $n = 40$) y linfoma. "Para revelar información biológica relevante el algoritmo espectral normalizado es por lejos superior a la versión no normalizada".

Los experimentos también muestran cómo el *autovector siguiente a Fiedler* ayuda a producir $k = 3$ conglomerados. Los k autovalores más pequeños dan autovectores que permiten identificar k conglomerados.

Cortes conectados a autovectores

¿Cómo se relaciona el corte del grafo que separa P de Q con el autovector de Fiedler en $A^{\mathrm{T}}CA\boldsymbol{z} = \lambda D\boldsymbol{z}$? El vínculo crucial se obtiene al comparar $Ncortes(P,Q)$ en (5) con el cociente de Rayleigh $\boldsymbol{y}^{\mathrm{T}}A^{\mathrm{T}}CA\boldsymbol{y}/\boldsymbol{y}^{\mathrm{T}}D\boldsymbol{y}$ en (11). El indicador perfecto de un corte sería un vector y con todos las componentes iguales a p o $-q$ (solo dos valores):

Dos valores Nodo i va en P si $y_i = p$ Nodo i va en Q si $y_i = -q$

$\mathbf{1}^{\mathrm{T}}D\boldsymbol{y}$ multiplica un grupo de d_i por p y el otro grupo por $-q$. La suma de los primeros d_i es igual al *tamaño*(P) = suma de w_{ij} (i en P) = suma de d_i (i en P). El segundo grupo de d_i suma a *tamaño*(Q). **La restricción $\mathbf{1}^{\mathrm{T}}D\boldsymbol{y} = 0$ se vuelve $\boldsymbol{p}$ tamaño$(\boldsymbol{P}) = \boldsymbol{q}$ tamaño$(\boldsymbol{Q})$.**

¡Cuando reemplazamos este $\boldsymbol{y}$ en el cociente de Rayleigh, obtenemos exactamente $Ncortes(P,Q)$! Las diferencias $y_i - y_j$ son cero dentro de P y de Q. Son $p+q$ a lo largo del corte:

$$\textbf{Numerador} \qquad \boldsymbol{y}^{\mathrm{T}}A^{\mathrm{T}}CA\boldsymbol{y} = \sum\sum w_{ij}\,(y_i - y_j)^2 = (p+q)^2\, enlaces(P,Q) \tag{12}$$

$$\textbf{Denominador} \qquad \boldsymbol{y}^{\mathrm{T}}D\boldsymbol{y} = p^2\, tamaño(P) + q^2\, tamaño(Q) = p\,(p\, tamaño(P)) + q\,(p\, tamaño(P)). \tag{13}$$

En el último paso se usó que $p\ tamaño(P) = q\ tamaño(Q)$. Cancelamos $p+q$ en el cociente:

$$\text{Cociente de Rayleigh} \quad \frac{(p+q)\ enlaces(P,Q)}{p\ tamaño(P)} = \frac{p\ enlaces(P,Q)}{p\ tamaño(P)} + \frac{q\ enlaces(P,Q)}{q\ tamaño(Q)} = Ncortes(P,Q). \tag{14}$$

El problema de *Ncortes* es el mismo que el problema de autovalores, con la restricción adicional de que las componentes de y tomen uno entre dos valores. (Este problema es NP-difícil: hay muchas opciones para P y Q). El vector de Fiedler $\boldsymbol{z}$ no satisface esta condición de tomar dos valores. Pero sus componentes en ese ejemplo especial están separadas en dos grupos. Conglomerar por $\boldsymbol{z}$ es un éxito si podemos hacerlo eficientemente.

Conglomerados usando k-medias

El problema más básico comienza con n puntos $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ en el espacio d-dimensional. El objetivo es particionar esos puntos en k conglomerados. Los conglomerados $P_1, \ldots, P_k$ tienen centroides $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k$. *Cada centroide $\boldsymbol{c}$ minimiza la distancia total* $\sum ||\boldsymbol{c} - \boldsymbol{a}_j||^2$ a puntos $\boldsymbol{a}_j$ en su conglomerado. El centroide es la media (el promedio) de esos n_j puntos:

Centroide de $\boldsymbol{P_j}$

$$\boldsymbol{c_j} = \frac{\text{suma de } \boldsymbol{a}}{\text{cantidad de } \boldsymbol{a}} \quad \text{minimiza } \sum ||\boldsymbol{c} - \boldsymbol{a}||^2 \text{ para los } a \text{ en el conglomerado } P_j.$$

El objetivo es encontrar la partición $P_1, \ldots, P_k$ con una **distancia total mínima $\boldsymbol{D}$ a los centroides**:

$$\boxed{\textbf{conglomerado} \qquad \text{Minimizar } D = D_1 + \cdots + D_k = \sum ||\boldsymbol{c}_j - \boldsymbol{a}_i||^2 \text{ con } \boldsymbol{a}_i \text{ en } P_j.} \tag{15}$$

Idea clave Los conglomerados $P_1, \ldots, P_k$ de los nodos producen k centroides (paso 1). **Cada conjunto de centroides produce un conglomerado** (paso 2), donde $\boldsymbol{a}$ se va a P_j si $\boldsymbol{c}_j$ es el centroide más cercano a $\boldsymbol{a}$. (En caso de dos centroides equidistantes, elegimos uno arbitrariamente.) El clásico "*algoritmo de k-medias por lotes*" itera desde un conglomerado a sus centroides y de allí a un nuevo conglomerado. La ecuación (1) es una factorización $A = CR$ vía cuadrados mínimos alternantes.

k-medias

1. Hallar los centroides $\boldsymbol{c_j}$ del (viejo) conglomerado $P_1, \ldots, P_k$.

2. Hallar el (nuevo) conglomerado que ubica a $\boldsymbol{a}$ en P_j si $\boldsymbol{c_j}$ es el centroide más cercano.

Cada paso reduce la distancia total D. Redefinimos los centroides $\boldsymbol{c}_j$ para cada P_j, y luego mejoramos a un nuevo P_j alrededor de esos $\boldsymbol{c}_j$. Como D disminuye en ambos pasos, el algoritmo de k-medias converge. Pero podría no converger al mínimo global.

Es difícil saber mucho sobre los conglomerados límites. Particiones no óptimas pueden dar mínimos locales. Las mejores particiones se obtienen usando distancias ponderadas.

El paso 1 es el más costoso, porque calcula todas las distancias $\|\boldsymbol{c}_j - \boldsymbol{a}_i\|^2$. La complejidad es normalmente $O(n^2)$ por iteración. Cuando el algoritmo se extiende a k-medias usando núcleos, generar una matriz nucleo K a partir de los datos puede costar $O(n^2 d)$.

El paso 2 es la "idea de Voronoi" de encontrar el conjunto más cercano a cada centroide.

Pesos y el método del núcleo

Cuando introducimos pesos en las distancias, estos aparecen en los centroides :

$$\textbf{Distancias} \quad d(\boldsymbol{x}, \boldsymbol{a_i}) = w_i \|\boldsymbol{x} - \boldsymbol{a_i}\|^2 \qquad \textbf{Centroide de } \boldsymbol{P_j} \quad \boldsymbol{c_j} = \frac{\sum w_i \boldsymbol{a_i}}{\sum w_i} \ (\boldsymbol{a_i} \text{ en } P_j) \tag{16}$$

La distancia ponderada $D_j = \sum w_i \|\boldsymbol{x} - \boldsymbol{a_i}\|^2$ se minimiza en $\boldsymbol{x} = \boldsymbol{c_j}$ en el paso 1. Para reducir el $D = D_1 + \cdots + D_k$ total, el paso 2 redefine los conglomerados. Cada $\boldsymbol{a_i}$ va con el centroide más cercano. Luego se itera el paso 1 (nuevos centroides) y el paso 2 (nuevos conglomerados).

Un punto clave es que la distancia a los centroides sólo requiere los productos punto $\boldsymbol{a_i} \cdot \boldsymbol{a_j}$:

$$\textbf{Para cada } \boldsymbol{a_i} \textbf{ en } \boldsymbol{P_j} \qquad \|\boldsymbol{c_j} - \boldsymbol{a_i}\|^2 = \boldsymbol{c_j} \cdot \boldsymbol{c_j} - 2\boldsymbol{c_j} \cdot \boldsymbol{a_i} + \boldsymbol{a_i} \cdot \boldsymbol{a_i} \tag{17}$$

Método del núcleo La matriz ponderada del núcleo K tiene elementos $\boldsymbol{a_i} \cdot \boldsymbol{a_\ell}$. Esos vectores $\boldsymbol{a}_i$ no necesitan ser posiciones reales en el espacio. Cada aplicación puede mapear los nodos del grafo a los vectores $\boldsymbol{a}_i$ de manera lineal o *no lineal* con su propia regla. Cuando los nodos son puntos $\boldsymbol{x_i}$ en un espacio de entrada, sus vectores representantes $\boldsymbol{a_i} = \phi(\boldsymbol{x_i})$ pueden ser puntos de un **espacio de predictores** de dimensión alta. Comúnmente se usan tres núcleos :

En visión	**Polinomial**	$K_{i\ell} = (\boldsymbol{x_i} \cdot \boldsymbol{x_\ell} + c)^d$
En estadísitica	**Gaussiano**	$K_{i\ell} = \exp(-\|\boldsymbol{x_i} - \boldsymbol{x_\ell}\|^2 / 2\sigma^2)$
En redes neuronales	**Sigmoide**	$K_{i\ell} = \tanh(c\, \boldsymbol{x_i} \cdot \boldsymbol{x_\ell} + \theta)$

Debido a la fórmula (16) para los centroides, la distancia en (17) solo necesita la matriz del núcleo.

$$\textbf{Suma sobre los nodos en } P_j \quad \sum \|c_j - a_i\|^2 = \frac{\sum\sum w_i w_\ell K_{i\ell}}{\left(\sum w_i\right)^2} - 2\,\frac{\sum w_i K_{i\ell}}{\sum w_i} + \sum K_{ii}. \quad (18)$$

El **algoritmo de núcleo de k-medias por lotes** usa la matriz K para calcular esta distancia total.

Para grandes conjuntos de datos, las k-medias y $\mathbf{eig}(A^{\mathrm{T}}CA, D)$ serán costosas. He aquí dos enfoques que crean una secuencia de problemas más manejables. Dada una muestra de los nodos, el **muestreo aleatorio** encuentra la mejor partición. Usa sus centroides para particionar todos los nodos, asignándoles el centroide más cercano. El muestreo se ha convertido en un importante tema de investigación, intentando demostrar que con probabilidades altas la partición es buena.

El código **graclus** de Dhillon usa **conglomerados multinivel**: de-refina el grafo, luego forma conglomerados a nivel de la base y luego refina. El de-refinamiento forma supernodos con la suma de los pesos de las aristas. Para el supergrafo pequeño a nivel de la base, el conglomerado espectral o el recursivo de 2-medias serán rápidos. **Este enfoque multinivel es como un método multigrilla algebraico**.

Aplicaciones de conglomerados

El motivo de esta sección es la amplia gama de usos. He aquí una colección que se extiende más allá de los conglomerados. Esta parte de la matemática aplicada ha crecido muy rápidamente.

1. Teoría de aprendizaje, conjuntos de entrenamiento, redes neuronales, modelos de Markov ocultos.
2. Clasificación, regresión, reconocimiento de patrones, máquinas de soporte vectorial.
3. Aprendizaje estadístico, máxima verosimilitud, estadística bayesiana, *kriging*, series temporales, modelos ARMA, procesos estacionarios, predicción.
4. Redes sociales, redes mundiales pequeñas, seis grados de separación, teoría de la organización, distribuciones de probabilidad con colas pesadas.
5. Minería de datos, indexación de documentos, indexación semántica, matriz palabra-documento, recuperación de imágenes, aprendizaje basado en núcleos, método de Nystrom, aproximación de bajo rango.
6. Bioinformática, datos de microarray, biología de sistemas, detección de homología de proteínas.
7. Quiminformática, diseño de drogas, unión de ligandos, similitud de pares, árboles de decisión.
8. Teoría de la información, cuantificación de vectores, teoría de la distorsión de la tasa, divergencias de Bregman.
9. Segmentación de imágenes, visión por ordenador, textura, minicorte, cortes normalizados.
10. Control predictivo, muestras de retorno, robótica, control adaptativo, ecuaciones de Riccati.

Problemas de la Sección IV.7

1 Si el grafo es una recta con 4 nodos, y todos los pesos son 1 ($C = I$), el mejor corte es por el medio. Encuentre este corte a partir de las componentes $\pm$ del vector de Fiedler $\boldsymbol{z}$:

$$A^{\mathrm{T}}CA\,\boldsymbol{z} = \begin{bmatrix} 1 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ & & -1 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \lambda_2 \begin{bmatrix} 1 & & & \\ & 2 & & \\ & & 2 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \lambda_2 D\boldsymbol{z}.$$

Aquí $\boldsymbol{\lambda_2 = \frac{1}{2}}$. Encuentre $\boldsymbol{z}$ resolviendo a mano y verifique que $[1 \;\; 1 \;\; 1 \;\; 1]\, D\boldsymbol{z} = 0$.

2 Para el mismo árbol de 4 nodos, calcule *enlaces*(P) y *tamaño*(P) y *Ncortes*(P, Q) para el corte por el medio.

3 Comenzando a partir de los mismos cuatro puntos $1, 2, 3, 4$ encuentre los centroides $\boldsymbol{c_P}$ y $\boldsymbol{c_Q}$ y la distancia total D para los conglomerados $P = \{1, 2\}$ y $Q = \{3, 4\}$. Cuando asigne los cuatro puntos a los centroides más cercanos, el algoritmo de k-medias no cambiará P y Q.

4 Comience el algoritmo de k-medias con $P = \{1, 2, 4\}$ y $Q = \{3\}$. Encuentre los dos centroides y reasigne puntos al centroide más cercano.

5 Para los conglomerados $P = \{1, 2, 3\}$ y $Q = \{4\}$, los centroides son $c_P = 2$ y $c_Q = 4$. Resolver un empate de la manera incorrecta dejará esta partición sin mejoras. Encuentre la distancia total D.

6 Si el grafo consiste de 8 nodos en una grilla de 2 por 4, con pesos $C = I$, use $\mathsf{eig}(A^{\mathrm{T}}A, D)$ para encontrar el vector de Fiedler $\boldsymbol{z}$. La matriz de incidencia A es de 10 por 8 y $D = \mathsf{diag}(\mathsf{diag}(A^{\mathrm{T}}A))$. ¿Qué conglomerados se obtienen de las componentes $\pm$ de $\boldsymbol{z}$?

7 Use el código de Fiedler con las probabilidades $p = 0.5$ y 0.6 en lugar de $p = 0.1$ y 0.7. Calcule z y grafique tanto el grafo como su partición.

Los problemas 8-11 tratan sobre el grafo con nodos $\boldsymbol{(0,0), (1,0), (3,0), (0,4), (0,8)}$.

8 ¿Cuáles conglomerados P y Q maximizan la distancia mínima D^* entre ellos?

9 Encuentre esos mejores conglomerados con el *algoritmo voraz* (*greedy* en inglés). Comience con cinco conglomerados y combine los dos conglomerados más cercanos. ¿Cuáles son los mejores k conglomerados con $k = 4, 3, 2$?

10 El **árbol generador mínimo** es el grupo de aristas más corto que conecta todos los nodos. Habrá $n - 1$ aristas y ningún bucle, o la longitud total no será mínima.

Algoritmo de Prim Comience con cualquier nodo, por ejemplo con $(0, 0)$. En cada paso, incluya la arista más corta que conecta un nodo nuevo al árbol parcial ya creado.

11 El árbol generador mínimo también puede encontrarse mediante la *inclusión voraz*. Con las aristas en orden creciente de longitud, *mantenga cada arista a menos que complete un bucle*.

IV.8. Completando matrices de rango uno

Completar los elementos faltantes en una matriz de rango 1 está estrechamente vinculado a encontrar (o no encontrar) ciclos en un grafo. Esta teoría de completación de rango 1 se desarrolló a partir de la siguiente pregunta:

Nos dan $m + n - 1$ elementos no nulos en una matriz A de m por n.

¿ *Cuándo el requerimiento de rango* $(A) = 1$ *determina todos los demás elementos* ?

La respuesta depende de las *posiciones* de esos $m+n-1$ elementos no nulos. He aquí tres ejemplos:

$$A_1 = \begin{bmatrix} \times & \times & \times \\ \times & & \\ \times & & \end{bmatrix} \qquad A_2 = \begin{bmatrix} \times & \times & \\ \times & \times & \\ & & \times \end{bmatrix} \qquad A_3 = \begin{bmatrix} \times & \times & & \\ & \times & \times & \\ \times & & \times & \\ & & & \times \end{bmatrix}$$

éxito **falla** **falla**

En A_1, nos dan la columna 1 con elementos no nulos. Si A tiene rango 1, las columnas 2 y 3 deben ser múltiplos de la columna 1. Como nos dan los primeros elementos en las columnas 2 y 3, esas columnas están completamente determinadas.

He aquí otro enfoque para A_1. En cualquier matriz de rango 1, *cada determinante de* 2 *por* 2 *debe ser cero*. Entonces, el elemento 2, 2 de A_1 se decide por la ecuación $a_{22}a_{11} = a_{12}a_{21}$.

En A_2, los primeros cuatro elementos pueden no satisfacer determinante $= 0$. Entonces estamos destinados a fallar. Si ese determinante *es* cero, podríamos elegir cualquier $a_{31} \neq 0$ en la columna 1, y completar A_2 para que tenga rango 1. Esta es la situación usual: ninguna solución o infinitas soluciones.

Ese ejemplo falla siempre que sepamos los cuatro elementos de una submatriz de 2 por 2. ¿ Todas las fallas ocurren de esta manera ? No, A_3 *tiene una falla distinta.*

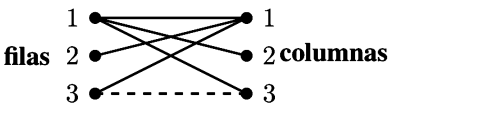
En A_3, la matriz de 3 por 3 superior izquierda tiene demasiados elementos especificados. Hay 6 en lugar de $3 + 3 - 1 = 5$. Considerando matrices de 3 por 3, para la mayoría de las elecciones de esos 6 elementos el rango 1 es imposible:

$$\begin{bmatrix} 1 & 1 & \\ & 1 & 1 \\ 1 & & 2 \end{bmatrix} \quad \text{lleva a} \quad \begin{bmatrix} 1 & 1 & \mathbf{1} \\ \mathbf{1} & 1 & 1 \\ 1 & \mathbf{2} & 2 \end{bmatrix} \quad \text{y el rango es 2.}$$

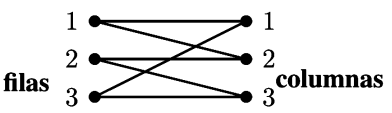
Árboles generadores en un grafo

Alex Postnikov explicó la manera correcta de atacar este problema de completación de rango 1. Construyó un grafo con m nodos para las m filas y n nodos para las n columnas. Para cada elemento prescrito A_{ij}, el grafo tiene una arista que conecta el nodo fila i con el nodo columna j. Así el patrón de $\times$ en las matrices mencionadas se vuelve un patrón de aristas en sus grafos fila-columna (siguiente página).

Los ejemplos A_1 y A_3 producen estos dos grafos :



Elementos no nulos prescritos en A_1 y A_3



Estos son **grafos bipartitos** porque todas sus aristas van de una parte a la otra. El éxito para A_1 y la falla para A_3 pueden explicarse con estos grafos.

Grafo A_1 Las 5 aristas forman un **árbol generador**. Es un *árbol* porque el grafo no tiene un bucle cerrado (**ningún ciclo**). El árbol es *generador* porque conecta los 6 nodos. Si queremos descubrir el elemento $3,3$ de A_1 (en una completación de rango 1), agregamos la arista indicada por una línea de puntos para cerrar o completar un ciclo. Así A_{33} está determinado por A_{11}, A_{13}, y A_{31} en el ciclo. Estos cuatro números producen un determinante cero.

Grafo A_3 Las 7 aristas *no* forman un árbol generador. No es un árbol porque hay un ciclo (en las seis aristas superiores). El ciclo impone un requisito a esos seis elementos de A_3 :

$$\text{Si } A_3 = \boldsymbol{uv}^{\mathrm{T}} \text{ tiene rango 1, se debe satisfacer que } \frac{A_{11}A_{22}A_{33}}{A_{12}A_{23}A_{31}} = \frac{(u_1v_1)(u_2v_2)(u_3v_3)}{(u_1v_2)(u_2v_3)(u_3v_1)} = \mathbf{1}.$$

Si esta condición se cumple, hay infinitas maneras de completar A_3 con rango 1.

Conclusión *La matriz con datos faltantes A tiene una única completación de rango 1 si y solo si los elementos prescritos A_{ij} producen $m+n-1$ aristas (fila i a columna j) que forman un **árbol generador** en el grafo fila-columna.* **El árbol conecta todos los nodos sin bucles.**

Problema abierto ¿ Cuáles $(m+n-2)2$ elementos de A pueden especificarse para permitir una completación única a una matriz de rango 2 ? El Apéndice C de este libro muestra que $(m+n-2)2$ es el número correcto de parámetros independientes para una matriz de rango 2.

Problemas de la Sección IV.8

1 Grafica el grafo bipartito con 3 nodos fila y 3 nodos columna para la matriz del ejemplo A_2. ¿ Conforman las 5 aristas de A_2 un árbol generador ?

2 Construya una matriz A_4 de 5 por 5 con $5+5-1=9$ elementos no nulos en un ciclo de longitud 8. ¿ Qué ecuación del tipo $A_{11}A_{22}A_{33} = A_{12}A_{23}A_{31}$ se debe cumplir para que sea posible la completación a una matriz de rango 1 ?

3 Para un grafo conexo con M aristas y N nodos, ¿ qué requisito sobre M y N se corresponde con cada una de las palabras *árbol generador* ?

4 ¿ Cómo sabe que un grafo con N nodos y $N-1$ aristas es un árbol generador ?

IV.9. El problema ortogonal de Procrustes

He aquí una aplicación linda (y útil) de la SVD. A partir de los vectores $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ e $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$. **¿Por qué matriz ortogonal Q podemos multiplicar los y para que estén lo más cerca posible a los x?** Esta pregunta aparece en una cantidad sorprendente de aplicaciones.

Observe la limitación de utilizar una matriz ortogonal Q. Estas no permiten traslados ni re-escalados. Si la media de las $\boldsymbol{y}$ es igual a la media de las $\boldsymbol{x}$, entonces no se necesita una traslación. Si eso no pasa, la mayoría de los códigos restarán esas medias para lograr nuevos $\boldsymbol{x}$ y nuevos $\boldsymbol{y}$ con $\sum \boldsymbol{x}_i = \sum \boldsymbol{y}_i =$ **vector cero**. Así los dos conjuntos están centrados con media cero. Y si permitimos re-escalar por adelantado los vectores para que sus longitudes sean iguales, estos nos lleva al "problema generalizado de Procrustes".

> *Debo contarles sobre Procrustes y el mito.* Procrustes era famoso por re-escalar. Invitaba extraños a pasar una noche cómoda en su cama especial. Afirmaba que esa cama ajustaba su longitud para que coincida con la altura del visitante, pero en realidad Procrustes ajustaba la altura del visitante para que entre en la cama. (Los visitantes bajos eran estirados en un potro y a los visitantes altos se les cortaban las piernas. No estoy seguro qué nos dice este mito sobre los griegos, pero no es algo bueno.) Teseo le hizo frente al desafío y ajustó a Procrustes a su propia cama. Los resultados fueron, desafortunadamente, fatales.

Ellos resolvieron el problema generalizado y nosotros resolvemos el problema estándar: Q ortogonal.

Solución
1. Construir matrices X e Y con columnas $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ e $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$
2. Armar la matriz cuadrada $Y^{\mathrm{T}}X$
3. Encontrar la descomposición en valores singulares de $Y^{\mathrm{T}}X = U\Sigma V^{\mathrm{T}}$
4. **La matriz ortogonal $\boldsymbol{Q = V^{\mathrm{T}}U}$ minimiza $\boldsymbol{||X - YQ||_F^2}$**

Discusión

La distancia entre las columnas $\boldsymbol{x}_k$ e $\boldsymbol{y}_k Q$ es la distancia euclídea usual $||\boldsymbol{x}_k - \boldsymbol{y}_k Q||$. Para calcular la distancia (al cuadrado) de todos los $\boldsymbol{x}$ a todos los $\boldsymbol{y}Q$, es natural sumar los cuadrados de las normas de esas columna. Esto produce la **norma de Frobenius** (al cuadrado) $||X - YQ||_F^2$. Debemos demostrar que **$\boldsymbol{Q = V^{\mathrm{T}}U}$ minimiza esta norma en el Paso 4**: es la mejor Q posible.

Tres pequeñas observaciones serán útiles para la prueba:

(i) La norma de Frobenius al cuadrado $||A||_F^2$ es la traza de $A^{\mathrm{T}}A$.

(ii) La traza de $A^{\mathrm{T}}B$ es igual a la traza de $B^{\mathrm{T}}A$ y también a la traza de BA^{T}.

(iii) La norma al cuadrado $||A||_F^2$ es igual a $||AQ||_F^2$ y $||A^{\mathrm{T}}||_F^2$.

La traza de una matriz cuadrada es la suma de los elementos de la diagonal principal. También es la suma de sus autovalores. Así las siguientes tres observaciones se comprenden fácilmente.

(i) Los elementos diagonales de $A^{\mathrm{T}}A$ son las longitudes al cuadrado de las columnas, entonces suman $||A||_F^2$.

(ii) $A^{\mathrm{T}}B$ y su traspuesta $B^{\mathrm{T}}A$ tienen la misma diagonal, y por lo tanto, la misma traza.

$A^{\mathrm{T}}B$ y BA^{T} tienen los mismos autovalores no nulos. Sus diagonales incluyen todos los $a_{ij}b_{ij}$.

(iii) Las normas de las columnas de AQ son las mismas que las de A, porque Q es una matriz ortogonal.

Ambas $||A||_F^2$ y $||A^{\mathrm{T}}||_F^2$ suman los cuadrados de todos los elementos de A, así que son iguales.

Prueba de que $Q = V^{\mathrm{T}}U$ es la matriz ortogonal que minimiza $||X - YQ||_F^2$.

Minimizamos traza $(X - YQ)^{\mathrm{T}}(X - YQ) =$ traza $(X^{\mathrm{T}}X) +$ traza $(Y^{\mathrm{T}}Y) -$ **2 traza** $(Q^{\mathrm{T}}Y^{\mathrm{T}}X)$.

$X^{\mathrm{T}}X$ e $Y^{\mathrm{T}}Y$ son fijas. Entonces, *maximizamos* esa última traza. Este es el momento de usar la SVD: $Y^{\mathrm{T}}X = U\Sigma V^{\mathrm{T}}$.

$$\textbf{traza}\,(Q^{\mathrm{T}}Y^{\mathrm{T}}X) = \text{traza}\,(Q^{\mathrm{T}}U\Sigma V^{\mathrm{T}}) = \text{traza}\,(V^{\mathrm{T}}Q^{\mathrm{T}}U\Sigma) = \textbf{traza}\,(Z\Sigma). \tag{1}$$

La matriz $Z = V^{\mathrm{T}}Q^{\mathrm{T}}U$ es un producto de matrices ortogonales y por lo tanto también es ortogonal. Σ es una matriz diagonal de números positivos $\sigma_1, \ldots, \sigma_r$. Así la traza de esa matriz $Z\Sigma$ es $z_{11}\sigma_1 + \cdots + z_{rr}\sigma_r$. Para maximizar ese número **la mejor opción es $Z = I$.**

$Z = V^{\mathrm{T}}Q^{\mathrm{T}}U = I$ significa que $Q = UV^{\mathrm{T}}$ resuelve el problema de Procrustes

En caso de que las matrices originales X e Y sean ortogonales, la rotación perfecta para producir $X = YQ$ (con error cero $X - YQ$) es claramente $Q = Y^{\mathrm{T}}X$. Esto concuerda con la respuesta $Q = UV^{\mathrm{T}}$. Los valores singulares de una matriz ortogonal son todos 1, así la descomposición $Y^{\mathrm{T}}X = U\Sigma V^{\mathrm{T}}$ en la ecuación (1) es exactamente UV^{T}.

Notas El problema de Procrustes fue originalmente resuelto en la tesis de Schöneman en 1964. *Procrustes Problems*, de Gower y Dijksterhuis, (Oxford University Press, 2004) desarrolla muchas de sus aplicaciones. La prueba presentada fue tomada prestada de de: Golub and Van Loan (*Matrix Computations* 4th edition, página 328). Esa página también explica las aplicaciones importantes de la SVD para **ángulos entre subespacios**.

Problemas de la Sección IV.9

1 ¿Cuál matriz ortogonal Q minimiza $||X - YQ||_F^2$? Use la solución $Q = UV^{\mathrm{T}}$ anterior y también minimice esa norma como función de θ (iguale a cero la derivada con respecto a θ):

$$X = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \qquad Y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad Q = \begin{bmatrix} \cos\theta & -\operatorname{sen}\theta \\ \operatorname{sen}\theta & \cos\theta \end{bmatrix}$$

IV.10. Matrices de distancias

Supongamos que n puntos se encuentran en las posiciones $\boldsymbol{x}_1$ a $\boldsymbol{x}_n$ en el espacio d-dimensional. La matriz de distancia D de n por n es simétrica ya que contiene las distancias al cuadrado $D_{ij} = ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2$ entre pares de puntos. La diagonal principal de D tiene ceros ya que $||\boldsymbol{x}_i - \boldsymbol{x}_i||^2 = 0$. He aquí la pregunta clave para extraer información de D :

¿ Podemos recuperar las posiciones $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ a partir de la matriz de distancia euclídea D ?

La respuesta inmediata es *no*. Supongamos que encontramos una solución, un conjunto de posiciones posibles $\boldsymbol{x}_i$. Entonces podríamos desplazar esas posiciones sumando una constante $\boldsymbol{x}_0$. Podríamos multiplicar todos los $\boldsymbol{x}$ por una matriz ortogonal cualquiera. Esos son **movimientos rígidos** y no cambian las distancias $||\boldsymbol{x}_i - \boldsymbol{x}_j||$. Aun reconociendo que una solución implica esta familia de soluciones equivalentes, sigue vigente la pregunta (y la responderemos) :

¿ Hay siempre posiciones $\boldsymbol{x}_1$ a $\boldsymbol{x}_n$ consistentes con la matriz de distancia D ? *Sí.*

Si las distancias de D satisfacen la desigualdad triangular, siempre hay una matriz de posición X (con columnas $\boldsymbol{x}_1$ a $\boldsymbol{x}_n$) que produce las distancias dadas. La matriz X tiene d filas cuando los puntos están en un espacio d-dimensional ($d = 2$ para un mapa, $d = 3$ para nuestro planeta, $d > 3$ se permite y ocurre). Un problema es determinar la dimensión mínima d.

Este problema de encontrar X a partir de D tiene una larga historia. Al principio era una pregunta puramente matemática. Pero pronto aparecieron las aplicaciones. Mencionaremos solo tres de ellas :

1. **Redes inalámbricas de sensores** : medidas de los tiempos de viaje entre pares de sensores da una D. Luego encontramos las posiciones de los sensores X (la topología de la red).

2. **Formas de moléculas** : la resonancia magnética nuclear da la distancia entre átomos. Así conocemos la matriz D. Encontramos las matrices de posición X. Este ejemplo y muchos más involucran **ruido** (errores en D) e incluso **elementos faltantes**.

3. **Aprendizaje automático** : las muestras de un conjunto de entrenamiento se convierten a *vectores de predictores* en dimensiones altas. Esos vectores pueden encontrarse cerca de un plano o una superficie curva de una dimensión mucho menor. Encontrar esa superficie (aproximadamente) es un paso gigante hacia entender los datos y hacer clasificación en nuevas muestras. Aquí el *truco del núcleo* reduce la dimensión. Esto también puede involucrar un entorno muy ruidoso.

Mientras preparaba esta sección del libro, me basé en un artículo maravilloso "*Euclidean Distance Matrices*" publicado en 2015 en el arXiv : 1502.07541v2 [cs.OH]. Sus autores son Ivan Dokmanic, Reza Parhizkar, Juri Ranieri, y Martin Vetterli. Este artículo está lleno de ideas, algoritmos, códigos y muchas aplicaciones. Con repecto a la teoría, le agradecemos a Abdo Alfakih por su ayuda y su excelente libro :

Euclidean Distance Matrices and Their Applications in Rigidity Theory, Springer (2018).

Posiciones X a partir de distancias D

He aquí la observación clave que simplifica el problema que conecta cada (distancia)^2 en D a cuatro elementos en la matriz $X^{\mathrm{T}}X$ (productos punto de los vectores deseados $\boldsymbol{x}_i$ y $\boldsymbol{x}_j$):

$$||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 = (\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{x}_j) = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_i - \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j - \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{x}_i + \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{x}_j \tag{1}$$

El primer término $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_i$ produce una matriz con filas constantes (no depende de j). El último término $\boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{x}_j$ produce una matriz con columnas constantes (no depende de i). Los números $||\boldsymbol{x}_i||^2$ y $||\boldsymbol{x}_j||^2$ en ambas matrices están sobre la diagonal principal de $\boldsymbol{G} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$. Esos son los números en el vector columna $\mathrm{diag}(\boldsymbol{G})$.

Los términos del medio $-2\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j$ en (1) son exactamente los números en $-2G = -2X^{\mathrm{T}}X$. Entonces podemos reescribir (1) como una ecuación que conecta D con G, usando el símbolo $\mathbf{1}$ para el vector columna de n unos.

$$\boxed{\boldsymbol{D} = \mathbf{1}\,\mathrm{diag}(\boldsymbol{G})^{\mathrm{T}} - 2\boldsymbol{G} + \mathrm{diag}(\boldsymbol{G})\,\mathbf{1}^{\mathrm{T}}.} \tag{2}$$

Nuestro problema es recuperar G a partir de D. Si G es semidefinida positiva, las posiciones en X serán obtenidas de $X^{\mathrm{T}}X = G$. Veremos que una solución de X puede ser multiplicada por cualquier matriz ortogonal Q, y todavía $(QX)^{\mathrm{T}}(QX) = G$. Q permite las rotaciones.

Resolver $X^{\mathrm{T}}X = G$ para la matriz X de d por n nos dirá que los puntos $\boldsymbol{x}_1$ a $\boldsymbol{x}_n$ pueden ubicarse en el espacio $\mathbf{R}^d$. El rango de G será la *dimensión espacial*, esta es la dimensión más pequeña consistente con la matriz de distancia D dada.

Ya que dos términos en la ecuación (2) son matrices de rango 1, concluimos que D tiene rango $d+1$ o $d+2$. Observemos que todos nuestros puntos pueden ser desplazados por un vector constante sin cambiar las distancias cuadradas en D. Así el rango de D es una dimensión de un espacio desplazado o *afín* en lugar de una dimensión *espacial*.

Ahora resolvemos la ecuación (2) para G. Ubiquemos el primer punto en el origen: $\boldsymbol{x}_1 = \mathbf{0}$. Luego cada $||\boldsymbol{x}_i - \boldsymbol{x}_1||^2$ es simplemente $||\boldsymbol{x}_i||^2$. La primera columna $\boldsymbol{d}_1$ de D (que está dada) debe coincidir exactamente con $\mathrm{diag}(X^{\mathrm{T}}X) = \mathrm{diag}(G) = (||\boldsymbol{x}_1||^2, ||\boldsymbol{x}_2||^2, \ldots, ||\boldsymbol{x}_n||^2)$.

$$\boxed{\mathrm{diag}(\boldsymbol{G}) = \boldsymbol{d}_1 \quad \text{y} \quad \boldsymbol{G} = -\tfrac{1}{2}(\boldsymbol{D} - \mathbf{1}\boldsymbol{d}_1^{\mathrm{T}} - \boldsymbol{d}_1\mathbf{1}^{\mathrm{T}}).} \tag{3}$$

Una vez que conocemos G encontramos X a partir de $X^{\mathrm{T}}X = G$. ¡Pero G debe ser semidefinida positiva! Esto nos indica la prueba que D debe pasar para ser una matriz de distancia aceptable.

Teorema Una matriz simétrica D con diagonal cero es una matriz de distancia euclídea si y solo si $\boldsymbol{x}^{\mathrm{T}}D\boldsymbol{x} \leq 0$ para todo vector con $\Sigma x_i = 0$ (o $\mathbf{1}^{\mathrm{T}}\boldsymbol{x} = 0$).

Ahora podemos resolver $X^{\mathrm{T}}X = G$ por eliminación en G o encontrando sus autovalores y sus autovectores:

Si $G = Q\Lambda Q^{\mathrm{T}}$ (autovalores y autovectores) entonces X puede ser $\sqrt{\Lambda}Q^{\mathrm{T}}$

Si $G = U^{\mathrm{T}}U$ (eliminación = factorización de Cholesky), entonces X puede ser U (triangular superior)

En ambos casos podemos mantener a lo sumo rango(G) filas en X. Las otras filas son todas cero, vienen de autovalores nulos en Λ o por terminación temprana de la eliminación.

Si quitamos las filas nulas de X veremos la dimensión d del conjunto de puntos $\boldsymbol{x}$. Si las distancias al cuadrado en D incluyen ruidos de medición, imponemos que los autovalores pequeños sean cero.

Este es el algoritmo MDS clásico : escalado multidimensional (*MultiDimensional Scaling*) con el primer punto $\boldsymbol{x}_1 = \mathbf{0}$.

Centrar X o rotar para coincidir con puntos anclados

Centrar : a menudo preferimos ubicar el **centroide** de los $\boldsymbol{x}$ en el origen. El centroide de $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ es tan solo el promedio de esos vectores :

$$\textbf{Centroide} \qquad \boldsymbol{c} = \frac{1}{n}(\boldsymbol{x}_1 + \cdots + \boldsymbol{x}_n) = \frac{1}{n}\boldsymbol{X}\,\mathbf{1}. \tag{4}$$

Basta multiplicar cualquier matriz de posición X por la matriz $I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}$ para colocar el centroide en $\mathbf{0}$.

Puntos anclados : posiblemente algunas de las posiciones ya hayan sido seleccionadas por adelantado : $\boldsymbol{N}$ **puntos anclados** $\boldsymbol{y_i}$ **en una matriz** $\boldsymbol{Y}$. Esas posiciones pueden no coincidir con los $\boldsymbol{x}_i$ calculados. Entonces elegimos las N columnas correspondientes a partir de la matriz de posición X calculada, y luego encontramos la rotación Q que lleva esas columnas lo más cerca posible de Y.

La mejor matriz ortogonal $\boldsymbol{Q}$ resuelve el problema de Procrustes de la Sección IV.9. Se encuentra a partir de la descomposición en valores singulares $X_N Y^{\mathrm{T}} = U\Sigma V^{\mathrm{T}}$. La matriz ortogonal Q que multiplicada por X_N da resultados más cercanos a que a los puntos en Y es $Q = VU^{\mathrm{T}}$.

J. C. Gower, Properties of Euclidean and non-Euclidean distance matrices, *Linear Algebra and Its Applications* **67** (1985) 81-97.

Problemas de la Sección IV.10

1 $||\boldsymbol{x}_1 - \boldsymbol{x}_2||^2 = 1$ y $||\boldsymbol{x}_2 - \boldsymbol{x}_3||^2 = 1$ y $||\boldsymbol{x}_1 - \boldsymbol{x}_3||^2 = 6$ no cumplen la desigualdad triangular. Construya G y confirme que no es semidefinida positiva : sin solución X en $G = X^{\mathrm{T}}X$.

2 $||\boldsymbol{x}_1 - \boldsymbol{x}_2||^2 = 9$ y $||\boldsymbol{x}_2 - \boldsymbol{x}_3||^2 = 16$ y $||\boldsymbol{x}_1 - \boldsymbol{x}_3||^2 = 25$ sí satisfacen la desigualdad triangular $3 + 4 > 5$. Construya G y encuentre puntos $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$ que coincidan con estas distancias.

3 Si cada $||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 = 1$ para $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4$, encuentre G y luego X. ¿ Para cuál dimensión d los puntos se encuentran en $\mathbf{R}^d$?

Parte V

Probabilidad y Estadística

V.1 **Media, varianza y probabilidad**

V.2 **Distribuciones de probabilidad**

V.3 **Momentos, cumulantes y desigualdades de la Estadística**

V.4 **Matrices de covarianza y probabilidades conjuntas**

V.5 **Gaussiana multivariada y mínimos cuadrados ponderados**

V.6 **Cadenas de Markov**

Parte V : Probabilidad y Estadística

Estos temas han aumentado considerablemente su importancia. Cuando se predice un resultado, necesitamos su probabilidad. Cuando medimos ese resultado, necesitamos sus estadísticas. Esos cálculos se restringían en el pasado a ecuaciones simples y muestras pequeñas. Ahora podemos resolver ecuaciones diferenciales para distribuciones de probabilidad (*ecuaciones maestras*). Podemos calcular los estadísticos de grandes muestras. Este capítulo apunta a una comprensión básica de las siguientes ideas clave :

1 Media m y varianza σ^2 : valor esperado y valor muestral

2 Distribución de probabilidad y distribución acumulada

3 Matriz de covarianza y probabilidades conjuntas

4 Distribución normal (gaussiana) : univariada y multivariada

5 Variable aleatoria estandarizada $(x - m)/\sigma$

6 El teorema del límite central

7 Distribución binomial y distribución uniforme

8 Desigualdades de Markov y de Chebyshev (distancia a la media)

9 Mínimos cuadrados ponderados y filtro de Kalman : $\widehat{\boldsymbol{x}}$ y su varianza

10 Matriz de Markov y cadena de Markov

V.1. Media, varianza y probabilidad

Comenzamos con los tres términos fundamentales de este capítulo: *media, varianza y probabilidad.* Antes de presentar las fórmulas, permítanme explicar a grandes rasgos sus significados:

La **media** es el *valor promedio* o valor esperado

La **varianza σ^2** mide el promedio de los *cuadrados de las distancias* a la media m

Las **probabilidades** de n resultados diferentes son los números positivos $p_1, \ldots, p_n$ que suman 1.

Sin duda la media es fácil de comprender. Comenzaremos por allí. Pero ya mismo tenemos dos situaciones diferentes que hay que tener en claro. Por un lado, podemos tener los resultados (*valores muestrales*) de un experimento realizado. Por otro lado, podemos tener los resultados esperados (*valores esperados*) de experimentos futuros. Veamos ejemplos de ambos:

Valores muestrales	Cinco ingresantes universitarios al azar con edades $\mathbf{18, 17, 18, 19, 17}$
Media muestral	$\frac{1}{5}(18 + 17 + 18 + 19 + 17) = \mathbf{17.8}$
Probabilidades	Las edades de los ingresantes son 17 ($\mathbf{20\,\%}$), 18 ($\mathbf{50\,\%}$), 19 ($\mathbf{30\,\%}$)

Un ingresante al azar tiene **edad esperada** $\mathbf{E}[x] = (0.2)\,17 + (0.5)\,18 + (0.3)\,19 = \mathbf{18.1}$

Ambos valores 17.8 y 18.1 son promedios correctos. La media muestral comienza con N muestras $x_1, \ldots, x_N$ de un experimento realizado. Su media es el *promedio* de las N muestras observadas:

$$\boxed{\textbf{Media muestral} \qquad m = \mu = \frac{1}{N}(x_1 + x_2 + \cdots + x_N)} \tag{1}$$

El **valor esperado de x** comienza con las probabilidades $p_1, \ldots, p_n$ de las edades $x_1, \ldots, x_n$:

$$\boxed{\textbf{Valor esperado} \quad m = \mathbf{E}[x] = p_1 x_1 + p_2 x_2 + \cdots + p_n x_n.} \tag{2}$$

Esto es $\boldsymbol{p} \cdot \boldsymbol{x}$. Observe que $m = \mathrm{E}[x]$ nos dice qué esperar, $m = \mu$ nos dice qué hemos obtenido.

Una moneda justa tiene una probabilidad $p_0 = \frac{1}{2}$ de que salga cruz y $p_1 = \frac{1}{2}$ de que salga cara. Así $\mathrm{E}[x] = \frac{1}{2}(0) + \frac{1}{2}(1)$. La fracción de caras luego de lanzar una moneda al aire N veces es la media de la muestra, y se espera que sea aproximadamente $\mathrm{E}[x] = \frac{1}{2}$. Al tomar muchas muestras (N grande), los resultados de la muestra se acercarán a las probabilidades. La **"ley de los grandes números"** dice que con probabilidad 1, la media muestral se acercará a su valor esperado $\mathrm{E}[x]$ a medida que el tamaño de la muestra N aumente.

Esto *no* quiere decir que porque hayamos visto más cruces que caras sea más probable que la siguiente muestra sea un cara. Las probabilidades siguen siendo 50-50. Los primeros 100 o 1000 lanzamientos de moneda afectan la media muestral. *Pero* 1000 *lanzamientos no afectan su límite*, porque estamos dividiendo por $N \to \infty$.

Varianza (alrededor de la media)

La **varianza** $\boldsymbol{\sigma^2}$ mide la distancia esperada (al cuadrado) a la media esperada $\mathrm{E}[x]$. La **varianza muestral** $\boldsymbol{S^2}$ mide la distancia real (al cuadrado) a la verdadera media muestral. La raíz cuadrada es la **desviación estándar** $\boldsymbol{\sigma}$ **o** $\boldsymbol{S}$. Después de un examen, envío μ y S a la clase. No conozco la media esperada y la varianza porque no conozco las probabilidades p_1 a p_{100} para cada resultado. (Luego de enseñar por 50 años, todavía no tengo idea de qué esperar.)

La desviación es siempre desviación *a la media*: esperada o muestral. Buscamos el tamaño de la "dispersión" alrededor del valor medio $x = m$. Comencemos con N muestras.

$$\textbf{Varianza muestral} \qquad \boldsymbol{S^2 = \frac{1}{N-1}\left[(x_1 - m)^2 + \cdots + (x_N - m)^2\right]} \tag{3}$$

Las edades en la muestra $x = 18, 17, 18, 19, 17$ tienen media $m = 17.8$. Esa muestra tiene varianza 0.7:

$$S^2 = \frac{1}{4}\left[(.2)^2 + (-.8)^2 + (.2)^2 + (1.2)^2 + (-.8)^2\right] = \frac{1}{4}(2.8) = \mathbf{0.7}$$

Los signos menos desaparecen cuando elevamos al cuadrado. Una observación: los estadistísticos dividen por $N - 1 = 4$ (y no $N = 5$) para que S^2 sea una estimación insesgada de σ^2. Un grado de libertad ya ha sido tenido en cuenta en la media muestral.

Una identidad importante proviene de escribir cada $(x - m)^2$ como $x^2 - 2mx + m^2$:

$$\begin{aligned} \text{suma de } (x_i - m)^2 &= (\text{suma de } x_i^2) - 2m(\text{suma de } x_i) + (\text{suma de } m^2) \\ &= (\text{suma de } x_i^2) - 2m(Nm) + Nm^2 \\ \textbf{suma de } \boldsymbol{(x_i - m)^2} &= \boldsymbol{(\text{suma de } x_i^2) - Nm^2}. \end{aligned} \tag{4}$$

Esta es una manera equivalente de encontrar $(x_1 - m)^2 + \cdots + (x_N - m^2)$, sumando $x_1^2 + \cdots + x_N^2$. Para encontrar la varianza muestral S^2, dividimos esto por $N - 1$.

Ahora comenzamos con las probabilidades p_i (¡ nunca negativas !) en lugar de muestras. Encontramos valores esperados en lugar de valores muestrales. La varianza σ^2 es el número crucial en estadística.

$$\textbf{Varianza} \qquad \boldsymbol{\sigma^2 = \mathrm{E}\left[(x - m)^2\right] = p_1(x_1 - m)^2 + \cdots + p_n(x_n - m)^2}. \tag{5}$$

Elevamos al cuadrado la distancia al valor esperado $m = \mathrm{E}[x]$. No tenemos muestras, solo esperanzas. Conocemos las probabilidades pero no los resultados del experimento.

Ejemplo 1 Encuentre la varianza σ^2 de las edades de los estudiantes de primer año.

Solución Las probabilidades de las edades $x_i = 17, 18, 19$ son $p_i = 0.2,\ 0.5$ y 0.3. El valor esperado es $\boldsymbol{m = \sum p_i x_i = 18.1}$. La varianza usa esas mismas probabilidades:

$$\begin{aligned} \boldsymbol{\sigma^2} &= (0.2)(\mathbf{17 - 18.1})^2 + (0.5)(\mathbf{18 - 18.1})^2 + (0.3)(\mathbf{19 - 18.1})^2 \\ &= (0.2)\mathbf{(1.21)} + (0.5)\mathbf{(0.01)} + (0.3)\mathbf{(0.81)} = \mathbf{0.49}. \end{aligned}$$

La **desviación estándar** es la raíz cuadrada $\boldsymbol{\sigma = 0.7}$.
Esto mide la dispersión de $17, 18, 19$ alrededor de $\mathrm{E}[x]$, ponderado por las probabilidades $0.2, 0.5, 0.3$.

La ecuación (4) da otra manera de calcular la varianza σ^2:

$$\boxed{\sigma^2 = \mathrm{E}\left[x^2\right] - (\mathrm{E}\left[x\right])^2 = \boldsymbol{\sum p_i\, x_i^2 - \left(\sum p_i\, x_i\right)^2}}$$

Distribuciones de probabilidad continua

Hasta ahora hemos permitido n resultados posibles $x_1, \ldots, x_n$. Con las edades 17, 18 y 19 solo tuvimos $n = 3$. Si medimos la edad en días en lugar de años, tendríamos miles de edades posibles (demasiadas). Sería mejor permitir *todo número entre* 17 *y* 20, un continuo de edades posibles. Así las probabilidades p_1, p_2, p_3 para las edades x_1, x_2, x_3 tienen que moverse a una **distribución de probabilidad $p(x)$** para un rango continuo completo de edades $17 \le x \le 20$.

La mejor manera de explicar distribución de probabilidades es con los siguientes dos ejemplos: la **distribución uniforme** y la **distribución normal**. La primera (uniforme) es sencilla. La distribución normal es importantísima.

Distribución uniforme Supongamos que las edades están distribuidas uniformemente entre 17.0 y 20.0. Todas las edades entre esos números son "igualmente probables". Por supuestos que una edad exacta no tiene chance. Hay probabilidad nula de que le acierte al número exacto $x = 17.1$ o $x = 17 + \sqrt{2}$. Lo que se puede obtener con certeza (utilizando la distribución uniforme) es **la probabilidad $F(x)$ de que la edad de un estudiante de primer año sea menor que x**:

La probabilidad de una edad menor que $x = 17$ es $F(17) = \mathbf{0}$	$x \le 17$ no ocurrirá
La probabilidad de una edad menor que $x = 20$ es $F(20) = \mathbf{1}$	$x \le 20$ ocurrirá
La probabilidad de una edad menor que x es $\boldsymbol{F(x) = \frac{1}{3}(x - 17)}$	**F está entre 0 y 1**

Esa fórmula $F(x) = \frac{1}{3}(x - 17)$ da $F = 0$ en $x = 17$; así $x \le 17$ no ocurrirá. Da $F(x) = 1$ en $x = 20$; así $x \le 20$ es seguro. Entre 17 y 20, el gráfico de la **distribución acumulada $F(x)$** aumenta linealmente para este modelo uniforme. He aquí los gráficos de $F(x)$ y su derivada $p(x) =$ "función de densidad de probabilidad".

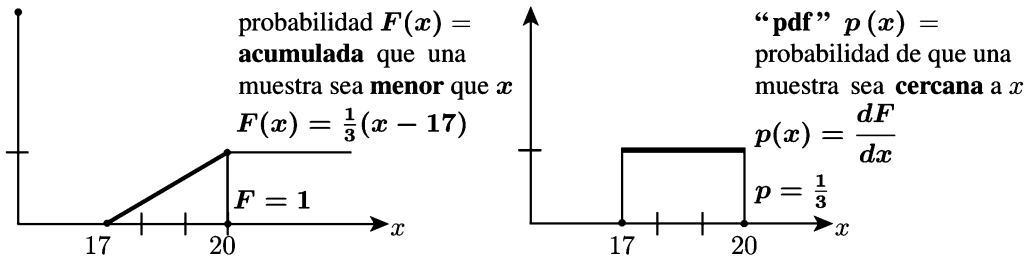


Figura V.1: $F(x)$ es la distribución acumulada y su derivada $p(x) = dF/dx$ es la **función de densidad de probabilidad** (**pdf**, por su sigla en inglés). Para esta **distribución uniforme**, $p(x)$ es constante entre 17 y 20. El área total debajo del gráfico de $p(x)$ es la probabilidad total $F = 1$.

Podría decirse que $p(x)\,dx$ es la probabilidad de que una muestra esté entre x y $x + dx$. Esto es "infinitésimamente cierto": $p(x)\,dx$ es $F(x + dx) - F(x)$. He aquí la conexión real entre $F(x)$ y $p(x)$:

$$\boldsymbol{F = \text{integral de } p} \qquad \textbf{Probabilidad de } a \le x \le b = \int_a^b p(x)\,dx = F(b) - F(a) \tag{6}$$

$F(b)$ es la probabilidad de que $x \le b$. Le restamos $F(a)$ para mantener $x \ge a$. Eso nos deja $a \le x \le b$.

Media y varianza de $p(x)$

¿ Qué son la media m y la varianza σ^2 de una distribución de probabilidad ? Antes sumábamos $\boldsymbol{p_i x_i}$ para obtener la media (valor esperado). Con una distribución continua **integramos** $\boldsymbol{x p(x)}$:

$$\textbf{Media} \qquad \boldsymbol{m = \mathbf{E}[x] = \int x\, p(x)\, dx = \int_{x=17}^{20} (x)\left(\frac{1}{3}\right) dx = 18.5}$$

Para esta distribución uniforme, la media m está a mitad de camino entre 17 y 20. Así la probabilidad de que un valor aleatorio x sea menor que la media $m = 18.5$ es $F(m) = \frac{1}{2}$.

En MATLAB, $x =$ rand (1) elige un número aleatorio uniformemente entre 0 y 1. Así la media esperada es $m = \frac{1}{2}$. El intervalo de 0 a x tiene probabilidad $F(x) = x$. El intervalo a la izquierd de la media m tiene siempre probabilidad $F(m) = \frac{1}{2}$.

La varianza es la media de la distancia al cuadrado a la media. Con N resultados, σ^2 es la suma de $p_i(x_i - m)^2$. Para una variable aleatoria continua x, la suma cambia a una **integral**.

$$\textbf{Varianza} \qquad \boldsymbol{\sigma^2 = \mathbf{E}\left[(x-m)^2\right] = \int p(x)\,(x-m)^2\, dx} \tag{7}$$

Cuando las edades son uniformes entre $17 \le x \le 20$, la integral puede trasladarse a $0 \le x \le 3$:

$$\sigma^2 = \int_{17}^{20} \frac{1}{3}(x-18.5)^2\, dx = \int_{0}^{3} \frac{1}{3}(x-1.5)^2\, dx = \frac{1}{9}(x-1.5)^3 \Big]_{x=0}^{x=3} = \frac{2}{9}(1.5)^3 = \frac{3}{4}.$$

Ese es un ejemplo típico para una $p(x)$ uniforme de 0 a a.

Distribución uniforme para $\boldsymbol{0 \le x \le a}$

Densidad $\boldsymbol{p(x) = \frac{1}{a}}$ **Acumulada** $\boldsymbol{F(x) = \frac{x}{a}}$

Media $\boldsymbol{m = \frac{a}{2}}$ **mitad de camino** **Varianza** $\boldsymbol{\sigma^2 = \int_0^a \frac{1}{a}\left(x - \frac{a}{2}\right)^2 dx = \frac{a^2}{12}}$ (8)

La media es un múltiplo de a, la varianza es un múltiplo de a^2. Para $a = 3$, $\sigma^2 = \frac{9}{12} = \frac{3}{4}$. Para un número aleatorio entre 0 y 1 $\left(\text{media } \frac{1}{2}\right)$ la varianza es $\sigma^2 = \frac{1}{12}$.

Distribución normal : curva acampanada

A la distribución normal también se la llama distribución "gaussiana" y es la más importante de todas las funciones de densidad de probabilidad $p(x)$. La razón de su suma importancia viene de promediar los resultados de repetir un experimento. Los experimentos tienen sus propias distribuciones (como la cara y la cruz). *El promedio se aproxima a una distribución normal.*

Teorema del límite central (informal) El promedio de N muestras de una distribución con probabilidad "cualquiera" se aproxima a una distribución normal a medida que $N \to \infty$ (se demostrará en la Sección V.3).

Comencemos con la "distribución normal estándar". Es simétrica alrededor de $x = 0$, por lo tanto el valor de su media es $m = 0$. Se elige con una varianza estándar $\sigma^2 = 1$. Se llama $\mathbf{N}(0, 1)$.

$$\textbf{Distribución normal estándar} \quad p(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}. \tag{9}$$

El gráfico de $p(x)$ es la **curva acampanada** en la Figura V.2. Lo estándar es

$$\textbf{Probabilidad total} = 1 \qquad \int_{-\infty}^{\infty} p(x)\,dx = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-x^2/2}\,dx = 1$$

$$\textbf{Media } \mathbf{E}[x] = 0 \qquad m = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} x e^{-x^2/2}\,dx = 0$$

$$\textbf{Varianza } \mathbf{E}[x^2] = 1 \qquad \sigma^2 = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} (x-0)^2 e^{-x^2/2}\,dx = 1$$

La media cero fue fácil porque estamos integrando una función impar. Cambiar x a $-x$ muestra que "integral $=$ $-$ integral". Entonces la integral debe ser $m = 0$.

Para las otras dos integrales se aplica la idea del Problema 13 para probar que dan 1. La Figura V.2 muestra un gráfico de $p(x)$ para la distribución normal $\mathbf{N}(0, \sigma^2)$ y también su distribución acumulada $F(x) =$ integral de $p(x)$. A partir de la simetría de $p(x)$ podemos observar que *la media* $=$ *cero*. En $F(x)$ vemos una aproximación práctica importante para las encuestas de opinión:

La probabilidad de que una muestra aleatoria caiga entre $-\sigma$ y σ es $\boldsymbol{F(\sigma) - F(-\sigma) \approx \frac{2}{3}}$.
Esto es así porque $\int_{-\sigma}^{\sigma} p(x)\,dx$ es igual a $\int_{-\infty}^{\sigma} p(x)\,dx - \int_{-\infty}^{-\sigma} p(x)\,dx = F(\sigma) - F(-\sigma)$.

Similarmente, la probabilidad de que un x aleatorio se encuentre entre -2σ y 2σ ("*menor que dos desviaciones estándar de la media*") es $F(2\sigma) - F(-2\sigma) \approx 0.95$. Si tenemos un resultado experimental alejado de la media en más de 2σ, podemos estar bastante seguros de que no es un accidente: la probabilidad es $= 0.05$. Las pruebas de drogas pueden buscar una confirmación más precisa, como una probabilidad 0.001. En la búsqueda del bosón de Higgs se usó una prueba súper estricta de 5σ de desviación para decir que no era mero accidente.

La distribución normal con una media m cualquiera y desviación estándar σ se obtiene de trasladar y estirar la estándar $\mathbf{N}(0, 1)$. **Trasladar de x a $x - m$. Estirar $x - m$ a $(x - m)/\sigma$.**

$$\textbf{Densidad gaussiana } p(x) \qquad p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-m)^2/2\sigma^2} \tag{10}$$
$$\textbf{Distribución normal } \mathbf{N}(m, \sigma^2)$$

La integral de $p(x)$ es $F(x)$: la probabilidad de que una muestra aleatoria sea menor que x. El diferencial $p(x)\,dx = F(x+dx) - F(x)$ es la probabilidad de que una muestra aleatoria esté entre x y $x+dx$. No existe una fórmula simple para integrar $e^{-x^2/2}$, entonces la distribución acumulada $F(x)$ se calcula y tabula con mucho cuidado.

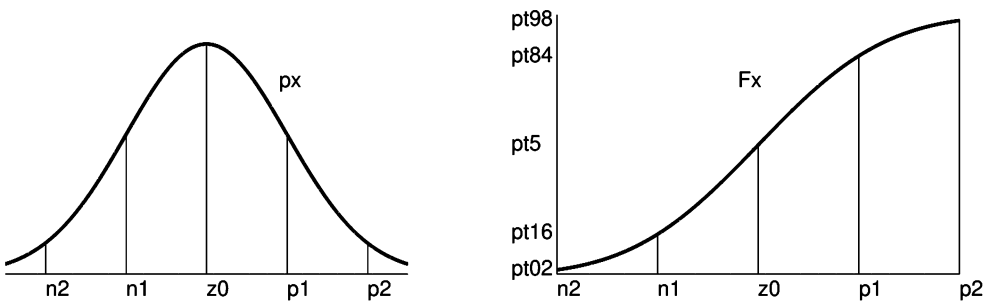


Figura V.2: la distribución normal estándar $p\,(x)$ tiene media $m = 0$ y $\sigma = 1$.

N Lanzamientos de monedas y $N \to \infty$

Ejemplo 2 **Supongamos que x es 1 o -1 con igual probabilidad : $p_1 = p_{-1} = \frac{1}{2}$.**

El valor medio es $\boldsymbol{m} = \frac{1}{2}(1) + \frac{1}{2}(-1) = \mathbf{0}$. La varianza es $\boldsymbol{\sigma^2} = \frac{1}{2}(1)^2 + \frac{1}{2}(-1)^2 = \mathbf{1}$.

La pregunta clave está en el *promedio* $A_N = (x_1 + \cdots + x_N)/N$. Los x_i independientes son ± 1 y dividimos su suma por N. La media esperada de A_N todavía es cero. La ley de los grandes números dice que el promedio muestral tiende a cero con probabilidad 1. ¿ Qué tan rápido A_N se aproxima a cero ? **¿ Cuál es su varianza $\boldsymbol{\sigma_N^2}$?**

$$\text{Por linealidad} \quad \boldsymbol{\sigma_N^2} = \frac{\sigma^2}{N^2} + \frac{\sigma^2}{N^2} + \cdots + \frac{\sigma^2}{N^2} = N\frac{\sigma^2}{N^2} = \frac{\mathbf{1}}{\boldsymbol{N}} \quad \text{ya que } \sigma^2 = 1. \tag{11}$$

Ejemplo 3 **Cambiar los resultados de 1 o -1 a $x = 1$ o $x = 0$. Mantener $p_1 = p_0 = \frac{1}{2}$.**

El nuevo valor medio $\boldsymbol{m} = \frac{\mathbf{1}}{\mathbf{2}}$ cae entre 0 y 1. La varianza cambia a $\boldsymbol{\sigma^2} = \frac{\mathbf{1}}{\mathbf{4}}$:

$$\boldsymbol{m} = \frac{1}{2}(1) + \frac{1}{2}(0) = \frac{\mathbf{1}}{\mathbf{2}} \quad \text{y} \quad \boldsymbol{\sigma^2} = \frac{1}{2}\left(1 - \frac{1}{2}\right)^2 + \frac{1}{2}\left(0 - \frac{1}{2}\right)^2 = \frac{\mathbf{1}}{\mathbf{4}}.$$

El promedio A_N ahora tiene media $\dfrac{1}{2}$ y varianza $\dfrac{1}{4N^2} + \cdots + \dfrac{1}{4N^2} = \dfrac{\mathbf{1}}{\mathbf{4N}} = \boldsymbol{\sigma_N^2}$. (12)

Esta σ_N es la mitad del tamaño de σ_N del Ejemplo 2. Esto debe ser correcto porque el nuevo rango 0 a 1 tiene longitud igual a la mitad de la distancia de -1 a 1. Los ejemplos 2-3 muestran una ley de linealidad.

La nueva variable x_{new} con valores 0–1, es $\frac{1}{2}\,x_{\text{old}} + \frac{1}{2}$. Así la media m aumenta a $\frac{1}{2}$ y la varianza es *multiplicada* por $\left(\frac{1}{2}\right)^2$. Una traslación cambia m y el reescalamiento cambia σ^2.

Linealidad $\quad x_{\text{new}} = ax_{\text{old}} + b$ tiene $m_{\text{new}} = am_{\text{old}} + b$ y $\sigma^2{}_{\text{new}} = a^2\sigma^2{}_{\text{old}}$ (13)

He aquí los resultados de tres pruebas numéricas: 0 o 1 aleatorio promediado en N pruebas.

[**48** unos de $N = \mathbf{100}$] [**5035** unos de $N = \mathbf{10000}$] [**19967** unos de $N = \mathbf{40000}$].

El X estandarizado $= (x - m)/\sigma = \left(A_N - \frac{1}{2}\right) \times 2\sqrt{N}$ fue [**−.40**] [**.70**] [**−.33**].

El teorema del límite central establece que el promedio de los lanzamientos de monedas se aproximará a una distribución normal. Veamos cómo ocurre: **binomial se aproxima a normal.**

Las probabilidades "binomiales" $p_0, \ldots, p_N$ cuentan la cantidad de caras en los N lanzamientos de monedas.

Por cada lanzamiento (justo), la probabilidad de cara es $\frac{1}{2}$. Para $N = 3$ lanzamientos, la probabilidad de cara en las tres instancias es $\left(\frac{1}{2}\right)^3 = \frac{1}{8}$. La probabilidad de cara dos veces y cruz una es $\frac{3}{8}$, de las tres secuencias HHT, HTH y THH (H:*heads*:cara, T:*tail*:cruz). Estos números $\frac{1}{8}$ y $\frac{3}{8}$ son términos de $\left(\frac{1}{2} + \frac{1}{2}\right)^3 = \frac{1}{8} + \frac{3}{8} + \frac{3}{8} + \frac{1}{8} = 1$. *La cantidad promedio de caras en* 3 *lanzamientos es* 1.5.

$$\textbf{Media} \quad m = (3\text{ caras})\frac{1}{8} + (2\text{ caras})\frac{3}{8} + (1\text{ cara})\frac{3}{8} + 0 = \frac{3}{8} + \frac{6}{8} + \frac{3}{8} = \mathbf{1.5\text{ caras}}$$

Con N lanzamientos, el Ejemplo 3 (o el sentido común) nos dan una media de $\boldsymbol{m = \Sigma\, x_i p_i = \frac{1}{2}N}$ caras.

La varianza σ^2 se basa en el *cuadrado de la distancia* a esta media $N/2$. Con $N = 3$ la varianza es $\sigma^2 = \frac{3}{4}$ (*es decir* $N/4$). Para hallar σ^2 sumamos $(x_i - m)^2\, p_i$ con $m = 1.5$:

$$\boldsymbol{\sigma^2} = (3 - 1.5)^2\,\frac{1}{8} + (2 - 1.5)^2\,\frac{3}{8} + (1 - 1.5)^2\,\frac{3}{8} + (0 - 1.5)^2\,\frac{1}{8} = \frac{9 + 3 + 3 + 9}{32} = \frac{\mathbf{3}}{\mathbf{4}}.$$

Para cualquier N, la varianza para una distribución binomial es $\boldsymbol{\sigma_N^2 = N/4}$. Así $\sigma_N = \sqrt{N}/2$.

La Figura V.3 muestra cómo las probabilidades de 0, 1, 2, 3, 4 caras en $N = 4$ se acercan a una gaussiana acampanada. Esa gaussiana esta centrada en el valor de la media $m = N/2 = 2$. Para obtener la gaussiana estándar (media 0 y varianza 1) trasladamos y reescalamos el gráfico. Si x es el número de caras en N lanzamientos —el promedio de N resultados cero-uno— entonces x es trasladado por su media $m = N/2$ y reescalado por $\sigma = \sqrt{N}/2$ para producir la X estándar:

Trasladada y escalada $\quad \boldsymbol{X = \dfrac{x - m}{\sigma} = \dfrac{x - \frac{1}{2}N}{\sqrt{N}/2}} \quad (\boldsymbol{N = 4}$ tiene $\boldsymbol{X = x - 2})$

Restar m es "centrar". La media de X es cero.

Dividir por σ es "normalizar" o "estandarizar". La varianza de X es 1.

Es divertido ver cómo el teorema del límite central da la respuesta correcta en el punto medio $X = 0$. En ese punto, el factor $e^{-X^2/2}$ es 1. Sabemos que la varianza para N lanzamientos de monedas es $\sigma^2 = N/4$. El centro de la curva acampanada tiene altura $1/\sqrt{2\pi\sigma^2} = \sqrt{\mathbf{2/N\pi}}$.

¿ Cuál es la altura en el centro de la distribución de lanzamientos de moneda p_0 a p_N (la distribución binomial) ? Para $N = 4$, las probabilidades para $0, 1, 2, 3, 4$ caras se obtiene de $\left(\frac{1}{2} + \frac{1}{2}\right)^4$.

Probabilidad del punto medio $\mathbf{\frac{6}{16}}$ $$\left(\frac{1}{2} + \frac{1}{2}\right)^4 = \frac{1}{16} + \frac{4}{16} + \frac{\mathbf{6}}{\mathbf{16}} + \frac{4}{16} + \frac{1}{16} = 1.$$

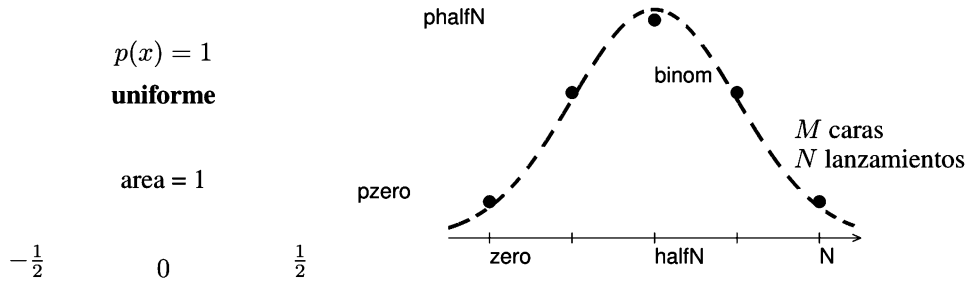


Figura V.3: las probabilidades $p = (1, 4, 6, 4, 1)/16$ para el número de caras en 4 lanzamientos. Estos p_i se aproximan a una distribución gaussiana con varianza $\sigma^2 = N/4$ centrada en $m = N/2$. Para X, el teorema del límite central da convergencia a la distribución normal $\mathbf{N}(0, 1)$.

El teorema binomial en el Problema 11 nos dice la probabilidad del punto medio $p_{N/2}$ para un N par :

La probabilidad del punto medio $\left(\frac{N}{2} \text{ cara}, \frac{N}{2} \text{ cruz}\right)$ es $\frac{1}{2^N} \frac{N!}{(N/2)!\,(N/2)!}$

Para $N = 4$, esos factoriales producen $4!/2!\,2! = 24/4 = 6$. Para N grande, la fórmula de Stirling $\sqrt{2\pi N}(N/e)^N$ es una buena aproximación a $N!$. Usemos esta fórmula una vez para N y dos veces para $N/2$:

Límite del lanzamiento de moneda
Probabilidad del punto medio
$$p_{N/2} \approx \frac{1}{2^N} \frac{\sqrt{2\pi N}(N/e)^N}{\pi N(N/2e)^N} = \frac{\sqrt{2}}{\sqrt{\pi N}} = \frac{1}{\sqrt{2\pi}\sigma}. \quad (14)$$

Para el último paso se usó la varianza $\sigma^2 = N/4$ para los lanzamientos de monedas. El resultado $1/\sqrt{2\pi}\sigma$ muestra que el valor medio coincide para la gaussiana. El teorema del límite central es cierto :

la distribución binomial centrada se aproxima a la distribución normal $p(x)$ cuando $N \to \infty$.

Métodos de estimación de Monte Carlo

El cálculo científico trabaja con errores en los datos. El cálculo financiero trabaja con cifras y predicciones inciertas. Gran parte de la matemática aplicada se encuentra con este problema: **aceptar la incertidumbre de los datos recibidos y estimar la varianza de los resultados**.

¿ Cómo se estima esa varianza? A menudo, la distribuciones de probabilidades $p(x)$ son desconocidas. Lo que podemos hacer es intentar con diferentes datos de entrada $\boldsymbol{b}$, calcular los resultados $\boldsymbol{x}$ y promediar. Esta es la forma más sencilla del **método de Monte Carlo** (cuyo nombre está basado en el palacio de apuestas de la Riviera, donde una vez vi una pelea sobre si la apuesta se había realizado a tiempo).
Monte Carlo aproxima un valor esperado $\mathrm{E}[x]$ *con un promedio muestral* $(x_1 + \cdots + x_N)/N$.

Es importante entender que cada x_k puede ser costoso de calcular. No estamos lanzando monedas. Cada muestra se obtiene de un conjunto de datos b_k. *Monte Carlo elige aleatoriamente estos datos* b_k*, calcula los resultados* x_k *y luego promedia esos* x. Para una precisión decente de $\mathrm{E}[x]$ a menudo se necesitan muchas muestras b y un costo computacional elevado. El error al aproximar $\mathrm{E}[x]$ por $(x_1+\cdots+x_N)/N$ suele ser de orden $1/\sqrt{N}$. *Una lenta mejora a medida que* N *aumenta.*

Esta estimación $1/\sqrt{N}$ se obtuvo a partir de los lanzamientos de moneda en la ecuación (11). Promediar N muestras independientes x_k con varianza σ^2 reduce la varianza a σ^2/N.

"Quasi-Monte Carlo" puede a veces reducir esta varianza a σ^2/N^2, que es una gran diferencia. Las entradas b_k se seleccionan con mucho cuidado, no tan solo al azar. En la revista *Acta Numerica*, 2013 se presenta un estado de arte de este QMC. Michael Giles desarrolló la nueva idea de "Monte Carlo multinivel" en *Acta Numerica* 2015. He aquí cómo funciona.

Supongamos que sea más simple simular otra variable $y(b)$ cercana a $x(b)$. Entonces usamos N cálculos de $y(b_k)$ y solo $N^* < N$ cálculos de $x(b_k)$ para estimar $\mathrm{E}[x]$.

Monte Carlo con 2 niveles
$$\mathrm{E}[\boldsymbol{x}] \approx \frac{\boldsymbol{1}}{\boldsymbol{N}} \sum_{\boldsymbol{1}}^{\boldsymbol{N}} \boldsymbol{y(b_k)} + \frac{\boldsymbol{1}}{\boldsymbol{N^*}} \sum_{\boldsymbol{1}}^{\boldsymbol{N^*}} \boldsymbol{[x(b_k) - y(b_k)]}$$

La idea es que $x - y$ tiene una varianza σ^* más pequeña que el x original. Por lo tanto N^* puede ser más pequeño que N, con la misma precisión para $\mathrm{E}[x]$. Realizamos N simulaciones económicas para encontrar los y. Esas cuestan C cada una. Solo realizamos N^* simulaciones costosas con las x. Esas cuestan C^* cada una. El costo total de cálculo es $NC + N^*C^*$.

Utilizando el cálculo podemos minimizar la varianza total para un costo total fijo. El cociente óptimo N^*/N es $\sqrt{C/C^*}\ \sigma^*/\sigma$. Un Monte Carlo con tres niveles simula x, y, y z :

$$\mathrm{E}[x] \approx \frac{1}{N} \sum_{1}^{N} \boldsymbol{z(b_k)} + \frac{1}{N^*} \sum_{1}^{N^*} \boldsymbol{[y(b_k) - z(b_k)]} + \frac{1}{N^{**}} \sum_{1}^{N^{**}} \boldsymbol{[x(b_k) - y(b_k)]}\,.$$

Giles optimiza $N, N^*, N^{**}, \ldots$ para mantener $\mathrm{E}[x] \le \mathrm{E}_0$ fijo y proporciona un código de MATLAB.

Repaso : tres fórmulas para la media y la varianza

Las fórmulas para m y σ^2 son el punto de inicio para toda la probabilidad y la estadística. Existen tres casos distintos que debemos tener en cuenta : valores **muestrales** X_i, valores **esperados** (p_i**discreto**), valores **esperados** ($p(x)$ **continuo**). He aquí la media m y la varianza S^2 o σ^2 :

Muestras X_1 a X_N	$\boldsymbol{m} = \dfrac{X_1 + \cdots + X_N}{N}$	$\boldsymbol{S^2} = \dfrac{(X_1 - m)^2 + \cdots + (X_N - m)^2}{N - 1}$
Suma de los resultados x_i multiplicados por las probabilidades p_i	$\boldsymbol{m} = \sum_1^n p_i x_i$	$\boldsymbol{\sigma^2} = \sum_1^n p_i (x_i - m)^2$
Integral de los resultados x con densidad de probabilidad $\rho(x)$	$\boldsymbol{m} = \int x\, p(x)\, dx$	$\boldsymbol{\sigma^2} = \int (x - m)^2\, p(x)\, dx$

Problemas de la Sección V.1

1 La línea 1 de la tabla anterior no tiene probabilidades p. ¿Cuál es el paralelo entre estas fórmulas ? Respuesta : *esperamos que una fracción p_i de las muestras sea $X = x_i$*. Si esto es exactamente cierto, $X = x_i$ se repite _____ vez/veces. Así las fórmulas de las líneas 1 y 2 dan el mismo m.

Cuando trabajamos con muestras, solo incluimos cada resultado X tan a menudo como aparece. Obtenemos la media "empírica" (línea 1) en lugar de la media esperada.

2 Sume 7 a cada resultado x. ¿ Qué le ocurre a la media y la varianza ? ¿ Cuáles son las nuevas media muestral, media esperada y varianza ?

3 Sabemos: $\frac{1}{3}$ de todos los números enteros son divisibles por 3 y $\frac{1}{7}$ de los números enteros son divisibles por 7. ¿ Qué fracción de los números enteros será divisible por 3, 7 o ambos ?

4 Supongamos que tomamos muestras de los números del 1 al 1000 con igual probabilidad $1/1000$. ¿ Cuáles son las probabilidades p_0 a p_9 de que el último dígito de una muestra sea $0, \ldots, 9$? ¿ Cuál es la media esperada m de ese último dígito? ¿ Cuál es su varianza σ^2 ?

5 Tome muestras nuevamente del 1 al 1000 pero observe el *cuadrado* del último dígito de la muestra . Ese cuadrado puede terminar en $x = 0, 1, 4, 5, 6,$ o 9. ¿ Cuáles son las probabilidades $p_0, p_1, p_4, p_5, p_6, p_9$? ¿ Cuál es la media (esperada) m y la varianza σ^2 de ese número x ?

6 (un poquito engañoso) Tome muestras de nuevo del 1 al 1000 con probabilidades iguales y sea x el *primer* dígito ($x = 1$ si el número es 15). ¿ Cuáles son las probabilidades p_1 a p_9 (deben sumar 1) de $x = 1, \ldots, 9$? ¿ Cuál es la media y la varianza de x ?

7 Suponga que tiene $N = 4$ muestras: $157, 312, 696, 602$. ¿ Cuáles son los *primeros dígitos* de 157^2, 312^2, 696^2, 602^2 ? ¿ Cuál es la media muestral μ ? ¿ Cuál es la varianza muestral S^2 ? Recuerde dividir por $N - 1 = 3$ y no por $N = 4$.

8 La ecuación (4) dio una segunda forma equivalente para calcular S^2 (la varianza muestral):

$$\boldsymbol{S^2} = \frac{1}{N-1}\ \text{suma de}\ (x_i - m)^2 = \frac{1}{N-1}\left[(\text{suma de } x_i^2) - Nm^2\right].$$

Verifique la identidad correspondiente para la varianza σ^2 (con $m = \Sigma p_i\, x_i$):

$$\boldsymbol{\sigma^2 = \text{ suma de } p_i\,(x_i - m)^2 = (\text{suma de } p_i\, x_i^2) - m^2.}$$

9 Si 24 muestras de una población producen la misma edad $x = 20$, ¿cuáles son la media muestral μ y la varianza muestral S^2? ¿Y si $x = 20$ o 21, 12 veces cada una?

10 Experimento en la computadora: encuentre el promedio $A_{1000000}$ de un millón de muestras 0-1 aleatorias. ¿Cuál es el valor de la variable estandarizada $X = \left(A_N - \frac{1}{2}\right) \times 2\sqrt{N}$?

11 La probabilidad p_i de obtener i caras en N lanzamientos de monedas es el *número binomial* $b_i = \binom{N}{i}$ dividido por 2^N. La suma de los b_i es $(1+1)^N = 2^N$ de manera que las probabilidades p_i suman 1.

$$p_0 + \cdots + p_N = \left(\frac{1}{2} + \frac{1}{2}\right)^N = \frac{1}{2^N}(b_0 + \cdots + b_N) \text{ con } b_i = \frac{N!}{i!\,(N-i)!}$$

$N = 4$ lleva a $b_0 = \dfrac{24}{24}$, $b_1 = \dfrac{24}{(1)(6)} = 4$, $b_2 = \dfrac{24}{(2)(2)} = 6$, $p_i = \dfrac{1}{16}(1, 4, 6, 4, 1)$.

Observe que $b_i = b_{N-i}$. *Problema*: Confirme que la media $m = 0p_0 + \cdots + Np_N$ es igual a $\frac{N}{2}$.

12 Para cualquier función $f(x)$ el valor esperado es $\mathrm{E}[f] = \sum p_i\, f(x_i)$ o $\int p(x)\, f(x)\, dx$ (probabilidad discreta o continua). La función puede ser x, $(x-m)^2$ o x^2.

Si la media es $\mathrm{E}[x] = m$ y la varianza es $\mathrm{E}[(x-m)^2] = \sigma^2$ **¿cuánto es $\mathrm{E}[x^2]$?**

13 Muestre que la distribución normal estándar $p(x)$ tiene probabilidad total $\int p(x)\, dx = 1$ como se requiere. Un truco famoso multiplica $\int p(x)\, dx$ por $\int p(y)\, dy$ y calcula la integral sobre todo x y todo y ($-\infty$ a ∞). El truco es reemplazar $dx\, dy$ en esa doble integral por $r\, dr\, d\theta$ (coordenadas polares con $x^2 + y^2 = r^2$). Explique cada paso:

$$2\pi \int_{-\infty}^{\infty} p(x)\, dx \int_{-\infty}^{\infty} p(y)\, dy = \iint_{-\infty}^{\infty} e^{-(x^2+y^2)/2} dx\, dy = \int_{\theta=0}^{2\pi} \int_{r=0}^{\infty} e^{-r^2/2}\, r\, dr\, d\theta = \boldsymbol{2\pi}.$$

V.2. Distribuciones de probabilidad

Las aplicaciones de probabilidad comienzan con los números $p_0, p_1, p_2 \ldots$ que dan la probabilidad de cada resultado posible. Para probabilidades continuas, tenemos una función de densidad $p(x)$. La probabilidad total es siempre $\sum p_i = 1$ o $\int p(x)\,dx = 1$.

Hay docenas de posibilidades famosas y útiles para p. Hemos elegido siete que son especialmente importantes : dos con probabilidades discretas $p_0, p_1, p_2 \ldots$ y cinco con probabilidades continuas $p(x)$ o $p(x, y)$.

Binomial	Lanzar una moneda n veces
Poisson	Eventos raros o infrecuentes
Exponencial	Olvida el pasado
Gaussiano = Normal	Promedios de muchos intentos
Log normal	El logaritmo tiene distribución normal
Chi cuadrado	Distancia al cuadrado
Gaussiana multivariable	Probabilidades para un vector (en V.5)

Cada una de esas tiene una media y una varianza (dispersión alrededor de la media). Queremos saber esos números fundamentales μ y σ^2. Wikipedia está organizada de una manera útil con gráficos de $p(x)$ y su integral $\Phi(x)$ con esa información (la distribución acumulada = probabilidad hasta x). Hay una lista sistemática de otras propiedades de esas siete y otras distribuciones de probabilidad especiales, y de otras también.

1. Distribución binomial

En cada intento, el resultado es 1 o 0 (éxito o fallo, cara o cruz). La probabilidad de éxito es $p_{1,1} = p$. La probabilidad de fracaso es $p_{0,1} = 1 - p = q$. Para una moneda justa $p = \frac{1}{2}$.

Las probabilidades de $0, 1, 2$ éxitos en $n = 2$ intentos son

$$p_{0,2} = (1-p)^2 \qquad p_{1,2} = 2p(1-p) \qquad p_{2,2} = p^2 \tag{1}$$

La probabilidad de exactamente k éxitos en n intentos involucra el coeficiente binomial $\binom{n}{k}$:

$$\boldsymbol{p_{k,n} = \binom{n}{k} p^k (1-p)^{n-k}} \quad \text{con} \quad \binom{n}{k} = \frac{n!}{k!\,(n-k)!} \quad \text{y} \quad 0! = 1 \tag{2}$$

Para $n = 2$ esos coeficientes binomiales son $1, 2, 1$ como se muestra en la ecuación (1) :

$$\binom{2}{0} = \frac{2!}{0!\ 2!} = \mathbf{1} \qquad \binom{2}{1} = \frac{2!}{1!\ 1!} = \mathbf{2} \qquad \binom{2}{2} = \frac{2!}{2!\ 0!} = \mathbf{1}$$

Para una moneda justa, $k = 0, 1, 2$ éxitos en $n = 2$ intentos tienen $p_{0,2} = \frac{1}{4}$ y $p_{1,2} = \frac{1}{2}$ y $p_{2,2} = \frac{1}{4}$.

Valor medio de un intento $\mu = (0)p_0 + (1)p_1 = (0)(1-p) + (1)p$	$\boldsymbol{\mu = \mathrm{E}[x] = p}$	(3)
Distribución binomial **Valor medio μ_n de n intentos es** $n\mu$	$\boldsymbol{\mu_n = np}$	(4)
Varianza de un intento $\sigma^2 = \mathrm{E}[(X-\mu)^2] = (1-p)(0-\mu)^2 + p(1-\mu)^2$ $= (1-p)p^2 + p(1-p)^2 = p(1-p)(p+1-p)$	$\boldsymbol{\sigma^2 = p(1-p)}$	(5)
Distribución binomial **Varianza σ_n^2 de n intentos es** $n\sigma^2$	$\boldsymbol{\sigma_n^2 = np(1-p)}$	(6)

Estos resultados para n intentos independientes se calculan fácil: *multiplicar por* n. Las mismas respuestas se obtienen con más trabajo a partir de las probabilidades binomiales $p_{k,n}$ de la ecuación (2) para k éxitos. La suma de $kp_{k,n}$ también es $\boldsymbol{\mu_n = np}$ y la suma de $(k-\mu_n)^2 p_{k,n}$ es $\boldsymbol{\sigma_n^2 = np(1-p)}$.

2. Distribución de Poisson

La *distribución de Poisson* es fascinante porque hay distintas maneras de abordarla. Una manera es conectarla directamente con la *distribución binomial* (tenemos probabilidad p de éxito en cada intento y $p_{k,n}$ de k éxitos en n intentos). Así Poisson explica esta situación límite de **eventos raros con λ éxitos en muchos intentos:**

$\boldsymbol{p \to 0}$ La probabilidad de éxito de cada intento p es pequeña (tiende a cero)
$\boldsymbol{n \to \infty}$ La cantidad de intentos n es grande (tiende a infinito)
$\boldsymbol{np = \lambda}$ La media (esperada) de la cantidad de éxitos en n intentos es $\boldsymbol{\lambda = np}$ **= constante**

¿Cuáles son las probabilidades de $0, 1, 2$ éxitos en n intentos cuando $p \to 0$ y $np = \lambda$?

n fracasos, 0 éxitos Probabilidad $p_{0,n} = (1-p)^n = \left(1 - \dfrac{\lambda}{n}\right)^n \to \boldsymbol{e^{-\lambda}}$

$n-1$ fracasos, 1 éxito Probabilidad $p_{1,n} = np(1-p)^{n-1} = \dfrac{\lambda}{1-p}\left(1 - \dfrac{\lambda}{n}\right)^n \to \boldsymbol{\lambda e^{-\lambda}}$

$n-2$ fracasos, 2 éxitos Probabilidad $p_{2,n} = \dfrac{1}{2}n(n-1)\,p^2(1-p)^{n-2}$

$$= \frac{1}{2}\frac{(\lambda^2 - \lambda p)}{(1-p)^2}\left(1 - \frac{\lambda}{n}\right)^n \to \boldsymbol{\frac{1}{2}\lambda^2 e^{-\lambda}}$$

En cada paso aplicamos los mismos resultados clave aprendidos en algún curso de cálculo:

$$\left(1 + \frac{1}{n}\right)^n \to \boldsymbol{e} \qquad \left(1 + \frac{\lambda}{n}\right)^n \to \boldsymbol{e^{\lambda}} \qquad \left(1 - \frac{\lambda}{n}\right)^n \to \boldsymbol{e^{-\lambda}}$$

Para k éxitos en n ensayos, donde cada uno tiene probabilidad $p = \lambda/n$, la probabilidad binomial $p_{n,k}$ se acerca a la probabilidad de Poisson $P_k = \lambda^k e^{-\lambda}/k\,!$

$$\boxed{\textbf{Probabilidad de Poisson} \quad \boldsymbol{P_k = \frac{\lambda^k}{k\,!}\, e^{-\lambda}}} \tag{7}$$

La suma de las probabilidades de Poisson P_k es igual a 1 :

$$\boldsymbol{P_0 + P_1 + P_2 + P_3 + \cdots = e^{-\lambda}\left(1 + \lambda + \frac{\lambda^2}{2\,!} + \frac{\lambda^3}{3\,!} + \cdots\right) = e^{-\lambda}e^{\lambda} = 1.}$$

Ahora haremos el cálculo de la media de Poisson (λ) y la varianza (también λ).

La media de la distribución de Poisson es λ. La manera lenta de encontrarla es

$$\boldsymbol{\mu_P} = 0P_0 + 1P_1 + 2P_2 + \cdots = e^{-\lambda}\left(0 + \lambda + \frac{\lambda^2}{1\,!} + \frac{\lambda^3}{2\,!} + \cdots\right) = e^{-\lambda}(\lambda e^{\lambda}) = \boldsymbol{\lambda}$$

La manera rápida: **Media de Poisson** $\boldsymbol{\mu_P}$ = límite de la media binomial np. Luego $\boldsymbol{\mu_P = \lambda}$

Varianza $\boldsymbol{\sigma_P^2}$ = límite de la varianza binomial $np(1-p)$. Luego $\boldsymbol{\sigma_P^2 = \lambda}$

Aplicaciones de Poisson

Asociamos Poisson con **eventos infrecuentes o raros** (tienen $p \to 0$). Pero esperamos un largo tiempo o incluimos una población grande para obtener una probabilidad decente de que uno o más eventos efectivamente ocurran. Poisson no se usa para la probabilidad de que te caiga un rayo : digamos que es una en un millón en toda tu vida. Se usa para la probabilidad de que le caiga un rayo a alguien en una ciudad de $100,000$ habitantes. En este caso $\boldsymbol{\lambda = pn} = 100,000/1,000,000 = \frac{1}{10}$. Esta es la cantidad esperada de que le caiga un rayo a una persona en esa ciudad.

La distribución de Poisson a menudo se aplica a eventos infrecuentes en un período extenso :

Cantidad de meteoros que caen en la Tierra

Cantidad de acampantes atacados por pumas

Cantidad de veces que los bancos grandes fracasan

Poisson supone eventos independientes. Es posible que esos ejemplos no encajen exactamente. Uno de los aspectos más complicados de la probabilidad aplicada es estimar la dependencia entre eventos. Por ejemplo, que un banco fracase puede significar que otros también lo hagan.

La suposición **i i d** significa **i**ndependientes e **i**dénticamente **d**istribuidos, que no siempre es cierta.

3. Distribución exponencial

La distribución exponencial es continua (no discreta). Describe el **tiempo de espera en un proceso de Poisson**. ¿ Cuánto falta para que caiga un rayo en su ciudad ? La suposición principal es :

El tiempo de espera es independiente del tiempo que ya haya esperado.

El futuro es independiente del pasado. ¿ Es esto cierto para un televisor destinado a fallar ? ¿ Es esto cierto para un ordenador destinado a fallar ? Si el fallo depende de un accidente aleatorio, el tiempo de espera tiene una distribución exponencial. La tasa de fallo λ es constante. Si el fallo es consecuencia de un deterioro lento, el requisito de independencia no se verifica.

La función de densidad de probabilidad (pdf, por su sigla en inglés) para una distribución exponencial es

$$p(x) = \lambda e^{-\lambda x} \quad \text{para} \quad x \geq 0 \tag{8}$$

La distribución acumulada (probabilidad de un evento antes del tiempo t) **es la integral de p** :

$$F(t) = \int_0^t \lambda e^{-\lambda x}\, dx = \left[-e^{-\lambda x} \right]_{x=0}^{x=-t} = 1 - e^{-\lambda t} \tag{9}$$

La **media** de $p(x)$ es el **tiempo de espera promedio** :

$$\mu = \int_0^\infty x\, p(x)\, dx = \int_0^\infty x \lambda e^{-\lambda x}\, dx = \frac{1}{\lambda} \tag{10}$$

La **varianza** de $p(x)$ es el valor esperado de $(x-\mu)^2$:

$$\sigma^2 = \int_0^\infty \left(x - \frac{1}{\lambda} \right)^2 \lambda e^{-\lambda x}\, dx = \frac{1}{\lambda^2} \tag{11}$$

Esa ecuación para la media $\mu = 1/\lambda$ no es una sorpresa. Si las mesas en un restaurante se liberan a una razón promedio de 3 mesas por hora (noche y día) entonces la espera promedio (el tiempo de espera esperado) es de 20 minutos.

Observe que este número $\beta = \dfrac{1}{\lambda}$ a menudo se usa en lugar de λ.

$$\textbf{Distribución exponencial} \quad p(x) = \frac{1}{\beta} e^{-x/\beta} \qquad \textbf{Media} \quad \mu = \beta. \tag{12}$$

La distribución exponencial no tiene memoria : la probabilidad de esperar al menos y horas más por una mesa, desafortunadamente no se ve afectada por ya haber estado esperando x horas :

$$\textbf{Sin memoria} \quad \textbf{Prob}\{t > x + y \ \textbf{dado que} \ t > x\} = \textbf{Prob}\{t > y\}. \tag{13}$$

Esto se reduce a una simple identidad de integrales de $p(t) = \lambda e^{-\lambda t}$:

$$\int_{x+y}^\infty \lambda e^{-\lambda t}\, dt \Big/ \int_x^\infty \lambda e^{-\lambda t}\, dt = \int_y^\infty \lambda e^{-\lambda t}\, dt \quad \text{es} \ \ e^{-\lambda(x+y)} / e^{-\lambda x} = e^{-\lambda y}.$$

Otro dato destacable. Supongamos que su teléfono y su computadora tienen tasa de fallo λ_p and λ_c y que sus fallas son independientes. ¿ Cuál es la distribución de probabilidad $p(t_{\text{min}})$ del tiempo t_{min} del primer fallo ? Respuesta : esta distribución es **exponencial con una tasa de fallo** $\boldsymbol{\lambda_{\text{tel}} + \lambda_{\text{comp}}}$. Miremos la supervivencia en lugar del fracaso en la ecuación (9), y multipliquemos :

$$\text{Prob}\{t_{\text{min}} > t\} = \big[\text{Prob}\{t_{\textbf{tel}} > t\}\big]\big[\text{Prob}\{t_{\textbf{comp}} > t\}\big] = \boldsymbol{e^{-\lambda_{\text{tel}} t}\, e^{-\lambda_{\text{comp}} t} = e^{-(\lambda_{\text{tel}} + \lambda_{\text{comp}})t}}.$$

Si las fallas solo ocurren en tiempos enteros $t = 0, 1, 2, 3, \ldots$, entonces la distribución exponencial (que tiene tiempo continuo) se reemplaza por la distribución geométrica (tiempo discreto).

$$\textbf{La probabilidad de fallo a tiempo } \boldsymbol{n} \textbf{ es } \boldsymbol{p_n = (1-\alpha)\alpha^n}. \tag{14}$$

Se incluye el factor $1 - \alpha$ para que $p_0 + p_1 + \cdots = (1-\alpha) + (\alpha - \alpha^2) + \cdots = 1$. La probabilidad total es 1.

4. Distribución normal (Distribución gaussiana)

La distribución normal se encuentra en el centro de la teoría de la probabilidad. Se produce cuando promediamos los resultados de otra distribución. Al mismo tiempo nos lleva a nuevas distribuciones que veremos a continuación : **log-normal, normal multivariable y "chi cuadrado".**

A menudo se obtienen fórmulas lindas, aunque la integral de e^{-x^2} no es una función elemental. Esa integral (de 0 a x) no puede expresarse en términos de exponenciales y potencias de x. La gracia salvadora es que las integrales de $-\infty$ a $+\infty$ sí tienen formas simples. La integral de $e^{-x^2/2}$ es $\sqrt{2\pi}$, así que dividimos por ese número.

Media 0, varianza 1 La distribución normal estándar $\mathbf{N(0,1)}$ tiene $\boldsymbol{p(x) = \dfrac{1}{\sqrt{2\pi}} e^{-x^2/2}}$

Media μ, varianza σ^2 La distribución $\mathbf{N}(\boldsymbol{\mu, \sigma^2})$ tiene $\boldsymbol{p(x) = \dfrac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}}$ (15)

Estas distribuciones son simétricas alrededor del punto $x = \mu$. La función $p(x)$ crece hasta ese punto y decrece para $x > \mu$. La segunda derivada es negativa entre $x = \mu - \sigma$ y $\mu + \sigma$. Esos son los puntos de inflexión donde $d^2p/dx^2 = 0$. Fuera de ese intervalo, la función es convexa ($d^2p/dx^2 > 0$). 67 % del área debajo de $p(x)$ se encuentra dentro de ese intervalo. Así la probabilidad de $|x - \mu| < \sigma$ es 0.67.

La probabilidad de $|x - \mu| < 2\sigma$ es 95 % (x está a menos de 2 desviaciones estándar de su valor medio μ). Es la famosa curva acampanada (Figura V.2). *No* tiene una cola pesada.

Pregunta Si $p(x) = e^{-ax^2+bx+c}$ es una distribución de probabilidad, ¿ quiénes son μ y σ ?

Respuesta Completamos cuadrados en ese exponente para obtener $-ax^2 + bx + c$ igual a $-a\left(x - \frac{b}{2a}\right)^2$ más una constante. De esta forma identificamos la media μ como $b/2a$. El número a afuera del paréntesis es $1/(2\sigma^2)$. El rol de la constante c es lograr que $\int p(x)\, dx = 1$.

Las distribuciones normales son simétricas alrededor de su punto medio $x = \mu$. Entonces no son apropiadas para las variables x que nunca son negativas. En este caso, una distribución log-normal puede ser más adecuada.

La distribución acumulada es la integral de $-\infty$ a x de $p(x)$. Cuando $p(x)$ es normal estándar ($\mu = 0$ y $\sigma^2 = 1$), su integral desde $-\infty$ a x a menudo se denota $\Phi(x)$:

$$\textbf{Distribución acumulada} \qquad \boldsymbol{\Phi(x)} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2}\, dt \tag{16}$$

Esto está estrechamente vinculado a la bien tabulada "función error" $\operatorname{erf}(x)$:

$$\textbf{Función error} \quad \boldsymbol{\operatorname{erf}(x)} = \frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-s^2}\, ds \tag{17}$$

Podemos cambiar esa variable s por $t/\sqrt{2}$, y sumar $\frac{1}{2}$ a la integral de $-\infty$ a 0. Así

$$\text{Integral de } p(x) = \boldsymbol{\Phi(x)} = \frac{\mathbf{1}}{\mathbf{2}}\left[\mathbf{1} + \operatorname{erf}\left(\frac{\boldsymbol{x}}{\sqrt{\mathbf{2}}}\right)\right] = \text{función error escalada y trasladada}$$

Para la distribución normal $\mathbf{N}(\mu, \sigma^2)$ solo cambiamos x por $(x - \mu)/\sigma$.

Las siguientes páginas describen dos distribuciones de probabilidad adicionales, que son importantes de por sí y que se obtienen directamente a partir de esta distribución normal univariada $p(x)$:

1. Cuando x tiene una distribución normal, la exponencial e^x tiene una **distribución log-normal**.
2. Si x tiene distribución $\mathbf{N}(0, 1)$ estándar, entonces x^2 tiene una **distribución chi cuadrado**.

Si $x_1, \ldots, x_n$ son normales independientes con media cero y varianza 1, entonces

3. Su suma $x_1 + \cdots + x_n$ tiene una **distribución normal** con media cero y varianza $\boldsymbol{n}$
4. $x_1^2 + \cdots + x_n^2$ tiene la **distribución chi cuadrado** $\boldsymbol{\chi_n^2}$ con n grados de libertad.

Los cocientes de las sumas de cuadrados independientes tienen **distribución** $\boldsymbol{F}$ (no estudiaremos aquí).

Lo que más queremos es permitir variables aleatorias que no sean independientes. En esta situación tendremos covarianzas además de varianzas. Los valores medios son $\mu_1, \ldots, \mu_n$:

Varianzas	$=$ **valores esperados de** $\boldsymbol{(x_i - \mu_i)^2}$
Covarianzas	$=$ **valores esperados de** $\boldsymbol{(x_i - \mu_i)(x_j - \mu_j)}$

Esos números están en la matriz de varianza-covarianza C de la Sección V.5. Cuando las variables $x_1, \ldots, x_n$ son independientes, las covarianzas son cero y C es diagonal. Cuando las variables no son independientes, como las acciones y los bonos, C es simétrica y definida positiva (en casos extremos, semidefinida). **Así la distribución conjunta de variables gaussianas $\boldsymbol{x_1, x_2, \ldots, x_n}$ es "gaussiana multivariada"** :

$$\boldsymbol{p(x) = p(x_1, \ldots, x_n)} = \frac{\mathbf{1}}{\boldsymbol{(\sqrt{2\pi})^n \sqrt{\det C}}}\, \boldsymbol{e^{-\frac{1}{2}(x-\mu)^{\mathrm{T}} C^{-1}(x-\mu)}} \tag{18}$$

Para $n = 1$, la matriz de covarianza de 1 por 1 es $C = [\sigma^2]$ y $p(x)$ es la gaussiana usual.

5. Distribución log-normal

La distribución de x es *log-normal* cuando la distribución de $y = \log x$ es normal. Esto requiere $\boldsymbol{x > 0}$. La distribución log-normal solo se admite si la variable aleatoria es positiva.

La distribución normal aparece (teorema del límite central) cuando se promedian muchas muestras independientes de una variable aleatoria y. Centramos y reescalamos por $(x - \mu)$, para trabajar con la normal estándar $\mathbf{N}(0, 1)$ con $\mu = 0$ y $\sigma^2 = 1$. De manera similar, la distribución log-normal aparece cuando tomamos el *producto* de muchos valores de muestra positivos. La media aritmética para la normal es como la media geométrica para la log-normal.

Para ver la fórmula de $p(x)$, comencemos con una distribución normal para $y = \log x$. La probabilidad total debe ser 1. Cambiamos las variables y recordamos que $dy = dx/x$. Esto trae un factor $1/x$ a la distribución log-normal $p(x)$:

$$1 = \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}}\, e^{-(y-\mu)^2/2\sigma^2}\, dy = \int_0^{\infty} \frac{1}{\sigma\sqrt{2\pi}\,\boldsymbol{x}}\, e^{-(\log x-\mu)^2/2\sigma^2}\, dx = \int_0^{\infty} p(x)\, dx. \qquad (19)$$

Las aplicaciones de la distribución log-normal siempre requieren que $x > 0$.

6. Distribución Chi cuadrado

Comencemos con la siguiente pregunta : si x tiene una distribución normal estándar con $\mu = 0$ y $\sigma^2 = 1$, ¿ cuál es la **distribución de probabilididad de $\boldsymbol{s = x^2}$** ? s tiene distribución χ_1^2, donde la letra griega *chi* y el subíndice 1 nos dicen que estamos elevando al cuadrado una variable normal estándar x. Claramente $s = x^2 \geq 0$.

> Necesitamos la probabilidad de que s se encuentre entre y e $y + dy$. Esto ocurre:
> o bien cuando $\sqrt{s}$ está entre $\sqrt{y}$ y $\sqrt{y + dy}$ o cuando $\sqrt{s}$ se encuentra entre $-\sqrt{y + dy}$ y $-\sqrt{y}$. Esos son igualmente probables ya que la normal estándar es simétrica alrededor del cero. Y $\sqrt{y + dy} = \sqrt{y} + dy/2\sqrt{y} +$ términos en $(dy)^2$:

$$\textbf{Prob}\,\{y < s < y + dy\} = \mathbf{2}\,\textbf{Prob}\left\{\sqrt{y} < \sqrt{s} < \sqrt{y} + \frac{dy}{2\sqrt{y}}\right\} = \frac{2}{\sqrt{2\pi}}\, e^{-(\sqrt{y})^2/2}\frac{dy}{2\sqrt{y}}$$

Esto responde nuestra primera pregunta. **La distribución de probabilidades de $\boldsymbol{s = y^2}$ es $\boldsymbol{p_1(s)}$** :

$$\textbf{distribución } \boldsymbol{\chi_1^2} \qquad \boldsymbol{p_1(s) = \frac{1}{\sqrt{2\pi s}}\, e^{-s/2}, \quad s > 0} \qquad (20)$$

Observación : mencionamos que la distribución acumulada de $s = x^2$ se conecta directamente con la distribución acumulada de la normal estándar $\boldsymbol{\Phi}(x)$ de la ecuación (16) :

$$\text{Prob}\{s < y\} = \text{Prob}\{-\sqrt{y} < x < \sqrt{y}\} = \Phi(\sqrt{y}) - (1 - \Phi(\sqrt{y})).$$

Por definición, la derivada en $y = s$ es la distribución de probabilidad χ_1^2, de acuerdo con (20) :

$$\begin{array}{l}\text{Nuevo enfoque}\\ \text{Misma fórmula}\end{array} \qquad p_1(s) = \frac{d}{dy}\left[2\Phi(\sqrt{y}) - 1\right] = \frac{1}{\sqrt{s}}\frac{1}{\sqrt{2\pi}}\, e^{-s/2}.$$

Nos enfocamos ahora en la **suma de dos cuadrados. Esto es** $\boldsymbol{s_2 = x_1^2 + x_2^2}$. Las variables normales estándares x_1 y x_2 son independientes. Tenemos que mirar todas las combinaciones $x_1^2 = s$ y $x_2^2 = s_2 - s$ que suman s_2. La distribución de probabilidad p_2 para s_2 es la integral sobre todas esas combinaciones :

$$p_2(s_2) = \int_0^{s_2} p_1(s)\, p_1(s_2 - s)\, ds = \text{“convolución de } p_1 \text{ con } p_1\text{”}. \tag{21}$$

Así la fórmula (20) para $p_1(s)$ nos lleva a (22) para $p_2(s_2)$, ¡es una distribución exponencial!

$$\textbf{Chi cuadrado con } \boldsymbol{n = 2} \textbf{ es exponencial : } \boldsymbol{s_2 = x_1^2 + x_2^2} \quad \boldsymbol{p_2(s_2) = \frac{1}{2} e^{-s_2/2}} \tag{22}$$

Esta convolución da un enfoque útil a χ_n^2 para todo $n = 3, 4, 5, \ldots$ La variable s_n es la suma de los cuadrados de $\boldsymbol{n}$ **normales estándares independientes**. Podemos considerar s_n como $s_1 +$ $s_{n-1} =$ un cuadrado más $n - 1$ cuadrados. Luego usamos la misma idea que en la ecuación (21) :

$$p_n(s_n) = \int_0^{s_n} p_1(s)\, p_{n-1}(s_n - s)\, ds = \text{“convolución de } p_1 \text{ con } p_{n-1}\text{”}$$

Esta integral p_n es de la forma $C s_n^{(n-2)/2}\, e^{-s_n/2}$. El número C es para que la probabilidad total $\int p_n ds_n$ sea igual a 1. La integración por partes reducirá sucesivamente el exponente $(n-2)/2$ hasta que alcancemos $-1/2$ o 0. Esos son los dos casos que hemos completado (para $n = 1$ y $n = 2$). Entonces sabemos cómo encontrar C, y tenemos la distribución de probabilidad χ_n^2 para $s_n = x_1^2 + \cdots + x_n^2$:

$$\boxed{\begin{array}{ll} \boldsymbol{s_n = \chi_n^2 = \text{suma de cuadrados de } n \text{ normales estándares}} & \boldsymbol{p_n(s_n) = C s_n^{(n-2)/2} e^{-s_n/2}} \\ \textbf{La integral de } \boldsymbol{p_n} \textbf{ debe ser 1} & \boldsymbol{C = \dfrac{1}{2^{n/2}\Gamma(n/2)} = \dfrac{1}{2^{n/2}\left(\frac{n}{2} - 1\right)!}} \end{array}} \tag{23}$$

La función gamma $\Gamma(n) = (n-1)\Gamma(n-1)$ es $(n-1)!$ y para $n = \frac{1}{2}$ es $\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}$.

Aplicación típica de χ^2

Fabricamos algo. Lo testeamos. Tenemos valores muestrales $x_1, \ldots, x_n$ de su espesor. Calculamos el espesor promedio $\overline{x}$ y la varianza muestral S^2 :

$$\overline{x} = \frac{1}{n}\sum_1^n x_i \quad \text{y} \quad S^2 = \frac{1}{n-1}\sum_1^n (x_i - \overline{x})^2.$$

S^2 es una suma de cuadrados con $n - 1$ grados de libertad. Un grado de libertad se usó en la media $\overline{x}$. Para $n = 1$, $\overline{x}$ es x_1 y $S = 0$. Para $n = 2$, $\overline{x}$ es $\frac{1}{2}(x_1 + x_2)$ y $S^2 = \frac{1}{2}(x_1 - x_2)^2$. S^2 tiene la distribución de probabilidad p_{n-1}, que es una χ_{n-1}^2 dada por (23).

Problemas de la Sección V.2

1 Si $p_1(x) = \dfrac{1}{\sqrt{2\pi}\,\sigma_1} e^{-x^2/2\sigma_1^2}$ y $p_2(x) = \dfrac{1}{\sqrt{2\pi}\,\sigma_2} e^{-x^2/2\sigma_2^2}$ muestre que existe una constante c tal que cp_1p_2 es una densidad de una distribución normal con media = cero y varianza $= \sigma^2 = \sigma_1^2\sigma_2^2/(\sigma_1^2 + \sigma_2^2)$.

El producto de gaussianas es gaussiano. Si p_1 y p_2 son densidades de variables aleatorias gaussianas, con medias m_1 y m_2, entonces para alguna constante c, resulta que cp_1p_2 será la densidad de una variable aleatoria gaussiana con media $(m_1\sigma_2^2 + m_2\sigma_1^2)/(\sigma_1^2 + \sigma_2^2)$.

2 Importante : la **convolución** de esas funciones de densidad gaussianas $p_1(x)$ y $p_2(x)$ también es una densidad gaussiana :

$$(\boldsymbol{p_1} * \boldsymbol{p_2})(x) = \int_{-\infty}^{\infty} p_1(t)\, p_2(x-t)\, dt \;=\; \frac{1}{\sqrt{2\pi(\sigma_1^2+\sigma_2^2)}}\, e^{-x^2/2(\sigma_1^2+\sigma_2^2)}$$

Una linda demostración usa transformadas de Fourier F y el teorema de convolución :

$$F[p_1(x) * p_2(x)] = F(p_1(x))\;\; F(p_2(x)).$$

Esto es así porque las transformadas $F(p_i(x))$ son múltiplos de $\exp(-\sigma_i^2 k^2/2)$. Multiplicar esas transformadas da un producto como en el Problema 1. *Ese producto contiene* $\sigma_1^2 + \sigma_2^2$. Así la inversa de la transformada de Fourier produce a $p_1 * p_2$ como otra densidad gaussiana.

Pregunta ¿ Cuál es la varianza σ_n^2 para la convolución de n densidades gaussianas idénticas $\mathrm{N}(0, \sigma^2)$?

3 Verifique que la convolución $P(x) = \displaystyle\int p(t)\, p(x-t)\, dt$ verifica $\displaystyle\int P(x)\, dx = 1$:

$$\int_{x=-\infty}^{\infty} P(x)\, dx = \int_x \int_t p(t)\, p(x-t)\, dt = \int_t \int_x p(t)\, p(x-t)\, dt = \underline{\quad\quad}.$$

4 Explique por qué la distribución de probabilidad $P(x)$ para la suma de dos variables aleatorias es la convolución $P = p_1 * p_2$ de sus distribuciones de probabilidad separadas. Un ejemplo es tirar dos dados y sumar los resultados :

$$\text{Probabilidades de la suma} \quad \left(\frac{1}{6},\frac{1}{6},\frac{1}{6},\frac{1}{6},\frac{1}{6},\frac{1}{6}\right) * \left(\frac{1}{6},\frac{1}{6},\frac{1}{6},\frac{1}{6},\frac{1}{6},\frac{1}{6}\right) =$$

$$\text{Probabilidades de 2 a 12} \quad \left(\frac{1}{36},\frac{2}{36},\frac{3}{36},\frac{4}{36},\frac{5}{36},\frac{6}{36},\frac{5}{36},\frac{4}{36},\frac{3}{36},\frac{2}{36},\frac{1}{36}\right)$$

V.3. Momentos, cumulantes y desigualdades de la Estadística

Supongamos que conocemos el valor promedio (la media $\overline{X} = \mathrm{E}[X]$) de una variable aleatoria X. Lo que queremos saber es algo sobre sus probabilidades: la probabilidad de que X sea mayor o igual a a. Un corte más grande a significa que la probabilidad será menor.

Markov encontró una cota simple $\overline{X}/a$ para la probabilidad de que $X \geq a$, para toda variable aleatoria *no negativa* X. A modo de ejemplo podemos elegir $a = 2\overline{X}$. Luego Markov dice: *una muestra aleatoria será mayor o igual a* $2\overline{X}$ *con probabilidad no mayor a* $\frac{1}{2}$.

La desigualdad de Markov supone que $X \geq 0$: ninguna muestra es negativa

Así la probabilidad de $X(s) \geq a$ a lo sumo $\dfrac{\mathbf{E}[X]}{a} = \dfrac{\textbf{media de } X}{a} = \dfrac{\overline{X}}{a}$

Un ejemplo mostrará por qué la desigualdad de Markov es cierta (y lo que esto significa). Supongamos que los números $0, 1, 2, \ldots$ (*todos no negativos*) aparecen con probabilidades $p_0, p_1, p_2, \ldots$ Y supongamos que esta distribución tiene media $\mathrm{E}[X] = \overline{X} = 1$:

$$\textbf{Valor medio} \qquad \overline{X} = 0p_0 + 1p_1 + 2p_2 + 3p_3 + 4p_4 + 5p_5 + \cdots = 1. \tag{1}$$

Así la desigualdad de Markov con $a = 3$ dice que la probabilidad de que $X \geq 3$ a lo sumo $\frac{1}{3}$:

$$\textbf{Markov} \qquad \boldsymbol{p_3 + p_4 + p_5 + \cdots \leq \frac{1}{3}}. \tag{2}$$

Prueba de Markov: escribir la Ecuación (1) de una manera que revele más:

$$0p_0 + 1p_1 + 2p_2 + \mathbf{3(}\boldsymbol{p_3 + p_4 + p_5 + \cdots}\mathbf{)} + p_4 + 2p_5 + \cdots = 1. \tag{3}$$

Todo término en la Ecuación (3) es mayor o igual a cero. Por lo tanto el término en negrita no podrá ser mayor que 1:

$$\mathbf{3(}\boldsymbol{p_3 + p_4 + p_5 + \cdots}\mathbf{) \leq 1} \textbf{ que es la desigualdad de Markov (2).} \tag{4}$$

La Ecuación (3) nos dice aún más. **¿Cuándo podría $p_3 + p_4 + p_5 + \cdots$ ser igual a $\frac{1}{3}$?** Ahora el término en negrita en la Ecuación (3) es igual a 1, entonces todos los demás términos deben ser cero:

$$p_1 = 0 \qquad 2p_2 = 0 \qquad p_4 = 0 \qquad 2p_5 = 0 \ldots$$

Esto deja solo a p_0 y p_3. Así fuerza a que $p_3 = \frac{1}{3}$ y $p_0 = \frac{2}{3}$ porque los p deben sumar 1.

Conclusiones Primero, la desigualdad de Markov es válida. Segundo, si hay igualdad, es decir, $\mathrm{Prob}\{x \geq a\}$ es igual a $\mathbf{E}[x]/a$, entonces todas las probabilidades son cero excepto :

$$\mathrm{Prob}\{x = a\} = \frac{\mathbf{E}[x]}{a} \quad \text{y} \quad \mathrm{Prob}\{x = 0\} = 1 - \frac{\mathbf{E}[x]}{a}.$$

Ahora daremos una prueba formal que permite tanto probabilidades continuas como discretas.

La prueba de la desigualdad de Markov tiene cuatro pasos fáciles.

$$\begin{aligned}\overline{\boldsymbol{X}} = \mathbf{E}[\boldsymbol{X}] &= \sum_{\text{todo } s} X(s) \text{ veces (Probabilidad de } X(s)) \\ &\geq \sum_{X(s) \geq a} X(s) \text{ veces (Probabilidad de } X(s)) \\ &\geq \sum_{X(s) \geq a} a \text{ veces (Probabilidad de } X(s)) \\ &= \boldsymbol{a} \textbf{ veces (Probabilidad de } \boldsymbol{X(s) \geq a})\end{aligned}$$

Dividiendo por a obtenemos la desigualdad de Markov: (**Probabilidad de** $\boldsymbol{X(s) \geq a}$) $\leq \overline{\boldsymbol{X}}/\boldsymbol{a}$.

La segunda desigualdad útil es la de Chebyshev. Esto se aplica a todas las variables aleatorias $X(s)$, no solo a las no negativas. Da una estimación para la probabilidad de eventos que están *lejos de la media* $\overline{X}$: estamos mirando la "cola" de la distribución de probabilidades. Para distribuciones con "colas pesadas" una desviación grande $|X(s) - \overline{X}|$ tiene una probabilidad mayor a la usual. La probabilidad de $|X - \overline{X}| \geq a$ decrecerá a medida que a crece.

La desigualdad de Chebyshev para cualquier distribución de probabilidad $\boldsymbol{X(s)}$

La probabilidad de $\boldsymbol{|X(s) - \overline{X}| \geq a}$ **es menor o igual a** $\dfrac{\boldsymbol{\sigma^2}}{\boldsymbol{a^2}}$

La prueba resulta de aplicar la desigualdad de Markov a la función no negativa $Y(s) = (X(s) - \overline{X})^2$. Por la definición de varianza, **la media de esa función** $\boldsymbol{Y}$ **es** $\boldsymbol{\sigma^2}$. Nos interesan los eventos con $|X(s) - \overline{X}| \geq a$. Elevemos al cuadrado ambos lados para obtener $Y(s) \geq a^2$. Luego usamos Markov:

$$\textbf{Chebyshev a partir de Markov} \qquad \text{Prob}\,(Y(s) \geq a^2) \;\leq\; \frac{\text{media de } Y}{a^2} = \frac{\sigma^2}{a^2}. \tag{5}$$

Esas son desigualdades sencillas y pueden mejorarse. A veces son suficientes para determinar que un algoritmo aleatorio en particular converge. Cuando no son suficientes, generalmente tenemos que ir más allá de la media y la varianza, a "momentos" y "cumulantes" de mayor orden. A menudo, la idea clave (como lo es en la desigualdad de Chernoff) es usar la **función generadora**, que conecta todos esos momentos.

Conectar una lista de números a una función es una idea poderosa en matemática. Una función real $f(x) = \Sigma\, a_n x^n$ o una función compleja $F(x) = \Sigma\, a_n e^{inx}$ contiene (de manera diferente) la información de todos los números a_n. Esos números podrían ser probabilidades, "momentos" o "cumulantes". En ese caso f será su *función generadora*.

Momentos y momentos centrales

La media y la varianza son números fundamentales tanto para las probabilidades p_i discretas como para las funciones de densidad de probabilidad $p(x)$. Pero la teoría básica de la estadística va más allá.

Para cada n, **los momentos son** $\boldsymbol{m_n = \mathrm{E}[x^n]}$. Por ahora conocemos m_0, m_1, m_2 :

Momento cero $= \mathbf{1}$	$\Sigma p_i = \mathbf{1}$ o $\int p(x)\,dx = \mathbf{1}$
Primer momento = media $= \mathrm{E}[x]$	$\Sigma i p_i = \boldsymbol{m}$ o $\int x p(x)\,dx = \boldsymbol{m}$
Segundo momento (alrededor de 0)	$\Sigma i^2 p_i$ o $\int x^2 p(x)\,dx = \boldsymbol{\sigma^2 + m^2} = \mathrm{E}[\boldsymbol{x^2}]$
Segundo momento central (alrededor de m)	$\Sigma (i-m)^2 p_i = \boldsymbol{\sigma^2}$ o $\int (x-m)^2 p(x)\,dx = \boldsymbol{\sigma^2}$

El n-ésimo momento m_n es $\Sigma i^n p_i$ o $\int x^n p(x)\,dx$. Pero los **momentos centrales** (alrededor de m) son más útiles. Son $\mu_n =$ valor esperado de $(x-m)^n$.

n-ésimo momento central $\boldsymbol{\mu_n = \Sigma (i-m)^n p_i}$ o $\boldsymbol{\int (x-m)^n p(x)\,dx}$	(6)
n-ésimo momento central normalizado $= \boldsymbol{\mu_n/\sigma^n}$	(7)

Toda distribución simétrica con $p_i = p_{-i}$ o $p(x) = p(-x)$ tendrá media cero (si existe). *Todos los momentos de orden impar (si existen) serán cero*, porque la función es impar.

Cuando $p(x)$ es *simétrica alrededor del valor de la media*, los momentos *centrales* impares $\mu_1, \mu_3, \ldots$ serán todos cero (si existen). [Mejor ejemplo = distribución normal.] El momento central tercero normalizado $\boldsymbol{\gamma = \mu_3/\sigma^3}$ se denomina **asimetría** de la distribución.

Pregunta : ¿ Cuál es la asimetría de las probabilidades $p_0 = \mathbf{1-p}$ y $p_1 = \boldsymbol{p}$ de una Bernoulli de los lanzamientos de moneda ?

Respuesta : Primero, la media es p. La varianza es $\mu_2 = \sigma^2 = (1-p)p^2 + p(1-p)^2 = \boldsymbol{p(1-p)}$. El tercer momento central es μ_3, y depende de las distancias a la media p :

$$\boldsymbol{\mu_3} = (1-p)(0-p)^3 + p(1-p)^3 = p(1-p)(1-2p) \qquad \textbf{asimetría } \boldsymbol{\gamma} = \frac{\mu_3}{\sigma^3} = \frac{1-2p}{\sqrt{p(1-p)}}.$$

Para la probabilidad, el centro es la media m. Una distribución con cola pesada tendrá un valor de μ_4 grande. Generalmente usamos el cuarto momento de una distribución normal (es $3\sigma^4$) como base para la comparación. Así a la **curtosis** de una distribución la denominamos kappa :

$$\textbf{Curtosis} \quad \boldsymbol{\kappa} = \frac{\boldsymbol{\mu_4 - 3\sigma^4}}{\boldsymbol{\sigma^4}} = \frac{\boldsymbol{\mu_4}}{\boldsymbol{\sigma^4}} \boldsymbol{- 3}. \tag{8}$$

Funciones generadoras y cumulantes

Cuatro funciones clave se generan a partir de las probabilidades, los momentos y los cumulantes (cuando estos existen). Comenzamos con una variable aleatoria discreta X. El evento $X = n$ tiene probabilidad p_n. Esos números p_n nos llevan a las primeras tres funciones. Luego $\boldsymbol{K(t) = \log M(t)}$.

Función generadora de probabilidades	$G(z) = \sum_0^\infty p_n z^n$	(9)
Función característica	$\phi(t) = \sum_0^\infty p_n e^{itn}$	(10)
Función generadora de momentos	$M(t) = \sum_0^\infty m_n \dfrac{t^n}{n!}$	(11)
Función generadora de cumulantes	$K(t) = \sum_0^\infty \kappa_n \dfrac{t^n}{n!}$	(12)

Todas las probabilidades p_n, momentos m_n y cumulantes κ_n pueden recuperarse de la n-ésima derivada de las funciones G, M, K en $z = 0$ y $t = 0$:

$$\boldsymbol{p_n = \frac{1}{n!}\frac{d^n G}{dz^n}(0)} \qquad \boldsymbol{m_n = \frac{d^n M}{dt^n}(0)} \qquad \boldsymbol{\kappa_n = \frac{d^n K}{dt^n}(0)} \tag{13}$$

Y hay una manera muy reveladora de conectar, con esperanzas, las cuatro funciones generadoras:

$$\boxed{\boldsymbol{G(z) = \mathbf{E}\left[z^X\right] \quad \phi(t) = \mathbf{E}\left[e^{itX}\right] \quad M(t) = \mathbf{E}\left[e^{tX}\right] \quad K(t) = \log \mathbf{E}\left[e^{tX}\right]}} \tag{14}$$

Por supuesto que hay un propósito detrás de estas cuatro funciones. Cada una tiene un conjunto infinito de coeficientes. El punto clave de los cumulantes es que nos llevan a esta propiedad de $K(t) = \log M(t)$:

$$\boldsymbol{K_{X+Y}(t) = K_X(t) + K_Y(t)} \text{ **para variables aleatorias independientes } } \boldsymbol{X} \text{ **e** } \boldsymbol{Y}.$$

Ejemplos Un lanzamiento de moneda nos da una **distribución de Bernoulli** con probabilidades $p_0 = 1 - p$ y $p_1 = p$. Así $p = \mathrm{E}[x] = \mathrm{E}[x^2] = \mathrm{E}[x^3] = \cdots$ y $\boldsymbol{M(t) = 1 - p + pe^t}$.

Función generadora de acumulantes $\boldsymbol{K(t) = \log(1 - p + pe^t)}$.

El primer cumulante es $\boldsymbol{\kappa_1 = \dfrac{dK}{dt}} = \left[\dfrac{pe^t}{1 - p + pe^t}\right]_{t=0} = \boldsymbol{p}$.

Para la **distribución binomial** (N lanzamientos independientes de una moneda) se multiplica cada cumulante por N.

Para la **distribución de Poisson** $p_n = e^{-\lambda}\lambda^n/n!$ la función $K(t)$ es $\lambda(e^t - 1)$. Todo $\kappa_n = \lambda$.

Funciones generadoras para distribuciones continuas

No podemos dejar de lado la distribución normal. Para una distribución continua, las funciones generadoras tienen integrales que involucran $p(x)$ en lugar de sumas que involucran p_n :

$$\phi(t) = \int_{-\infty}^{\infty} p(x)\, \boldsymbol{e^{itx}}\, dx \qquad M(t) = \int_{-\infty}^{\infty} p(x)\, \boldsymbol{e^{tx}}\, dx \qquad K(t) = \log M(t). \tag{15}$$

La función generadora de momentos para una distribución normal (media μ, varianza σ^2) es $M(t) = e^{\mu t} e^{\sigma^2 t^2/2}$. La función generadora de cumulantes es su logaritmo $\boldsymbol{K(t) = \mu t + \sigma^2 t^2/2}$. Así la distribución normal es extremadamente excepcional con solo dos cumulantes no nulos:

Distribución normal	$\boldsymbol{\kappa_1 = \mu}$	$\boldsymbol{\kappa_2 = \sigma^2}$	$\boldsymbol{\kappa_3 = \kappa_4 = \ldots = 0}$

Datos clave sobre 3 cumulantes $\boldsymbol{\kappa_1}$ **= media** $\boldsymbol{\kappa_2}$ **= varianza** $\boldsymbol{\kappa_3}$ **= tercer momento central**

Debido a que se pueden sumar los cumulantes de procesos independientes, estos aparecen frecuentemente en combinatoria, estadística y física. Los de orden superior son más complicados que κ_3.

El teorema del límite central

En pocas líneas podemos justificar el gran teorema límite de la probabilidad. Se trata de los promedios estandarizados $Z_n = \sum(X_k - m)/\sigma\sqrt{N}$ de N muestras independientes $X_1, \ldots, X_N$ con media m y varianza σ^2. El teorema del límite central establece : **la distribución de $\boldsymbol{Z_n}$ se aproxima a la distribución normal estándar (media cero, varianza 1) cuando $\boldsymbol{N \to \infty}$.**

La prueba usa la función característica de la variable estandarizada $Y = (X - m)/\sigma$:

$$\mathrm{E}\left[e^{itY}\right] = \mathrm{E}\left[1 + itY - \frac{1}{2}t^2Y^2 + O(t^3)\right] = \mathbf{1} + \mathbf{0} - \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{t^2} + O(t^3) \tag{16}$$

Sin duda $Z_N = (Y_1 + Y_2 + \cdots + Y_N)/\sqrt{N}$. Así su función característica es un producto de N funciones características idénticas de $Y/\sqrt{N}$ con t fijo. El último problemas de esta sección da un ejemplo.

$$\left(\mathrm{E}\left[e^{itY/\sqrt{N}}\right]\right)^{\boldsymbol{N}} = \left[1 - \frac{1}{2}\left(\frac{t}{\sqrt{N}}\right)^2 + O\left(\frac{t}{\sqrt{N}}\right)^3\right]^{\boldsymbol{N}} \to \boldsymbol{e^{-t^2/2}} \quad \text{cuando} \quad N \to \infty. \tag{17}$$

Ese límite $\boldsymbol{e^{-t^2/2}}$ es la función característica de una distribución normal estándar $\mathrm{N}(0, 1)$.

Desigualdad de Chernoff para sumas

El valor medio de una suma $X = X_1 + \cdots + X_n$ es siempre $\overline{X} = \overline{X}_1 + \cdots + \overline{X}_n$. Si esas variables son *independientes*, entonces también la varianza σ^2 de X es $\sigma^2 = \sigma_1^2 + \cdots + \sigma_n^2$.

La desigualdad de Chebyshev del principio de esta sección, da una cota para la probabilidad de que las muestras X_i estén lejos de sus medias $\overline{X}_i$. También se aplica a sus sumas X y $\overline{X}$.

$$\boxed{\textbf{Chebyshev para una suma} \quad \textbf{Prob}\,\{|X-\overline{X}|\geq a\} \leq \frac{\sigma_1^2+\cdots+\sigma_n^2}{a^2}} \tag{18}$$

La desigualdad es justa para $n=1$. ¿Puede mejorarse la desigualdad para una suma? Yo habría dicho que *no*. Chernoff dice que *sí*. Aquí hay una cuestión muy sutil. Chernoff supuso no solo la independencia de cada par X_i y X_j, sino *independencia conjunta* de los X_i:

$$\text{Multiplicar probabilidades} \qquad p(x_1,\ldots,x_n) = p_1(x_1)\cdots p_n(x_n). \tag{19}$$

La independencia de a pares es suficiente para que la matriz de covarianza sea diagonal. La Ecuación (19) dice más. Y lleva a cotas para una suma (20) mucho más precisas que la desigualdad de Chebyshev (18).

El dato clave sobre la desigualdad de Chernoff es su *cota exponencial*. Sumas X que se encuentran lejos de su media $\overline{X}$ son exponencialmente improbables. Eso se debe a que una suma excepcional $X = X_1+\cdots+X_n$ suele necesitar que *varios* X_i estén lejos de sus medias $\overline{X}_i$.

Un ejemplo es la cantidad de caras después de lanzar n monedas. Así $\overline{X}_i = p_i$ para cada moneda. La cantidad total X de caras tendrá un valor medio $\overline{X} = p_1+\cdots+p_n$.

$$\boxed{\begin{array}{ll} \textbf{Chernoff superior} & \textbf{Prob}\,\{X\geq(1+\delta)\overline{X}\} \leq e^{-\overline{X}\delta^2/(2+\delta)} \\ \textbf{Chernoff inferior} & \textbf{Prob}\,\{X\leq(1-\delta)\overline{X}\} \leq e^{-\overline{X}\delta^2/2} \end{array}} \tag{20}$$

Michel Goemans, en sus apuntes online para la clase 18.310 del MIT, señala cuán fuerte es esto. Supongamos que tenemos n lanzamientos de una moneda justa. Así $\overline{X} = n/2$ (mitad cara, mitad cruz). Ahora tomemos $\delta^2 = (4\log n)/n$ un pequeño, y comparemos las cotas de Chernoff (20) para δ y 2δ.

> La probabilidad de que $X \leq (1-2\delta)\overline{X}$ es mucho más pequeña que la probabilidad de que $X \leq (1-\delta)\overline{X}$. Al duplicar δ, el δ^2 en el exponente de Chernoff cambia a $4\delta^2$. La probabilidad baja de $1/n$ a $1/n^4$:
>
> $$\begin{aligned} \overline{X}\delta^2/2 &= \log n \quad \text{da una cota } \boldsymbol{e^{-\log n} = 1/n} \\ \overline{X}(2\delta)^2/2 &= 4\log n \quad \text{da una cota } \boldsymbol{e^{-4\log n} = 1/n^4} \end{aligned}$$

Chebyshev hubiera obtenido $\boldsymbol{1/4n}$ y Chernoff, $\boldsymbol{1/n^4}$: ¡una diferencia exponencial! Podemos indicar el paso clave para demostrar las cotas de Chernoff. Primero centrar X para que $\overline{X} = 0$:

$$\begin{array}{lll} \textbf{Chebyshev usual} & \textbf{Prob}\,\{|X|\geq a\} = \text{Prob}\,\{X^2\geq a^2\} & \leq \mathbf{E}\,[X^2]/a^2 \\ \textbf{Chernoff superior} & \textbf{Prob}\,\{X\geq a\} = \text{Prob}\,\{e^{sX}\geq e^{sa}\} & \leq \mathbf{E}\left[e^{sX}\right]/e^{sa} \\ \textbf{Chernoff inferior} & \textbf{Prob}\,\{X\leq a\} = \text{Prob}\,\{e^{-sX}\geq e^{-sa}\} & \leq \mathbf{E}\left[e^{-sX}\right]/e^{-sa} \end{array}$$

Necesitamos ambas exponenciales (más y menos). Especialmente necesitamos la función generadora de momentos. Esto nos dice que $M(s) = \mathrm{E}\,[e^{sX}]$ y $M(-s) = \mathrm{E}\,[e^{-sX}]$. Así Chernoff es consecuencia de una elección cuidadosa de s. Tiene muchas aplicaciones en el álgebra lineal aleatorizada.

Para trabajar con muestras no independientes, necesitamos tanto **covarianzas** como varianzas. También necesitamos las desigualdades de Markov-Chebyshev-Chernoff para matrices. Es lo que viene a continuación.

Desigualdades de Markov y Chebyshev para matrices

El algoritmo del descenso por gradiente estocástico (Sección VI.5) es el algoritmo más aceptado para optimizar los pesos en una red neuronal. Estocástico significa *aleatorio*, y esos pesos se almacenan en matrices. Entonces, un paso esencial en la estadística del aprendizaje profundo consiste en desarrollar desigualdades para los *autovalores y la traza de matrices aleatorias.*

No haremos aquí una presentación exhaustiva sino solo los resultados básicos. Escribiremos $\boldsymbol{X \leq A}$ cuando $A - X$ sea semidefinida positiva : energía ≥ 0 y todos los autovalores ≥ 0. De otro modo $\boldsymbol{X \not\leq A}$. En este caso $A - X$ tiene al menos un autovalor negativo.

Desigualdad de Markov Sea $X \geq 0$ una matriz aleatoria semidefinida o definida positiva con media $\mathrm{E}[X] = \overline{X}$. Si A es cualquier matriz definida positiva, entonces

$$\textbf{Prob}\,\{\boldsymbol{X \not\leq A}\} = \textbf{Prob}\,\{A - X \text{ no es semidefinida positiva }\} \leq \textbf{Traza de } \boldsymbol{\overline{X} A^{-1}}. \tag{21}$$

Si X y A son escalares x y a, comparemos (21) con la desigualdad de Markov $\text{Prob}\,\{x \geq a\} \leq \overline{x}/a$.

Demostración Si $A^{1/2}$ es la raíz cuadrada definida positiva de A, entonces el paso clave es :

$$(\text{Traza de } A^{-1/2} X A^{-1/2}) > 1 \text{ si } X \not\leq A. \tag{22}$$

Cuando $A - X$ no es semidefinida positiva debe haber un vector $\boldsymbol{v}$ con energía negativa $\boldsymbol{v}^{\mathrm{T}}(A - X)\boldsymbol{v} < 0$. Sea $\boldsymbol{w} = A^{1/2}\boldsymbol{v}$ de manera que $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} < \boldsymbol{w}^{\mathrm{T}} A^{-1/2} X A^{-1/2}\boldsymbol{w}$. Así el autovalor más grande de $A^{-1/2} X A^{-1/2}$ es $\lambda_{\text{máx}} > 1$:

$$\textbf{Cociente de Rayleigh} \qquad \lambda_{\text{máx}} = \text{máx}\, \frac{\boldsymbol{y}^{\mathrm{T}} A^{-1/2} X A^{-1/2} \boldsymbol{y}}{\boldsymbol{y}^{\mathrm{T}} \boldsymbol{y}} > 1$$

Ningún autovalor de $A^{-1/2} X A^{-1/2}$ es negativo, entonces su traza es más grande que 1. Esto es (22). Así, tomando esperanzas a ambos lados de (22) obtenemos la desigualdad de Markov (21) :

$$\text{Prob}\,\{X \not\leq A\} \leq \text{E[traza}\,(A^{-1/2} X A^{-1/2})] = \text{traza}\,(A^{-1/2} \overline{X} A^{-1/2}) = \text{traza}\,(\overline{X} A^{-1}).$$

Ahora recurrimos a la desigualdad de Chebyshev $\text{Prob}\,\{|X - \overline{X}| \geq a\} \leq \sigma^2/a^2$. Para una matriz simétrica $A = Q \Lambda Q^{\mathrm{T}}$, usaremos un hecho acerca de su valor absoluto $|A| = Q|\Lambda|Q^{\mathrm{T}}$:

Si $\boldsymbol{A^2 - B^2}$ *es semidefinida positiva, entonces* $\boldsymbol{|A| - |B|}$ *también es semidefinida positiva.*

La prueba no es muy simple. La afirmación opuesta no es cierta. Guardamos ejemplos para la sección de problemas y usamos esta información ahora :

Desigualdad de Chebyshev para una matriz aleatoria X con media 0
Si A es definida positiva, entonces $\textbf{Prob}\,\{|X| \not\leq A\} < \textbf{traza}\,(\mathrm{E}[X^2] A^{-2})$. (23)

Si $A - |X|$ no es semidefinida positiva, entonces $A^2 - X^2$ no es semidefinida positiva (como se indicó arriba) :

$$\text{Prob}\,\{|X| \not\leq A\} \leq \text{Prob}\,\{X^2 \not\leq A^2\} < \text{traza}\,(\mathrm{E}[X^2] A^{-2}).$$

Ese último paso fue la desigualdad de Markov aplicada a la matriz semidefinida positiva X^2.

En Wikipedia hay una desigualdad de Chebyshev multidimensional para una variable aleatoria $\boldsymbol{x}$ en $\mathbf{R}^N$. Supongamos que tiene media $\mathrm{E}[\boldsymbol{x}] = \boldsymbol{m}$ y matriz de covarianza $V = \mathrm{E}[(\boldsymbol{x}-\boldsymbol{m})(\boldsymbol{x}-\boldsymbol{m})^{\mathrm{T}}]$:

Si V es definida positiva y $t > 0$ entonces $\mathbf{Prob}\{(\boldsymbol{x}-\boldsymbol{m})^{\mathrm{T}} V^{-1}(\boldsymbol{x}-\boldsymbol{m}) > t^2\} \leq N/t^2$.

Desigualdades matriciales de Chernoff

Las desigualdades de Chernoff lidian con una suma Y de variables aleatorias X_k (previamente eran escalares y ahora son matrices de n por n semidefinidas positivas o definidas positivas). Miramos el autovalor más pequeño y más grande de esa suma. El punto central de Chernoff es que para que una suma esté lejos de su valor medio, normalmente necesita que *varios términos* en la suma estén bastante lejos, y esa combinación de eventos inusuales es exponencialmente improbable.

Así obtenemos cotas exponencialmente pequeñas para las colas de la distribución (distancias a la media).

Chernoff matricial Supongamos que cada matriz X_k en $Y = \Sigma X_k$ tiene autovalores $0 \leq \lambda \leq C$. Sean $\mu_{\text{mín}}$ y $\mu_{\text{máx}}$ los autovalores extremos de la suma promedio $\overline{Y} = \Sigma \overline{X}_k$. Entonces

$$\mathbf{E}[\lambda_{\text{mín}}(Y)] \geq \left(1 - \frac{1}{e}\right) \mu_{\text{mín}} - C \log n \tag{24}$$

$$\mathbf{E}[\lambda_{\text{máx}}(Y)] \leq (e-1)\ \mu_{\text{máx}} + C \log n \tag{25}$$

Que los autovalores de la suma Y estén lejos de su media es exponencialmente improbable:

$$\mathbf{Prob}\{\lambda_{\text{mín}}(Y) \leq t\,\mu_{\text{mín}}\} \leq n\, e^{-(1-t)^2 \mu_{\min}/2C} \tag{26}$$

$$\mathbf{Prob}\{\lambda_{\text{máx}}(Y) \geq t\,\mu_{\text{máx}}\} \leq n \left(\frac{e}{t}\right)^{t\mu_{\max}/C} \text{ para } t \geq e \tag{27}$$

El libro de texto (*online*) de Joel Tropp contiene una excelente presentación de las desigualdades matriciales. Su libro ofrece estimaciones más ajustadas (24)–(27) que para las esperanzas y las cotas exponenciales.

Joel Tropp, *An Introduction to Matrix Concentration Inequalities*, arXiv :1501.01571.

Hay muchas más desigualdades que se desprenden de Chernoff. Nos detenemos en una aplicación de la desigualdad (26) a la probabilidad de que un grafo aleatorio con aristas, sea conexo.

Grafos aleatorios de Erdős-Renyi

Esta es una buena oportunidad para definir grafos aleatorios. Comenzamos con n nodos. **Cada arista está presente con probabilidad p.** Entonces la pregunta es: **¿para qué valor de p es probable que el grafo sea conexo?**

La matriz de adyacencia de n por n tiene $M_{jk} = M_{kj} = 1$ si los nodos j y k están conectados. La variable aleatoria x_{jk} para esa arista bidireccional es 0 o 1, con probabilidad $1-p$ y p:

$$\textbf{Matriz de adyacencias} \qquad \boldsymbol{M} = \text{ suma de matrices aleatorias } = \sum_{j<k} x_{jk}(E_{jk} + E_{kj}). \tag{28}$$

E_{jk} es la matriz con un solo 1 en la posición (j,k). La matriz laplaciana es $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{M}$.

Suma de las filas menos M $$\boldsymbol{L} = \sum_{j<k} x_{jk}(E_{jj} + E_{kk} - E_{jk} - E_{kj}) \tag{29}$$

Por cada arista (es decir $x_{jk} = 1$), la matriz D de grados tiene dos unos de $E_{jj} + E_{kk}$. La matriz M de adyacencias tiene dos unos de $E_{jk} + E_{kj}$. Juntos esos cuatro elementos implican $\lambda = 2$ y 0. Así L es semidefinida positiva y a $\lambda = 0$ le corresponde el autovector $(1, \ldots, 1)$.

El segundo autovalor más pequeño de L será positivo cuando el grafo sea conexo. Si el grafo tiene dos partes, podemos numerar los nodos de una parte antes que los nodos de la otra parte. Así L tiene bloques separados correspondientes a las dos partes. Cada bloque tiene un autovector de todos unos, y L tiene $\lambda_1 = \lambda_2 = 0$ (dos autovalores cero con autovectores separados de unos).

Necesitamos una matriz aleatoria Y de tamaño $n-1$ cuyo mínimo autovalor sea $\boldsymbol{\lambda_1(Y) = \lambda_2(L)}$. Así un grafo conexo tendrá $\lambda_1(Y) > 0$. Una Y apropiada es ULU^{T}, donde las $n-1$ filas de U son vectores unitarios ortogonales que son perpendiculares al vector de solo unos. Y es como L, pero sin el autovalor cero y sin el autovector correspondiente de solo unos.

Ahora aplicamos el teorema de Chernoff matricial para descubrir cuándo $\lambda_{\text{mín}}(Y) > 0$ y así el grafo es conexo. Y es la suma de matrices aleatorias X_{jk} (para todo $j < k$):

$$\boldsymbol{Y} = ULU^{\mathrm{T}} = \sum x_{jk}\, U\,(E_{jj} + E_{kk} - E_{jk} - E_{kj})\, U^{\mathrm{T}} = \sum X_{jk} \tag{30}$$

Ya que $||U|| = ||U^{\mathrm{T}}|| = 1$, cada X_{jk} es semidefinida con autovalores ≤ 2 como se mostró antes.

Así $C = 2$ en el teorema de Chernoff. También necesitamos el autovalor más pequeño $\mu_{\text{mín}}$ de la matriz promedio $\overline{Y}$. El valor esperado de cada número aleatorio x_{jk} es p (la probabilidad de incluir esa arista en el grafo). El valor esperado de $Y = \sum X_{jk}$ es $p\,n\,I_{n-1}$:

$$\overline{\boldsymbol{Y}} = p\,U\left[\sum_{j<k}(E_{jj} + E_{kk} - E_{jk} - E_{kj})\right] U^{\mathrm{T}}$$

$$= p\,U[(n-1)I_n - (\mathbf{1}\mathbf{1}^{\mathrm{T}} - I_n)]U^{\mathrm{T}} = \boldsymbol{p\,n\,I_{n-1}} \tag{31}$$

Ese término $(n-1)I_n$ se obtuvo de sumar todas las matrices diagonales E_{jj} y E_{kk}. Las matrices no diagonales E_{jk} y E_{kj} suman $\mathbf{1}\mathbf{1}^{\mathbf{T}} - I_n$ = una matriz toda de unos y con ceros en la diagonal. $UU^{\mathrm{T}} = I_{n-1}$ produjo $p\,n\,I_{n-1}$, y hemos encontrado el autovalor más pequeño $\boldsymbol{\mu_{\text{mín}} = p\,n}$.

Ahora apliquemos la desigualdad (26) del teorema de Chernoff con $C = 2$:

$$\textbf{Prob}\,\{\lambda_2(L) \leq t\,p\,n\} = \textbf{Prob}\,\{\lambda_1(Y) \leq t\,p\,n\} \leq (n-1)\,e^{-(1-t)^2 pn/4}. \tag{32}$$

A medida que $t \to 0$, la cantidad crucial es $(n-1)e^{-pn/4}$. Esto es menor que 1 si su logaritmo es menor que cero.

$$\log(n-1) - \frac{1}{4}pn < 0 \quad \text{o} \quad p > \frac{4\log(n-1)}{n}. \tag{33}$$

Las aristas en el grafo aleatorio están incluidas con probabilibad p. *Si p es suficientemente grande para satisfacer* (33), *entonces el grafo es probablemente conexo.* Un argumento más preciso quitaría el factor 4. Eso produce el valor de corte óptimo de p para grafos aleatorios conexos.

Problemas de la Sección V.3

1 Encuentre la función generadora de probabilidades $G(z)$ para la distribución de Poisson $p_n = e^{-\lambda}\lambda^n/n!$

2 Las variables aleatorias independientes x e y verifican $p(x,y) = p(x)\,p(y)$. Derive la propiedad especial $K_{X+Y}(t) = K_X(t) + K_Y(t)$ de sus funciones generadoras de cumulantes.

3 Un lanzamiento de moneda justa da los resultados $X = 0$ y $X = 1$ con probabilidades $\frac{1}{2}$ y $\frac{1}{2}$. ¿Cuál es la probabilidad de que $\boldsymbol{X \geq 2\overline{X}}$? Muestre que la desigualdad de Markov da la probabilidad exacta $1/2$ en este caso.

4 Al lanzar dos dados ordinarios y sumar obtenemos X entre $X = 2$ and $X = 12$ (dos 1 o dos 6). El valor medio es 7. ¿Cuál es la probabilidad real p de que $X \geq 12$? Muestre que, para $a = 12$, el $\overline{X}/12$ de Markov sobrestima esa probabilidad p en este caso.

5 He aquí otra prueba de la desigualdad básica de Markov para una variable no negativa X.

Para $a > 0$, la variable aleatoria $Y = \begin{cases} 0 & \text{si } X \leq a \\ a & \text{si } X > a \end{cases}$ satisface $Y \leq X$. ¿Por qué?

Explique el paso final $\boldsymbol{a}$ **Prob** $[\boldsymbol{X \geq t}] = \mathbf{E}[\boldsymbol{Y}] \leq \mathbf{E}[\boldsymbol{X}]$ para obtener la cota de Markov $\mathrm{E}[X]/a$.

6 Muestre que el autovalor más grande de una $Y = Y^{\mathrm{T}}$ aleatoria es una función convexa de Y:

$$\boldsymbol{\lambda_{\text{máx}}(\overline{Y})} = \lambda_{\text{máx}}(p_1 Y_1 + \cdots + p_n Y_n) \leq p_1 \lambda_{\text{máx}}(Y_1) + \cdots + p_n \lambda_{\text{máx}}(Y_n) = \mathrm{E}[\boldsymbol{\lambda_{\text{máx}}(Y)}].$$

7 Muestre que $A - B$ es semidefinida positiva pero $A^2 - B^2$ no lo es:

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

8 Demuestre esta asombrosa identidad cuando las muestras aleatorias $0 < x_1 < x_2 < \cdots < x_n$ tienen probabilidades p_1 a p_n:

$$\textbf{media} = \mathbf{E}[x] = \sum_1^n p_i x_i = \int_{t=0}^{\infty} (\textbf{Probabilidad de que } \boldsymbol{x > t})\, dt.$$

Pista: esa probabilidad es $\sum p_i = 1$ para $t \leq x_1$ y luego es $1 - p_1$ para $t \leq x_2$.

9 Encuentre m, σ^2, y $\mathbf{E}[e^{itX}] = \cos t$ si $X = 1$ con $p_1 = \frac{1}{2}$ y $X = -1$ con $p_{-1} = \frac{1}{2}$.

¿Cuánto valen p_2, p_0, p_{-2} para la suma Z de dos muestras de X? Verifique que $\mathbf{E}[e^{itZ}] = (\mathbf{E}[e^{itX}])^2$.

Usando que $\left(1 - \frac{C}{N}\right)^N \to e^{-C}$, explique (17): $\left[\mathbf{E}\left(e^{itX/\sqrt{N}}\right)\right]^N \to e^{-t^2/2}$.

V.4. Matrices de covarianza y probabilidades conjuntas

Recurrimos al álgebra lineal cuando realizamos M experimentos diferentes a la vez. Podemos medir la edad, la altura y el peso ($M = 3$ medidas de N personas). Cada experimento tiene su propio valor medio. Entonces tenemos un vector $\boldsymbol{m} = (m_1, m_2, m_3)$ que contiene los M valores medios. Esos podrían ser *medias muestrales* de la edad, la altura y el peso. O m_1, m_2, m_3 podrían ser los *valores esperados* de edad, altura y peso basados en probabilidades conocidas.

Cuando miramos las varianzas tenemos que trabajar con matrices. Cada experimento tendrá una varianza muestral S_i^2 o esperada $\sigma_i^2 = \mathrm{E}\left[(x_i - m_i)^2\right]$ basada en el cuadrado de la distancia a su media. Esos M números $\sigma_1^2, \ldots, \sigma_M^2$ irán en la diagonal principal de la "matriz de varianza-covarianza". Hasta el momento no hemos logrado vincular los M experimentos paralelos. Estos miden diferentes variables aleatorias, pero los experimentos no son necesariamente independientes.

Si medimos la edad, la altura y el peso $(\boldsymbol{a}, \boldsymbol{h}, \boldsymbol{w})$ de niños, los resultados estarán fuertemente correlacionados. Los niños mayores suelen ser más altos y más pesados. Supongamos que conocemos las medias m_a, m_h, m_w. Así $\sigma_a^2, \sigma_h^2, \sigma_w^2$ son las varianzas de edad, altura y peso. **Los nuevos números son las covarianzas $\boldsymbol{\sigma_{ah}}$, que mide la relación entre la edad y la altura.**

$$\textbf{Covarianza} \quad \boldsymbol{\sigma_{ah}} = \mathbf{E}\left[(\textbf{edad} - \textbf{edad media})\ (\textbf{altura} - \textbf{altura media})\right]. \tag{1}$$

Observemos detenidamente esta definción. Para calcular σ_{ah}, no es suficiente conocer la probabilidad de cada edad y la probabilidad de cada altura. Debemos conocer la **probabilidad conjunta de cada par (edad y altura)**. Esto es así porque la edad está relacionada con la altura.

$\boldsymbol{p_{ah}}$ = probabilidad de que un niño aleatorio tenga edad $= \boldsymbol{a}$ y altura $= \boldsymbol{h}$: ambos al mismo tiempo

$\boldsymbol{p_{ij}}$ = **probabilidad de que el experimento 1 produzca $\boldsymbol{x_i}$ y el experimento 2 produzca $\boldsymbol{y_j}$**

Supongamos que el experimento 1 (edad) tiene media m_1 y el experimento 2 (altura) tiene media m_2. La covarianza en la ecuación (1) entre los experimentos 1 y 2 considera **todos los pares** de las edades x_i y las alturas y_j. Multiplicamos por la probabilidad conjunta p_{ij} de ese par.

Valor esperado de $(x - m_1)(y - m_2)$

$$\textbf{Covarianza} \quad \boldsymbol{\sigma_{12}} = \sum_{\textbf{todo } i,j} \sum \boldsymbol{p_{ij}(x_i - m_1)(y_j - m_2)} \tag{2}$$

Para comprender esta idea de "probabilidad conjunta p_{ij}" comenzamos con dos ejemplos sencillos.

Ejemplo 1 **Lanzar dos monedas por separado**. Si escribimos 1 si es cara y 0 si es cruz, los resultados posibles son $(1,1)$, $(1,0)$, $(0,1)$ o $(0,0)$. Esos cuatro resultados tienen probabilidad $\left(\frac{1}{2}\right)^2 = \frac{1}{4}$. Para experimentos independientes, multiplicamos las probabilidades:

$\boldsymbol{p_{ij}} = \textbf{Probabilidad de } \boldsymbol{(i,j)} = \textbf{(Probabilidad de } \boldsymbol{i}\textbf{) multiplicada por (Probabilidad de } \boldsymbol{j}\textbf{)}.$

Ejemplo 2 *Pegamos las monedas*, apuntando hacia el mismo lado. Las únicas posibilidades son $(1,1)$ y $(0,0)$, que tienen probabilidades $\frac{1}{2}$ y $\frac{1}{2}$. Las probabilidades p_{10} y p_{01} son cero. $(1,0)$ y $(0,1)$ no ocurrirán porque las monedas están pegadas : ambas salen cara o ambas salen cruz.

Matrices de probabilidades conjuntas para los Ejemplos 1 y 2

$$\boldsymbol{P} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} \end{bmatrix} \quad \text{y} \quad \boldsymbol{P} = \begin{bmatrix} \frac{1}{2} & \mathbf{0} \\ \mathbf{0} & \frac{1}{2} \end{bmatrix}.$$

Mostremos P en correcta notación matricial. La matriz muestra la probabilidad p_{ij} de cada par (x_i, y_j) comenzando con $(x_1, y_1) =$ (cara, cara) y $(x_1, y_2) =$ (cara, cruz). Observemos la suma de las filas p_1, p_2 y las sumas de columnas P_1, P_2 y la suma total $= 1$.

Matriz de probabilidad $P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}$ $\begin{matrix} p_{11}+p_{12} = \boldsymbol{p_1} \\ p_{21}+p_{22} = \boldsymbol{p_2} \end{matrix}$ $\begin{pmatrix} \text{primera} \\ \text{moneda} \end{pmatrix}$

(segunda moneda) sumas de columnas $\boldsymbol{P_1}$ $\boldsymbol{P_2}$ los 4 elementos suman 1

Esas sumas p_1, p_2 y P_1, P_2 son las probabilidades **marginales** de la matriz de probabilidad conjunta P:

$p_1 = p_{11} + p_{12} =$ probabilidades de cara en la **moneda 1** (la moneda 2 puede ser cara o cruz)
$P_1 = p_{11} + p_{21} =$ probabilidades de cara en la **moneda 2** (la moneda 1 puede ser cara o cruz)

El Ejemplo 1 mostró variables aleatorias *independientes*. Cada probabilidad p_{ij} equivale a p_i multiplicado por p_j $\left(\frac{1}{2}\right.$ multiplicado por $\frac{1}{2}$ da $p_{ij} = \frac{1}{4}$ en ese ejemplo$\left.\right)$. En este caso **la covarianza $\boldsymbol{\sigma_{12}}$ será cero**. Que sea cara o cruz la primera moneda no da información sobre la segunda moneda.

Covarianza cero $\boldsymbol{\sigma_{12}}$ para pruebas independientes $\boldsymbol{V} = \begin{bmatrix} \boldsymbol{\sigma_1^2} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\sigma_2^2} \end{bmatrix}$ = **matriz de covarianza diagonal $\boldsymbol{V}$.**

Los experimentos independientes satisfacen $\sigma_{12} = 0$ porque todo p_{ij} equivale a $(p_i)(p_j)$ en la ecuación (2):

$$\sigma_{12} = \sum_i \sum_j (p_i)(p_j)(x_i - m_1)(y_j - m_2) = \left[\sum_i (p_i)(x_i - m_1)\right]\left[\sum_j (p_j)(y_j - m_2)\right] = [\mathbf{0}][\mathbf{0}].$$

Ejemplo 3 Las monedas pegadas muestran una correlación perfecta. Que una sea cara significa que la otra también lo es. La covarianza σ_{12} cambia de 0 a $\boldsymbol{\sigma_1}$ **por** $\boldsymbol{\sigma_2}$. Este es el valor más grande posible de σ_{12}. Aquí es $\left(\frac{1}{2}\right)\left(\frac{1}{2}\right) = \sigma_{12} = \left(\frac{1}{4}\right)$, tal como lo confirma un cálculo independiente :

$$\textbf{Media} = \frac{\mathbf{1}}{\mathbf{2}} \qquad \boldsymbol{\sigma_{12}} = \frac{\mathbf{1}}{\mathbf{2}}\left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{2}\right) + \mathbf{0} + \mathbf{0} + \frac{\mathbf{1}}{\mathbf{2}}\left(0 - \frac{1}{2}\right)\left(0 - \frac{1}{2}\right) = \frac{\mathbf{1}}{\mathbf{4}}$$

Si sabemos que salió cara o cruz en la moneda 1 tenemos toda la información sobre qué ocurrió con la segunda moneda, que esta pegada, si salió cara o cruz :

Las monedas pegadas dan la mayor covarianza posible
Matriz de covarianza singular: determinante $= 0$

$$\boldsymbol{V}_{\textbf{pegadas}} = \begin{bmatrix} \boldsymbol{\sigma_1^2} & \boldsymbol{\sigma_1\sigma_2} \\ \boldsymbol{\sigma_1\sigma_2} & \boldsymbol{\sigma_2^2} \end{bmatrix}$$

Siempre $\sigma_1^2\sigma_2^2 \geq (\sigma_{12})^2$. Por lo tanto σ_{12} está *entre* $-\sigma_1\sigma_2$ *y* $\sigma_1\sigma_2$. La matriz V es **definida positiva** (o en el caso singular de las monedas pegadas, V es **semidefinida positiva**). Estos son resultados importantes válidos para todas las matrices de covarianza V de M por M para M experimentos.

Observe que la **matriz de covarianza muestral** $\boldsymbol{S}$ de N experimentos es sin duda semidefinida. Cada muestra nueva X = (edad, altura, peso) contribuye a la **media muestral** $\overline{\boldsymbol{X}}$ (un vector). Cada término de rango uno $(X_i - \overline{X})(X_i - \overline{X})^{\mathrm{T}}$ es semidefinido positivo y los sumamos para obtener la matriz S. No hay probabilidades en S, solo información sobre datos reales:

$$\boxed{\overline{\boldsymbol{X}} = \frac{X_1 + \cdots + X_N}{N} \qquad \boldsymbol{S} = \frac{(X_1 - \overline{X})(X_1 - \overline{X})^{\mathrm{T}} + \cdots + (X_N - \overline{X})(X_N - \overline{X})^{\mathrm{T}}}{N - 1}} \tag{3}$$

La matriz de covarianza V es semidefinida positiva

Regresemos a la covarianza *esperada* σ_{12} entre dos experimentos 1 y 2 (dos monedas):

$$\boxed{\begin{array}{rcl} \sigma_{12} & = & \text{valor esperado de } \ [(\textit{resultado}\,1 - \textit{media}\,1) \ \text{por} \ (\textit{resultado}\,2 - \textit{media}\,2)] \\ \boldsymbol{\sigma_{12}} & = & \boldsymbol{\sum\sum p_{ij}\,(x_i - m_1)\,(y_j - m_2)}. \text{ La suma incluye todos los } i, j. \end{array}} \tag{4}$$

$p_{ij} \geq 0$ es la probabilidad de obtener el resultados x_i en el experimento 1 **y** resultado y_j en el experimento 2. Algunos pares de resultados deben aparecer. Por lo tanto las N^2 probabilidades conjuntas p_{ij} suman 1.

La probabilidad total (todos los pares) es 1 $$\sum_{\text{todo } i,j} \sum p_{ij} = 1. \tag{5}$$

He aquí otro dato que necesitamos. *Enfoquémonos en un resultado particular* x_i en el experimento 1. Permitamos *todos los resultados* y_j en el experimento 2. Sumemos las probabilidades de $(x_i, y_1), (x_i, y_2), \ldots, (x_i, y_n)$:

$$\textbf{Suma de fila } \boldsymbol{p_i} \textbf{ de } \boldsymbol{P} \quad \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \boldsymbol{p_{ij}} = \textbf{probabilidad } \boldsymbol{p_i} \textbf{ de } \boldsymbol{x_i} \textbf{ en el experimento 1.} \tag{6}$$

Algún y_j debe ocurrir en el experimento 2. Estén las dos monedas pegadas o no, obtenemos la misma respuesta $\frac{1}{2}$ para la probabilidad $p_H = p_{HH} + p_{HT}$ de que esa moneda 1 sea cara:

$$\text{(separadas) } P_{HH} + P_{HT} = \frac{1}{4} + \frac{1}{4} = \boldsymbol{\frac{1}{2}} \qquad \text{(pegadas) } P_{HH} + P_{HT} = \frac{1}{2} + 0 = \boldsymbol{\frac{1}{2}}.$$

Ese razonamiento básico nos permite escribir una fórmula matricial que incluye la covarianza σ_{12} junto con las varianzas σ_1^2 y σ_2^2 para los experimentos 1 y 2. Obtenemos la matriz de covarianza V al sumar las matrices V_{ij} para cada par (i, j):

$$\boxed{\textbf{Matriz de covarianza } \boldsymbol{V} \textbf{ = suma de los } \boldsymbol{V_{ij}} \qquad \boldsymbol{V} = \sum_{\text{todo } i,j} \sum p_{ij} \begin{bmatrix} (x_i - m_1)^2 & (x_i - m_1)(y_j - m_2) \\ (x_i - m_1)(y_j - m_2) & (y_j - m_2)^2 \end{bmatrix}} \tag{7}$$

Fuera de la diagonal, ésta es la ecuación (2) para la covarianza σ_{12}. En la diagonal, obtenemos las varianzas usuales σ_1^2 y σ_2^2. Mostraré en detalle cómo obtenemos $V_{11} = \sigma_1^2$ a partir de la ecuación (6). Permitiendo todos los j la suma da la probabilidad p_i de x_i en el experimento 1 :

$$\boldsymbol{V_{11}} = \sum_{\text{todo } i,j} \sum p_{ij}(x_i - m_1)^2 = \sum_{\text{todo } i} (\text{probabilidad de } x_i)\,(x_i - m_1)^2 = \boldsymbol{\sigma_1^2}. \tag{8}$$

Démosle otra mirada a esta fórmula, ya que es la clave para producir la matriz de covarianza completa con una fórmula (7). La belleza de esa fórmula es que combina matrices V_{ij} de 2 por 2. Y la matriz V_{ij} en (7) para cada par de resultados i, j es **semidefinida positiva** :

V_{ij} tiene elementos diagonales $p_{ij}(x_i - m_1)^2 \geq 0$ y $p_{ij}(y_j - m_2)^2 \geq 0$ y $\det(V_{ij}) = 0$.

Esa matriz V_{ij} tiene rango 1. En la ecuación (7) p_{ij} se multiplica *por la columna* $\boldsymbol{U}$ *por la fila* $\boldsymbol{U}^{\mathrm{T}}$:

$$\begin{bmatrix} (x_i - m_1)^2 & (x_i - m_1)(y_j - m_2) \\ (x_i - m_1)(y_j - m_2) & (y_j - m_2)^2 \end{bmatrix} = \begin{bmatrix} x_i - m_1 \\ y_j - m_2 \end{bmatrix} \begin{bmatrix} x_i - m_1 & y_j - m_2 \end{bmatrix} \tag{9}$$

Cada matriz $\boldsymbol{p_{ij} U U^{\mathrm{T}}}$ *es semidefinida positiva*. Entonces toda la matriz V (la suma de esas matrices de rango 1) es **al menos semidefinida** y será probablemente definida positiva.

La matriz de covarianza V es definida positiva a menos que los experimentos sean dependientes.

Ahora pasamos de dos variables x e y a M variables como edad, altura y peso. El resultado de cada prueba es un vector X con M componentes. (Cada niño tiene un vector edad-altura-peso $\boldsymbol{X}$ con 3 componentes.) La matriz de covarianza V ahora es de M por M. La matriz $\boldsymbol{V}$ se crea a partir de los vectores resultantes $\boldsymbol{X}$ y su promedio $\overline{\boldsymbol{X}} = \mathbf{E}[\boldsymbol{X}]$:

Matriz de covarianza	$\boldsymbol{V} = \mathbf{E}\left[(\boldsymbol{X} - \overline{\boldsymbol{X}})\,(\boldsymbol{X} - \overline{\boldsymbol{X}})^{\mathrm{T}}\right]$	$\boldsymbol{V_{ij}} = \boldsymbol{p_{ij}}\,(\boldsymbol{X_i} - \boldsymbol{m_i})\,(\boldsymbol{X_j} - \boldsymbol{m_j})$ (10)

Recordemos que $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ y $\overline{\boldsymbol{X}}\,\overline{\boldsymbol{X}}^{\mathrm{T}}$ = (columna) (fila) son matrices de M por M.

Para $M = 1$ (una variable) podemos ver que $\overline{X}$ es la media m y V es la varianza σ^2. Para $M = 2$ (dos monedas) vemos que $\overline{X}$ es (m_1, m_2) y V coincide con la ecuación (7). La esperanza siempre suma los resultados multiplicados por sus probabilidades. Para edad-altura-peso, el resultado podría ser $X = (5$ años, 31 pulgadas, 48 libras) y su probabilidad es $\boldsymbol{p_{5,31,48}}$.

Ahora consideremos una idea nueva. *Tomemos cualquier combinación lineal* $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{X} = c_1 X_1 + \cdots + c_M X_M$. Con $\boldsymbol{c} = (6, 2, 5)$ esto sería $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{X} = 6 \times \text{edad} + 2 \times \text{altura} + 5 \times \text{peso}$. Por linealidad sabemos que el valor esperado $\mathrm{E}[\boldsymbol{c}^{\mathrm{T}}\boldsymbol{X}]$ es $\boldsymbol{c}^{\mathrm{T}}\mathrm{E}[\boldsymbol{X}] = \boldsymbol{c}^{\mathrm{T}}\overline{\boldsymbol{X}}$:

$$\mathbf{E}[\boldsymbol{c}^{\mathrm{T}}\boldsymbol{X}] = \boldsymbol{c}^{\mathrm{T}}\mathbf{E}[\boldsymbol{X}] = 6\,(\text{edad esperada}) + 2\,(\text{altura esperada}) + 5\,(\text{peso esperado}).$$

No solo conocemos la media de $c^{\mathrm{T}}X$, sino también su *varianza* $\boldsymbol{\sigma^2 = c^{\mathrm{T}} V c}$:

$$\begin{aligned} \text{Varianza de } \boldsymbol{c^{\mathrm{T}} X} &= \mathbf{E}\left[\left(\boldsymbol{c^{\mathrm{T}} X} - \boldsymbol{c^{\mathrm{T}}}\,\overline{\boldsymbol{X}}\right)\left(\boldsymbol{c^{\mathrm{T}} X} - \boldsymbol{c^{\mathrm{T}}}\,\overline{\boldsymbol{X}}\right)^{\mathrm{T}}\right] \\ &= \boldsymbol{c^{\mathrm{T}}}\mathbf{E}\left[\left(\boldsymbol{X} - \overline{\boldsymbol{X}}\right)\left(\boldsymbol{X} - \overline{\boldsymbol{X}}\right)^{\mathrm{T}}\right]\boldsymbol{c} = \boldsymbol{c^{\mathrm{T}}\, V c} \end{aligned} \tag{11}$$

Ahora el punto clave: *la varianza de* $\boldsymbol{c^{\mathrm{T}} X}$ *nunca puede ser negativa.* Así $\boldsymbol{c^{\mathrm{T}} V c \geq 0}$. Prueba nueva: *la matriz de covarianza* V *es semidefinida positiva por el test de energía* $\boldsymbol{c^{\mathrm{T}} V c \geq 0}$.

Las matrices de covarianza V nos muestran el vínculo entre la probabilidad y el álgebra lineal: $\boldsymbol{V}$ **es igual a** $\boldsymbol{Q \Lambda Q^{\mathrm{T}}}$ con autovalores $\lambda_i \geq 0$ y autovectores ortonormales $\boldsymbol{q_1}$ a $\boldsymbol{q_M}$.

> **Diagonalizar la matriz de covarianza V significa encontrar M experimentos *independientes* que son combinaciones de los M experimentos originales.**

Confesión No estoy totalmente satisfecho con esa prueba basada en $\boldsymbol{c^{\mathrm{T}} V c \geq 0}$. El símbolo de la esperanza **E** esconde la idea clave de la probabilidad conjunta. Permítame mostrarle directamente que la matriz de covarianza V es semidefinida positiva (al menos para el ejemplo edad-altura-peso). La prueba es simplemente que $\boldsymbol{V}$ **es la suma de la probabilidad conjunta** $\boldsymbol{p_{ahw}}$ **de cada combinación (edad, altura, peso) multiplicada por la matriz semidefinida positiva** $\boldsymbol{UU^{\mathrm{T}}}$, con $U = X - \overline{X}$:

$$\boldsymbol{V} = \sum_{\text{todo } a,h,w} \boldsymbol{p_{ahw}\, U\, U^{\mathrm{T}}} \quad \text{con} \quad \boldsymbol{U} = \begin{bmatrix} \text{edad} \\ \text{altura} \\ \text{peso} \end{bmatrix} - \begin{bmatrix} \text{edad media} \\ \text{altura media} \\ \text{peso medio} \end{bmatrix}. \tag{12}$$

Esto es exactamente como la matriz de lanzamiento de moneda V de 2 por 2 en la Ecuación (7). Ahora $M = 3$.

Lo valioso del símbolo de esperanza E es que también permite las *pdf*: funciones de densidad de probabilidad como $p(x, y, z)$ para variables aleatorias continuas x, y y z. Si permitimos todos los números como posibles edades, alturas y pesos, en lugar de edad $i = 0, 1, 2, 3 \ldots$, entonces necesitamos $p(x, y, z)$ en lugar de p_{ijk}. Todas las sumas de esta sección del libro se transforman a integrales. Pero todavía tenemos $\boldsymbol{V} = \mathrm{E}\left[\boldsymbol{UU^{\mathrm{T}}}\right]$:

$$\textbf{Matriz de covarianza } \boldsymbol{V} = \iiint p(x, y, z)\, \boldsymbol{UU^{\mathrm{T}}}\, dx\, dy\, dz \quad \text{con} \quad \boldsymbol{U} = \begin{bmatrix} x - \overline{x} \\ y - \overline{y} \\ z - \overline{z} \end{bmatrix}. \tag{13}$$

Siempre $\iiint p = 1$. Los Ejemplos 1–2 enfatizan cómo p puede dar $\boldsymbol{V}$ diagonal o $\boldsymbol{V}$ singular:

Variables independientes $\boldsymbol{x, y, z}$ $\quad p(x, y, z) = \boldsymbol{p_1(x)\, p_2(y)\, p_3(z)}$.

Variables dependientes $\boldsymbol{x, y, z}$ $\quad p(x, y, z) = \mathbf{0}$ excepto cuando $cx + dy + ez = 0$.

La media y la varianza de $z = x + y$

Comencemos con la media muestral. Tenemos N muestras de x. Su media (= promedio) es el número m_x. También tenemos N muestras de y y su media es m_y. **La media muestral de $z = x + y$ es claramente $m_z = m_x + m_y$** :

$$\boxed{\textbf{Media de la suma} = \textbf{Suma de las medias} \qquad \frac{1}{N}\sum_1^N (x_i + y_i) = \frac{1}{N}\sum_1^N x_i + \frac{1}{N}\sum_1^N y_i.} \tag{14}$$

Qué agradable es ver algo tan sencillo. La media *esperada* de $z = x + y$ no se ve tan simple, pero debe suceder que $\mathbf{E}[z] = \mathbf{E}[x] + \mathbf{E}[y]$. He aquí una manera de verlo.

La probabilidad conjunta del par (x_i, y_j) es p_{ij}. Su valor depende de si los experimentos son independientes o no, lo cual no sabemos. Pero para la esperanza de la suma $z = x+y$, la dependencia o independencia de x e y no importan. *Los valores esperados igual se suman* :

$$\mathbf{E}[x + y] = \sum_i \sum_j p_{ij}(x_i + y_j) = \sum_i \sum_j p_{ij} x_i + \sum_i \sum_j p_{ij} y_j. \tag{15}$$

Todas las sumas van de 1 a N. Podemos sumar en cualquier orden. Para el primer término del lado derecho, sumamos p_{ij} a lo largo de la fila i de la matriz de probabilidad P para obtener p_i. Esa suma doble nos da $\mathrm{E}[x]$:

$$\sum_i \sum_j p_{ij} x_i = \sum_i (p_{i1} + \cdots + p_{iN}) x_i = \sum_i p_i x_i = \mathbf{E}[x].$$

Para el último término, sumamos p_{ij} a lo largo de la columna j de la matriz para obtener la probabilidad P_j de y_j. Esos pares (x_1, y_j), (x_2, y_j), ... y (x_N, y_j) son todas las maneras de obtener y_j :

$$\sum_i \sum_j p_{ij} y_j = \sum_j (p_{1j} + \cdots + p_{Nj}) y_j = \sum_j P_j y_j = \mathbf{E}[y].$$

Ahora la ecuación (15) establece que $\mathrm{E}[x + y] = \mathrm{E}[x] + \mathrm{E}[y]$. Media de la suma = Suma de las medias.

¿ Qué hay de la varianza de $z = x + y$? Las probabilidades conjuntas p_{ij} y la covarianza σ_{xy} estarán involucradas. Separemos la varianza de $x + y$ en tres simples términos :

$$\begin{aligned} \boldsymbol{\sigma_z^2} &= \sum\sum p_{ij}(x_i + y_j - m_x - m_y)^2 \\ &= \sum\sum p_{ij}(x_i - m_x)^2 + \sum\sum p_{ij}(y_j - m_y)^2 + 2\sum\sum p_{ij}(x_i - m_x)(y_j - m_y) \end{aligned}$$

El primer término es $\boldsymbol{\sigma_x^2}$. El segundo término es $\boldsymbol{\sigma_y^2}$. El úlitmo término es $\boldsymbol{2\sigma_{xy}}$.

$$\boxed{\textbf{La varianza de } z = x + y \quad \textbf{es} \quad \boldsymbol{\sigma_z^2 = \sigma_x^2 + \sigma_y^2 + 2\sigma_{xy}.}} \tag{16}$$

La matriz de covarianza de $Z = AX$

He aquí una buena manera de interpretar σ_z^2 cuando $z = x + y$. Pensemos (x, y) como un vector columna $\boldsymbol{X}$. Pensemos a la matriz $\boldsymbol{A} = \begin{bmatrix} 1 & 1 \end{bmatrix}$ de 1 por 2 multiplicando ese vector $\boldsymbol{X} = (x, y)$. Así $\boldsymbol{AX}$ es la suma $z = x + y$. La varianza $\boldsymbol{\sigma_z^2}$ de la Ecuación (16) se escribe en notación matricial como

$$\boldsymbol{\sigma_z^2} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \text{Que es} \quad \boldsymbol{\sigma_z^2 = AVA^{\mathrm{T}}}. \tag{17}$$

Se puede ver que $\sigma_z^2 = AVA^{\mathrm{T}}$ de (17) coincide con $\sigma_x^2 + \sigma_y^2 + 2\sigma_{xy}$ de (16).

Vayamos ahora al punto principal. El vector $\boldsymbol{X}$ podría tener M componentes provenientes de M experimentos (en lugar de solo 2). Esos experimentos tendrán una matriz de covarianza $\boldsymbol{V_X}$ de M por M. La matriz A podría ser de K por M. Así $\boldsymbol{AX}$ es un vector con K combinaciones de los M resultados (en lugar de una combinación $x + y$ de dos resultados).

Ese vector $\boldsymbol{Z = AX}$ de longitud K tiene una matriz de covarianza $V_{\boldsymbol{Z}}$ de K por K. Así la regla para las matrices de covarianza, de las cuales la Ecuación (17) era tan solo un ejemplo de 1 por 2 es esta hermosa fórmula: la matriz de covarianza de $\boldsymbol{AX}$ es $\boldsymbol{A}$ **(matriz de covarianza de** $\boldsymbol{X}$**)** $\boldsymbol{A^{\mathrm{T}}}$

$$\boxed{\textbf{La matriz de covarianza de } \boldsymbol{Z = AX} \textbf{ es } \boldsymbol{V_Z = AV_X A^{\mathrm{T}}}} \tag{18}$$

Para mí, esta elegante fórmula muestra la belleza de la multiplicación matricial. No demostraré esta fórmula, pero es digna de admiración. Se usa constantemente en aplicaciones.

La correlación ρ

La correlación ρ_{xy} está estrechamente relacionada a la covarianza σ_{xy}. Ambas miden la dependencia o la independencia. Comenzamos reescalando o "estandarizando" las variables aleatorias x e y. **La nuevas** $\boldsymbol{X = x/\sigma_x}$ **e** $\boldsymbol{Y = y/\sigma_y}$ **tienen varianza** $\boldsymbol{\sigma_X^2 = \sigma_Y^2 = 1}$. Esto es como cuando dividimos un vector $\boldsymbol{v}$ por su longitud para producir un vector unitario $\boldsymbol{v}/||\boldsymbol{v}||$ de longitud 1.

La correlación entre $\boldsymbol{x}$ **e** $\boldsymbol{y}$ **es la covarianza de** $\boldsymbol{X}$ **e** $\boldsymbol{Y}$. Si la covarianza original de x e y fuera σ_{xy}, entonces reescalar a X e Y dividirá por σ_x y por σ_y:

$$\boxed{\textbf{Correlación } \boldsymbol{\rho_{xy}} = \frac{\boldsymbol{\sigma_{xy}}}{\boldsymbol{\sigma_x \sigma_y}} = \textbf{covarianza de } \frac{\boldsymbol{x}}{\boldsymbol{\sigma_x}} \textbf{ y } \frac{\boldsymbol{y}}{\boldsymbol{\sigma_y}}} \quad \textbf{Siempre } \boldsymbol{-1 \le \rho_{xy} \le 1}$$

Covarianza cero da correlación cero. *Variables aleatorias independientes* dan $\rho_{xy} = 0$.

Sabemos que $(\sigma_{xy})^2 \le \sigma_x^2 \sigma_y^2$ es cierto siempre (la matriz de covarianza V es al menos semi-definida positiva). Así $\boldsymbol{(\rho_{xy})^2 \le 1}$. Correlación cerca de $\rho = +1$ significa una fuerte dependencia en la misma dirección: a menudo votan lo mismo. Una correlación negativa significa que y tiende a estar debajo de su media cuando x está por encima de su media: van en dirección contraria, votan en sentido opuesto, ρ está cerca de -1.

Ejemplo 4 *Supongamos que y es $-x$.* Un lanzamiento de moneda tiene resultados $x = 0$ o 1. El mismo lanzamiento tiene resultados $y = 0$ o -1. La media m_x es $\frac{1}{2}$ para una moneda justa y m_y es $-\frac{1}{2}$. La covarianza de x e y es $\sigma_{xy} = -\sigma_x\sigma_y$. Así, $\boldsymbol{\rho_{xy} = -1}$. En este caso, la matriz de correlación R tiene determinante cero (singular y solo semidefinida positiva):

Matriz de correlación $\boldsymbol{R = \begin{bmatrix} 1 & \rho_{xy} \\ \rho_{xy} & 1 \end{bmatrix}}$ $\qquad \boldsymbol{R = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}}$ cuando $\boldsymbol{y = -x}$

R siempre tiene 1 sobre la diagonal porque normalizamos a $\sigma_X = \sigma_Y = 1$. R es la matriz de correlación para x e y, y también es la matriz de covarianza para $X = x/\sigma_x$ e $Y = y/\sigma_y$.

Ese número ρ_{xy} es también conocido como el coeficiente de Pearson.

Ejemplo 5 Si las variables aleatorias x, y, z son *independientes. ¿ Cómo es la matriz R ?*

Respuesta *R es la matriz identidad.* Las tres correlaciones $\rho_{xx}, \rho_{yy}, \rho_{zz}$ son 1 por definición. Las tres correlaciones cruzadas $\rho_{xy}, \rho_{xz}, \rho_{yz}$ son cero por independencia.

La matriz de correlación R se obtiene de la matriz de covarianza V, cuando reescalamos cada fila y cada columna. Dividimos cada fila i y columna i por la i-ésima, desviación estándar σ_i.

(a) $\boldsymbol{R = DVD}$ tomando la matriz diagonal $D = \text{diag}\,[1/\sigma_1, \ldots, 1/\sigma_M]$. Así cada $R_{ii} = 1$.

(b) Si la covarianza V es definida positiva, la correlación $R = DVD$ también es definida positiva.

■ EJEMPLO RESUELTO ■

Sean x e y variables aleatorias independientes con media 0 y varianza 1. Así la matriz de covarianza $\boldsymbol{V_X}$ para $\boldsymbol{X} = (x, y)$ es la matriz identidad de 2 por 2. ¿Cuáles son la media $\boldsymbol{m_Z}$ y la matriz de covarianza $\boldsymbol{V_Z}$ para el vector de tres componentes $\boldsymbol{Z} = (x, y, ax + by)$?

Solución

Z está conectada a X por medio de A $\qquad \boldsymbol{Z} = \begin{bmatrix} x \\ y \\ ax + by \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ a & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = A\boldsymbol{X}.$

El vector $\boldsymbol{m_X}$ contiene las medias de las M componentes de $\boldsymbol{X}$. El vector $\boldsymbol{m_Z}$ contiene las medias de las K componentes de $\boldsymbol{Z} = A\boldsymbol{X}$. La conexión matricial entre las medias de $\boldsymbol{X}$ y $\boldsymbol{Z}$ tiene que ser lineal: $\boldsymbol{m_Z} = A\,\boldsymbol{m_X}$. La media de $ax + by$ es $am_x + bm_y$.

Cuando $\boldsymbol{V_X}$ es la matriz identidad de 2 por 2, la matriz de covarianza de $\boldsymbol{Z}$ es $\boldsymbol{V_Z} = AA^{\mathrm{T}}$:

$$\boldsymbol{V_Z} = \begin{matrix}\textbf{matriz de covarianza de} \\ \boldsymbol{Z = (x, y, ax + by)}\end{matrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ a & b \end{bmatrix} \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \boldsymbol{a} \\ \mathbf{0} & \mathbf{1} & \boldsymbol{b} \\ \boldsymbol{a} & \boldsymbol{b} & \boldsymbol{a^2 + b^2} \end{bmatrix}.$$

Interpretación : x e y son independientes con covarianza $\sigma_{xy} = 0$. Así la covarianza entre x y $ax+by$ es a y la covarianza entre y y $ax + by$ es b. Finalmente, la ecuación (18) da la varianza de $ax + by$:

$$\textbf{Usemos } \boldsymbol{V_Z = AV_X A^{\mathrm{T}}} \qquad \sigma^2_{ax+by} = \sigma^2_{ax} + \sigma^2_{by} + 2\sigma_{ax,by} = a^2 + b^2 + 0.$$

La matriz $\boldsymbol{V_Z}$ de 3 por 3 es *singular*. Su determinante es $a^2+b^2-a^2-b^2 = 0$. La tercera componente $z = ax + by$ es completamente dependiente de x e y. El rango de $\boldsymbol{V_Z}$ es solo 2.

Ejemplo del GPS La señal de un satélite GPS incluye el horario de su partida. El reloj receptor da el horario de llegada. El receptor multiplica el tiempo de viaje por la velocidad de la luz. Así obtiene la distancia a ese satélite. Las distancias a cuatro o más satélites determinan la posición del receptor (¡ usando mínimos cuadrados !).

Un problema : la velocidad de la luz cambia en la ionosfera. Pero la corrección será casi la misma para todos los receptores cercanos. Si un receptor se mantiene en una posición conocida podemos tomar las diferencias a partir de él. El **GPS diferencial** reduce la varianza del error dejando fija la posición de un receptor :

$$\begin{matrix}\textbf{Matriz de diferencias} \\ \boldsymbol{A = [\,1 \quad -1\,]}\end{matrix} \qquad \begin{matrix}\textbf{Matriz de covarianza} \\ \boldsymbol{V_Z = AV_X A^{\mathrm{T}}}\end{matrix} \qquad \begin{aligned} \boldsymbol{V_Z} &= \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \\ &= \boldsymbol{\sigma_1^2 - 2\sigma_{12} + \sigma_2^2} \end{aligned}$$

Los errores en la velocidad de la luz desaparecieron. Así se logra una precisión de centímetros en las posiciones. (Las ideas claves se encuentran en la página 320 de *Algorithms for Global Positioning* de Borre y Strang.) El mundo de los GPS trata sobre el tiempo, el espacio y precisiones extraordinaria.

Problemas de la Sección V.4

1 (a) Calcule la varianza σ^2 cuando las probabilidades de lanzamiento de moneda son p y $1-p$ (cruz $= 0$, cara $= 1$).

(b) La suma de N lanzamientos independientes (0 o 1) es la cantidad de caras después de N pruebas. La regla (16-17-18) para la varianza de una suma da $\sigma^2 =$ _____.

2 ¿ Cuál es la covarianza σ_{kl} entre los resultados $x_1, \ldots, x_n$ del experimento 3 y los resultados $y_1, \ldots, y_n$ del experimento 5 ? Su fórmula será similar a la de σ_{12} en la ecuación (2). Así los elementos $(3, 5)$ y $(5, 3)$ de la matriz de covarianza V son iguales.

3 Para $M = 3$ experimentos, la matriz de varianza-covarianza V será de 3 por 3. Habrá una probabilidad p_{ijk} de que los tres resultados sean x_i, y_j y z_k. Escriba una fórmula similar a la Ecuación (7) para la matriz V.

4 ¿Cuál es la matriz de covarianza V para $M = 3$ experimentos independientes con medias m_1, m_2, m_3 y varianzas $\sigma_1^2, \sigma_2^2, \sigma_3^2$?

5 Cuando la matriz de covarianza para las variables $\boldsymbol{X}$ es $\boldsymbol{V}$, la matriz de covarianza para los variables $\boldsymbol{Z} = \boldsymbol{AX}$ es $\boldsymbol{AVA^{\mathrm{T}}}$. Explique esta bella fórmula usando linealidad:

$$V_Z = \mathrm{E}\left[(AX - \overline{AX})(AX - \overline{AX})^{\mathrm{T}}\right] = A\mathrm{E}\left[(X - \overline{X})(X - \overline{X})^{\mathrm{T}}\right] A^{\mathrm{T}}.$$

Los problemas 6–10 son sobre la probabilidad condicional de que $\boldsymbol{Y = y_j}$ cuando sabemos que $\boldsymbol{X = x_i}$.
Nota: **Prob $(\boldsymbol{Y = y_j | X = x_i})$** = probabilidad del resultado y_j sabiendo que $X = x_i$.

Ejemplo 1 *La moneda* 1 *está pegada a la moneda* 2. Así Prob ($Y =$ cara cuando $X =$ cara) es **1**.
Ejemplo 2 *Lanzamientos de monedas independientes*: X no da información sobre Y. Es inútil saber X. Así Prob ($Y =$ cara $|X =$ cara) es lo mismo que Prob ($Y =$ cara).

6 Explique la **regla de la suma** de la probabilidad condicional:

$$\text{Prob}\,(Y = y_j) = \text{ suma sobre todos los resultados } x_i \text{ de Prob}\,(Y = y_j | X = x_i).$$

7 La matriz P de n por n contiene **probabilidades conjuntas $\boldsymbol{p_{ij}}$** = Prob ($X = x_i$ **y** $Y = y_j$).

Explique por qué la Prob $(Y = y_j | X = x_i)$ condicional es igual a $\dfrac{p_{ij}}{p_{i1} + \cdots + p_{in}} = \dfrac{p_{ij}}{p_i}$.

8 Para esta matriz de probabilidad conjunta con Prob $(x_1, y_2) = 0.3$, encuentre Prob $(y_2 | x_1)$ y Prob (x_1).

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} = \begin{bmatrix} 0.1 & 0.3 \\ 0.2 & 0.4 \end{bmatrix}$$

Los elementos p_{ij} suman 1.
Algunos i, j deben ocurrir.

9 Explique la **regla del producto** para la probabilidad condicional:

$p_{ij} =$ Prob $(X = x_i$ **y** $Y = y_j)$ es igual a Prob $(Y = y_j | X = x_i)$ por Prob $(X = x_i)$.

10 Deduzca este **teorema de Bayes** para p_{ij} a partir de la regla del producto del problema 9:

$$\text{Prob}\,(Y = y_j \text{ y } X = x_i) = \frac{\text{Prob}\,(X = x_i | Y = y_j)\ \text{Prob}\,(Y = y_j)}{\text{Prob}\,(X = x_i)}$$

Los "bayesianos" usan información previa. Los "frecuentistas" solo usan información muestral.

V.5. Gaussiana multivariada y mínimos cuadrados ponderados

La densidad de probabilidad normal $p(x)$ (la gaussiana) depende de solo dos números:

$$\textbf{Media } m \textbf{ y varianza } \sigma^2 \qquad p(x) = \frac{1}{\sqrt{2\pi}\,\sigma}\, e^{-(x-m)^2/2\sigma^2}. \tag{1}$$

El gráfico de $p(x)$ es una curva acampanada centrada en $x = m$. La variable continua x puede tomar cualquier valor entre $-\infty$ e ∞. Con probabilidad cercana a $\frac{2}{3}$, la variable aleatoria x estará entre $m - \sigma$ y $m + \sigma$ (menos de una desviación estándar σ de su media m).

$$\int_{-\infty}^{\infty} p(x)\,dx = 1 \quad \text{y} \quad \int_{m-\sigma}^{m+\sigma} p(x)\,dx = \frac{1}{\sqrt{2\pi}} \int_{-1}^{1} e^{-X^2/2}\,dX \approx \frac{2}{3}. \tag{2}$$

Esa integral tiene un cambio de variables de x a $X = (x - m)/\sigma$. Esto simplifica el exponente a $-X^2/2$ y simplifica los límites de integración a -1 y 1. El $1/\sigma$ de $p(x)$ desaparece porque dX es igual a dx/σ. Toda gausiana se vuelve una **gaussiana estándar** con media $m = 0$ y varianza $\sigma^2 = 1$:

$$\textbf{La distribución normal estándar } N(0,1) \quad \textbf{satisface} \quad p(x) = \frac{1}{\sqrt{2\pi}}\, e^{-x^2/2}. \tag{3}$$

Integrar $p(x)$ de $-\infty$ a x da la distribución acumulada $F(x)$: la probabilidad de que una muestra aleatoria sea menor que x. Esa probabilidad será $F = \frac{1}{2}$ en $x = 0$ (la media).

Gaussianas bidimensionales

Ahora tenemos $M = 2$ variables aleatorias gaussianas x e y con medias m_1 y m_2. Tienen varianzas σ_1^2 y σ_2^2. Si son *independientes*, entonces su densidad de probabilidad $p(x, y)$ es tan solo $p_1(x)$ **multiplicado por** $p_2(y)$. Multiplicamos las probabilidades cuando las variables son independientes:

$$x \textbf{ e } y \textbf{ independientes} \quad p(x,y) = \frac{1}{2\pi\sigma_1\sigma_2}\, e^{-(x-m_1)^2/2\sigma_1^2}\, e^{-(y-m_2)^2/2\sigma_2^2} \tag{4}$$

La covarianza de x e y será $\sigma_{12} = 0$. La matriz de covarianza V será *diagonal*. Las varianzas σ_1^2 y σ_2^2 están siempre sobre la diagonal principal de V. El exponente en $p(x, y)$ es tan solo la suma del exponente x y del exponente y. Es bueno notar que los dos exponentes pueden combinarse en $-\frac{1}{2}(x - m)^{\mathrm{T}} V^{-1} (x - m)$ con la matriz de covarianza inversa V^{-1} en el medio. Este exponente es $-[x - m]^{\mathrm{T}} V^{-1} [x - m]/2$:

$$-\frac{(x-m_1)^2}{2\sigma_1^2} - \frac{(y-m_2)^2}{2\sigma_2^2} = -\frac{1}{2} \begin{bmatrix} x - m_1 & y - m_2 \end{bmatrix} \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}^{-1} \begin{bmatrix} x - m_1 \\ y - m_2 \end{bmatrix} \tag{5}$$

x e y no independientes

Estamos listos para renunciar a la independencia. El exponente (5) con V^{-1} todavía es correcto cuando V ya no es más una matriz diagonal. **Ahora la gaussiana depende de un vector m y una matriz V.**

Cuando $M = 2$, la primera variable x puede darnos información parcial sobre la segunda variable y (y vice versa). Tal vez parte de y está determinada por x y parte es verdaderamente independiente. Es en la matriz de covarianza V de M por M donde las dependencias entre las M variables $\boldsymbol{x} = x_1, \ldots, x_M$ se ven reflejadas. **La inversa de la matriz de covarianza $\boldsymbol{V^{-1}}$ aparece en $\boldsymbol{p(x)}$:**

Distribución de probabilidad de una gaussiana multivariada

$$\boxed{p(\boldsymbol{x}) = \frac{1}{(\sqrt{2\pi})^M \sqrt{\det V}}\, e^{-(\boldsymbol{x}-\boldsymbol{m})^{\mathrm{T}} \boldsymbol{V^{-1}} (\boldsymbol{x}-\boldsymbol{m})/2}} \tag{6}$$

Los vectores $\boldsymbol{x} = (x_1, \ldots, x_M)$ y $\boldsymbol{m} = (m_1, \ldots, m_M)$ contienen las variables aleatorias y sus medias. Las M raíces cuadradas de 2π y el determinante de V se incluyen para lograr que la probabilidad total sea igual a 1. Verifiquemos usando álgebra lineal. Tomamos los autovalores λ y los autovectores ortonormales $\boldsymbol{q}$ de la matriz simétrica $V = Q\Lambda Q^{\mathrm{T}}$. Así $\boldsymbol{V^{-1}} = \boldsymbol{Q\Lambda^{-1}Q^{\mathrm{T}}}$:

$$\boldsymbol{X} = \boldsymbol{x} - \boldsymbol{m} \qquad (\boldsymbol{x} - \boldsymbol{m})^{\mathrm{T}} \boldsymbol{V^{-1}} (\boldsymbol{x} - \boldsymbol{m}) = \boldsymbol{X^{\mathrm{T}} Q \Lambda^{-1} Q^{\mathrm{T}} X} = \boldsymbol{Y^{\mathrm{T}} \Lambda^{-1} Y}$$

Observe: las combinaciones $\boldsymbol{Y} = \boldsymbol{Q^{\mathrm{T}} X} = \boldsymbol{Q^{\mathrm{T}}(x - m)}$ son estadísticamente independientes. *Su matriz de covarianza Λ es diagonal.*

> Este paso de diagonalizar V por su matriz de autovectores Q es lo mismo que descorrelacionar las variables aleatorias. Las covarianzas de las nuevas variables $Y_1, \ldots Y_M$ son cero. Este es el punto donde el álgebra lineal asiste al cálculo para calcular integrales multidimensionales.

La integral de $p(\boldsymbol{x})$ no cambia cuando centramos la variable $\boldsymbol{x}$ restando $\boldsymbol{m}$ para obtener $\boldsymbol{X}$, y rotamos esa variable para obtener $\boldsymbol{Y} = \boldsymbol{Q^{\mathrm{T}} X}$. La matriz $\boldsymbol{\Lambda}$ es diagonal. Entonces la integral que queremos calcular se divide en M integrales separadas, de una dimensión, que ya conocemos :

$$\int \cdots \int \boldsymbol{e^{-Y^{\mathrm{T}} \Lambda^{-1} Y/2}}\, \boldsymbol{dY} = \left(\int_{-\infty}^{\infty} e^{-y_1^2/2\lambda_1}\, dy_1 \right) \cdots \left(\int_{-\infty}^{\infty} e^{-y_M^2/2\lambda_M}\, dy_M \right)$$

$$= \left(\sqrt{2\pi\lambda_1}\right) \cdots \left(\sqrt{2\pi\lambda_M}\right) = \left(\sqrt{2\pi}\right)^M \sqrt{\det V}. \tag{7}$$

El determinante de V (también el determinante de Λ) es el producto $(\lambda_1) \ldots (\lambda_M)$ de los autovalores. Así (7) da el número correcto por el cual dividir para que $p(x_1, \ldots, x_M)$ en la ecuación (6) tenga integral $= 1$ como queríamos.

La media y la varianza de $p(\boldsymbol{x})$ también son integrales M-dimensionales. La misma idea de diagonalizar V por sus autovectores $\boldsymbol{q}_1$ a $\boldsymbol{q}_M$ e introducir $\boldsymbol{Y} = \boldsymbol{Q^{\mathrm{T}} X}$ permitirá hallar esas integrales :

Vector $\boldsymbol{m}$ de medias
$$\int \cdots \int \boldsymbol{x}\, p(\boldsymbol{x})\, \boldsymbol{dx} = (m_1, m_2, \ldots) = \boldsymbol{m} \tag{8}$$

Matriz de covarianza $\boldsymbol{V}$
$$\int \cdots \int (\boldsymbol{x} - \boldsymbol{m})\, p(\boldsymbol{x}) (\boldsymbol{x} - \boldsymbol{m})^{\mathrm{T}}\, \boldsymbol{dx} = \boldsymbol{V} \tag{9}$$

Conclusión: la fórmula (6) para la densidad de probabilidad $p(\boldsymbol{x})$ tiene todas las propiedades que queremos.

Mínimos cuadrados ponderados

En el Capítulo 4, los mínimos cuadrados aparecieron a partir de un sistema irresoluble $A\boldsymbol{x} = \boldsymbol{b}$. Elegimos $\widehat{\boldsymbol{x}}$ para minimizar el error $||\boldsymbol{b} - A\boldsymbol{x}||^2$. Eso nos llevó a la ecuación de mínimos cuadrados $A^{\mathrm{T}}A\widehat{\boldsymbol{x}} = A^{\mathrm{T}}\boldsymbol{b}$. El mejor $A\widehat{\boldsymbol{x}}$ es la proyección de $\boldsymbol{b}$ en el espacio columna de A. Pero ¿ es la distancia al cuadrado $E = ||\boldsymbol{b} - A\boldsymbol{x}||^2$ la mejor medida de error para minimizar ?

Si los errores de medición en $\boldsymbol{b}$ son variables aleatorias independientes, con media $m = 0$ y varianza $\sigma^2 = 1$ y con distribución normal, Gauss diría que **sí**: *use mínimos cuadrados*. Si los errores no son independientes o sus varianzas no son iguales, Gauss diría que **no**: *use mínimos cuadrados* ***ponderados***.

Esta sección mostrará que una buena medida de error es $\boldsymbol{E} = (\boldsymbol{b} - A\boldsymbol{x})^{\mathrm{T}}\boldsymbol{V^{-1}}(\boldsymbol{b} - A\boldsymbol{x})$. La ecuación para el mejor $\widehat{\boldsymbol{x}}$ usa la matriz V de covarianza de $\boldsymbol{b}$:

$$\boxed{\textbf{Mínimos cuadrados ponderados} \qquad \boldsymbol{A^{\mathrm{T}}V^{-1}A\widehat{x} = A^{\mathrm{T}}V^{-1}b.}} \tag{10}$$

En los ejemplos más importantes los m errores en $\boldsymbol{b}$ son *independientes*. Esos errores tienen varianzas $\sigma_1^2, \ldots, \sigma_m^2$. Por independencia, V es una matriz diagonal. Los pesos $1/\sigma_1^2, \ldots, 1/\sigma_m^2$ se obtienen de V^{-1}. *Ponderamos los errores en* $\boldsymbol{b}$ *para lograr* **varianza** $= \mathbf{1}$ (y **covarianza** $= \mathbf{0}$).

Mínimos cuadrados ponderados
Errores independientes en $\boldsymbol{b}$

$$\boxed{\text{Minimizar} \quad E = \sum_{i=1}^{m} \frac{(\boldsymbol{b} - A\boldsymbol{x})_i^2}{\sigma_i^2}} \tag{11}$$

Pesando los errores "blanqueamos" el ruido. El **ruido blanco** es una descripción rápida de errores independientes con distribución gaussiana estándar $\mathbf{N}(0, 1)$ con media cero y $\sigma^2 = 1$.

Escribamos los pasos que conducen a las ecuaciones (10) y (11) para el mejor $\widehat{\boldsymbol{x}}$:

Comenzar con $A\boldsymbol{x} = \boldsymbol{b}$ (m ecuaciones, n incógnitas, $m > n$, sin solución)

Cada lado derecho b_i tiene media cero y varianza σ_i^2. Los b_i son independientes

Dividir la i-ésima ecuación por σ_i para obtener varianza $= 1$ para cada b_i/σ_i

Esa división convierte a $A\boldsymbol{x} = \boldsymbol{b}$ en $V^{-1/2}A\boldsymbol{x} = V^{-1/2}\boldsymbol{b}$ con $V^{-1/2} = \operatorname{diag}(1/\sigma_1, \ldots, 1/\sigma_m)$

Los mínimos cuadrados usuales en esas ecuaciones ponderadas dan $A \to V^{-1/2}A$ y $\boldsymbol{b} \to V^{-1/2}\boldsymbol{b}$

$$\boxed{(V^{-1/2}A)^{\mathrm{T}}(V^{-1/2}A)\widehat{\boldsymbol{x}} = (V^{-1/2}A)^{\mathrm{T}}V^{-1/2}\,\boldsymbol{b} \quad \text{es} \quad \boldsymbol{A^{\mathrm{T}}V^{-1}A\widehat{x} = A^{\mathrm{T}}V^{-1}b.}} \tag{12}$$

Debido a los $1/\sigma^2$ en V^{-1}, las ecuaciones más confiables (σ *más pequeño*) tienen pesos más pesados. Esta es la característica principal de los mínimos cuadrados ponderados.

Esas ponderaciones en la diagonal (ecuaciones desacopladas) son las más frecuentes y las más sencillas. Se aplican cuando hay *errores independientes en* b_i. Cuando esos errores de medición no son independientes, V ya no es más diagonal, pero (12) sigue siendo la ecuación ponderada correcta.

En la práctica, encontrar todas las covarianzas puede ser un trabajo complejo. Es más sencillo con V diagonal.

La varianza del $\widehat{x}$ estimado

Un punto más: a menudo la pregunta importante no es encontrar el mejor $\widehat{\boldsymbol{x}}$ para un $\boldsymbol{b}$ específico. Esta es solo una muestra. El objetivo real es conocer la **fiabilidad de todo el experimento.** Esto se mide (como siempre se mide la fiabilidad) con la **varianza del estimador** $\widehat{x}$. Primero, media cero en $\boldsymbol{b}$ da media cero en $\widehat{\boldsymbol{x}}$. Así, la formula que conecta la varianza $\boldsymbol{V}$ de los elementos $\boldsymbol{b}$ a la varianza $\boldsymbol{W}$ de los resultados $\widehat{\boldsymbol{x}}$ resulta ser de gran belleza:

Matriz de varianza-covarianza de $\widehat{x}$ $\quad \boldsymbol{W} = \mathrm{E}[(\widehat{\boldsymbol{x}} - \boldsymbol{x})(\widehat{\boldsymbol{x}} - \boldsymbol{x})^{\mathrm{T}}] = \boldsymbol{(A^{\mathrm{T}}V^{-1}A)^{-1}}.$ (13)

La varianza más pequeña posible se obtiene de la mejor matriz de ponderación posible, que es V^{-1}.

Esta fórmula clave es una aplicación perfecta de la Sección V.4. **Si $\boldsymbol{b}$ tiene una matriz de covarianza V, entonces $\widehat{\boldsymbol{x}} = \boldsymbol{Lb}$ tiene una matriz de covarianza $\boldsymbol{LVL^{\mathrm{T}}}$.** La matriz L es entonces $(A^{\mathrm{T}}V^{-1}A)^{-1}A^{\mathrm{T}}V^{-1}$, porque $\widehat{\boldsymbol{x}} = \boldsymbol{Lb}$ resuelve la ecuación ponderada $(A^{\mathrm{T}}V^{-1}A)\,\widehat{\boldsymbol{x}} = A^{\mathrm{T}}V^{-1}\boldsymbol{b}$. Reemplazamos esto en LVL^{T} y podemos ver cómo aparece la ecuación (13):

$$\boldsymbol{LVL^{\mathrm{T}}} \;=\; (A^{\mathrm{T}}V^{-1}A)^{-1}A^{\mathrm{T}}V^{-1}\;\; V \;\; V^{-1}A\,(A^{\mathrm{T}}V^{-1}A)^{-1} \;=\; \boldsymbol{(A^{\mathrm{T}}V^{-1}A)^{-1}}.$$

Esta es la matriz de covarianza W de $\widehat{\boldsymbol{x}}$. Es momento de un ejemplo.

Ejemplo 1 Supongamos que un doctor mide su frecuencia cardíaca x tres veces $(m = 3, n = 1)$:

$$\begin{matrix} x = b_1 \\ x = b_2 \\ x = b_3 \end{matrix} \quad \text{es} \quad A\boldsymbol{x} = \boldsymbol{b} \quad \text{con} \quad A = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \text{y} \quad V = \begin{bmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{bmatrix}$$

Las varianzas podrían ser $\sigma_1^2 = 1/9$, $\sigma_2^2 = 1/4$ y $\sigma_3^2 = \mathbf{1}$. Los pesos son 3 luego 2 luego 1. Usted se va poniendo más nervioso a medida que le miden más veces la frecuencia cardíaca: b_3 es menos confiable que b_2 y b_1. Los tres resultados de las mediciones contienen algo de información, así todos se usan para obtener la mejor estimación (ponderada) $\widehat{\boldsymbol{x}}$:

$$V^{-1/2}A\widehat{\boldsymbol{x}} = V^{-1/2}\boldsymbol{b} \quad \text{es} \quad \begin{matrix} 3x = 3b_1 \\ 2x = 2b_2 \\ 1x = 1b_3 \end{matrix} \quad \text{que lleva a} \quad \boldsymbol{A^{\mathrm{T}}V^{-1}A\widehat{x} = A^{\mathrm{T}}V^{-1}b}$$

$$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 9 & & \\ & 4 & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \widehat{\boldsymbol{x}} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 9 & & \\ & 4 & \\ & & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$\boldsymbol{\widehat{x} = \frac{9b_1 + 4b_2 + b_3}{14}} \quad \textbf{es el mejor promedio ponderado de } \boldsymbol{b_1, b_2, b_3}$$

La mayor parte del peso está en b_1 ya que su varianza σ_1 es la menor. La varianza de $\widehat{\boldsymbol{x}}$ tiene la hermosa fórmula $\boldsymbol{W = (A^{\mathrm{T}} V^{-1} A)^{-1}}$. Esa varianza $W = \frac{1}{14}$ resultó menor que $\frac{1}{9}$ al incluir b_2 y b_3:

$$\textbf{Varianza de } \widehat{\boldsymbol{x}} \quad \left(\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 9 & & \\ & 4 & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right)^{-1} = \frac{1}{14} \quad \text{es menor que} \quad \frac{1}{9}$$

El teorema MELI de Gauss (demostrado en el sitio web) establece que nuestro $\widehat{\boldsymbol{x}} = L\boldsymbol{b}$ es el Mejor Estimador Lineal Insesgado (Best Linear Unbiased Estimate, BLUE) de la solución de $A\boldsymbol{x} = \boldsymbol{b}$. Para cualquier otra opción insesgada $\boldsymbol{x}^* = L^*\boldsymbol{b}$, la varianza $W^* = L^* V L^{*\mathrm{T}}$ será mayor que $W = LVL^{\mathrm{T}}$.
Observación: mayor significa que $W^* - W$ será semidefinida positiva. Insesgado significa que $L^*A = I$. Así un $A\boldsymbol{x} = \boldsymbol{b}$ exacto producirá la respuesta correcta $\boldsymbol{x} = L^*\boldsymbol{b} = L^*A\boldsymbol{x}$.

Debo agregar que existen motivos por los cuales no es bueno minimizar los errores cuadrados en primer lugar. Una razón: este $\widehat{\boldsymbol{x}}$ a menudo tiene muchas componentes pequeñas. Los cuadrados de números pequeños son muy pequeños y aparecen cuando minimizamos. Es más fácil darle sentido a los vectores *ralos*, con solo algunas componentes no nulas. Los estadísticos a menudo prefieren minimizar **errores no cuadráticos**: **la suma de** $|(\boldsymbol{b} - A\boldsymbol{x})_i|$. *Esta medida de error es* ℓ^1 *en lugar de* ℓ^2. Debido a los valores absolutos, la ecuación para $\widehat{\boldsymbol{x}}$ con la norma $\boldsymbol{\ell^1}$ se vuelve no lineal.

Algoritmos nuevos calculan con rapidez un $\widehat{\boldsymbol{x}}$ ralo y **el futuro le pertenece a** $\boldsymbol{\ell^1}$**.** La Sección III.5 de compressed sensing fue una aplicación increíble de regresión en $\boldsymbol{\ell^1}$.

El filtro de Kalman

El "filtro Kalman" es el gran algoritmo de mínimos cuadrados dinámicos. La palabra *dinámico* significa que las nuevas mediciones $\boldsymbol{b}_k$ continúan llegando. Así la mejor estimación $\widehat{\boldsymbol{x}}_k$ sigue cambiando (basada en todos los $\boldsymbol{b}_0, \ldots, \boldsymbol{b}_k$). Más que eso, la matriz A va cambiando. Así $\widehat{\boldsymbol{x}}_2$ será nuestra mejor estimación por mínimos cuadrados de la última solución de **toda la historia de las ecuaciones de observación y de actualización (ecuaciones de estado).** Hasta el segundo paso, existen 3 observaciones y 2 ecuaciones de estado:

$$A_0\boldsymbol{x}_0 = \boldsymbol{b}_0 \qquad \boldsymbol{x}_1 = F_0\boldsymbol{x}_0 \qquad A_1\boldsymbol{x}_1 = \boldsymbol{b}_1 \qquad \boldsymbol{x}_2 = F_1\boldsymbol{x}_1 \qquad A_2\boldsymbol{x}_2 = \boldsymbol{b}_2 \tag{14}$$

La idea de Kalman es introducir una ecuación a la vez. Cada ecuación tendrá errores. Con cada ecuación nueva, actualizamos la mejor estimación $\widehat{\boldsymbol{x}}_k$ del $\boldsymbol{x}_k$ actual. Pero no olvidamos la historia. Esta nueva estimación $\widehat{\boldsymbol{x}}_k$ usa todas las observaciones pasadas $\boldsymbol{b}_0$ a $\boldsymbol{b}_{k-1}$ y todas las ecuaciones de estado $\boldsymbol{x}_{\text{new}} = F_{\text{old}}\,\boldsymbol{x}_{\text{old}}$. Es un problema de mínimos cuadrados grande y creciente.

Otro punto importante sobre (14). Cada ecuación de mínimos cuadrados se **pondera** con la matriz de covarianza V_k del error en $\boldsymbol{b}_k$. Incluso hay una matriz de covarianza C_k para errores en las ecuaciones de actualización $\boldsymbol{x}_{k+1} = F_k x_k$. El mejor $\widehat{\boldsymbol{x}}_1$ entonces depende de $\boldsymbol{b}_0, \boldsymbol{b}_1, F_1, V_0, V_1$ y C_1. La manera correcta de escribir $\widehat{\boldsymbol{x}}_1$ es como una actualización del $\widehat{\boldsymbol{x}}_0$ previo.

Concentrémonos en un problema simplificado, sin las matrices F_k ni las covarianzas C_k. Estimamos el mismo $\boldsymbol{x}$ verdadero en todos los pasos. ¿ Cómo obtenemos $\widehat{\boldsymbol{x}}_1$ de $\widehat{\boldsymbol{x}}_0$?

VIEJO $A_0\,\boldsymbol{x}_0 = \boldsymbol{b}_0$ lleva a la ecuación ponderada $A_0^{\mathrm{T}}\,V_0^{-1}\,A_0\,\widehat{\boldsymbol{x}}_0 = A_0^{\mathrm{T}}\,V_0^{-1}\,\boldsymbol{b}_0$. (15)

NUEVO $\begin{bmatrix} A_0 \\ A_1 \end{bmatrix} \widehat{\boldsymbol{x}}_1 = \begin{bmatrix} \boldsymbol{b}_0 \\ \boldsymbol{b}_1 \end{bmatrix}$ lleva a la siguiente ecuación ponderada para $\widehat{\boldsymbol{x}}_{\mathbf{1}}$:

$$\begin{bmatrix} A_0^{\mathrm{T}} & A_1^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} V_0^{-1} & \\ & V_1^{-1} \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \end{bmatrix} \widehat{\boldsymbol{x}}_1 = \begin{bmatrix} A_0^{\mathrm{T}} & A_1^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} V_0^{-1} & \\ & V_1^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{b}_0 \\ \boldsymbol{b}_1 \end{bmatrix}. \tag{16}$$

Sí, podemos resolver ese nuevo problema y olvidar el viejo. Pero la solución vieja $\widehat{\boldsymbol{x}}_0$ requirió cálculos que esperamos poder reutilizar en $\widehat{\boldsymbol{x}}_1$. Lo que buscamos es **una actualización de $\widehat{\boldsymbol{x}}_0$** :

La actualización de Kalman da $\widehat{\boldsymbol{x}}_1$ a partir de $\widehat{\boldsymbol{x}}_0$ $\qquad \widehat{\boldsymbol{x}}_1 = \widehat{\boldsymbol{x}}_0 + K_1(\boldsymbol{b}_1 - A_1\,\widehat{\boldsymbol{x}}_0).$ (17)

La corrección de la actualización es $\boldsymbol{b}_1 - A_1\widehat{\boldsymbol{x}}_0$, la diferencia entre el estado viejo $\widehat{\boldsymbol{x}}_0$ y las mediciones nuevas $\boldsymbol{b}_1$ multiplicadas por la *matriz de ganancia de Kalman* K_1. La fórmula para K_1 se obtiene de la comparación de las soluciones $\widehat{\boldsymbol{x}}_1$ y $\widehat{\boldsymbol{x}}_0$ de (15) y (16). Cuando actualizamos $\widehat{\boldsymbol{x}}_0$ a $\widehat{\boldsymbol{x}}_1$ basados en datos nuevos $\boldsymbol{b}_1$, **también actualizamos la matriz de covarianza $\boldsymbol{W_0}$ a $\boldsymbol{W_1}$.**

Recordemos $W_0 = (A_0^{\mathrm{T}}\,V_0^{-1}\,A_0)^{-1}$ de la ecuación (13). Actualicemos su inversa de W_0^{-1} a W_1^{-1}:

Covarianza $\boldsymbol{W_1}$ de errores en $\widehat{\boldsymbol{x}}_1$ $\qquad \boldsymbol{W_1^{-1} = W_0^{-1} + A_1^{\mathrm{T}}\,V_1^{-1}\,A_1}$ (18)

Matriz de ganancia de Kalman $\boldsymbol{K_1}$ $\qquad \boldsymbol{K_1 = W_1\,A_1^{\mathrm{T}}\,V_1^{-1}}$ (19)

Esta es la esencia del filtro de Kalman. Observemos la importancia de las matrices de covarianza W_k. Esas matrices miden la fiabilidad de todo el proceso, y el vector $\widehat{\boldsymbol{x}}_k$ estima el estado actual basado en las medidas particulares $\boldsymbol{b}_0$ a $\boldsymbol{b}_k$.

Se han escrito capítulos y libros enteros para explicar el filtro de Kalman dinámico, cuando los estados $\boldsymbol{x}_k$ también son cambiantes (basados en las matrices F_k). Se *predice* $\boldsymbol{x}_k$ con F, y le sigue una *corrección* en base a los datos nuevos $\boldsymbol{b}$. Tal vez sea mejor deternernos aquí.

Esta página trató sobre **mínimos cuadrados recursivos**: se suman datos nuevos $\boldsymbol{b_k}$ y se actualiza la mejor estimación $\widehat{\boldsymbol{x}}_k$ usando todos los datos y se actualiza su matriz de covarianza W_k. La idea de actualizar se originó a partir de la fórmula de Sherman-Morrison-Woodbury para $(A - UV^{\mathrm{T}})^{-1}$ de la Sección III.1. Numéricamente esa es la clave del éxito de Kalman, intercambiar matrices inversas de tamaño n por matrices inversas de tamaño k.

Problemas de la Sección V.5

1 Dos mediciones de la misma variable x dan dos ecuaciones $x = b_1$ y $x = b_2$. Supongamos que las medias son cero y las varianzas son σ_1^2 y σ_2^2, con errores independientes: V es diagonal con elementos σ_1^2 y σ_2^2. Escriba las dos ecuaciones de la forma $A\boldsymbol{x} = \boldsymbol{b}$ (A es de 2 por 1). Como en el Ejemplo 1 del texto, encuentre la mejor estimación $\widehat{\boldsymbol{x}}$ basada en b_1 y b_2:

$$\widehat{\boldsymbol{x}} = \frac{b_1/\sigma_1^2 + b_2/\sigma_2^2}{1/\sigma_1^2 + 1/\sigma_2^2} \qquad \mathrm{E}\left[\widehat{\boldsymbol{x}}\,\widehat{\boldsymbol{x}}^{\mathrm{T}}\right] = \left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}\right)^{-1}.$$

2 (a) En el Problema 1, suponga que la segunda medición b_2 se vuelva superexacta y su varianza $\sigma_2 \to 0$. ¿Cuál es la mejor estimación $\widehat{\boldsymbol{x}}$ cuando σ_2 es cero?

(b) El caso opuesto tiene $\sigma_2 \to \infty$ y ninguna información en b_2. ¿Cuál es ahora la mejor estimación $\widehat{\boldsymbol{x}}$ basada en b_1 y b_2?

3 Si x e y son independientes con probabilidades $p_1(x)$ y $p_2(y)$, entonces $p(x,y) = p_1(x)\,p_2(y)$. Separando las integrales dobles en productos de integrales simples ($-\infty$ a ∞) muestre que $\iint p(x,y)\,dx\,dy = \mathbf{1}$ y $\iint (x+y)\,p(x,y)\,dx\,dy = \boldsymbol{m_1 + m_2}$.

4 Continúe el Problema 3 considerando x, y independientes para mostrar que $p(x,y) = p_1(x)\,p_2(y)$ implica

$$\iint (x - m_1)^2\,p(x,y)\,dx\,dy = \boldsymbol{\sigma_1^2} \qquad \iint (x - m_1)(y - m_2)\,p(x,y)\,dx\,dy = \mathbf{0}.$$

Así la matriz de covarianza V de 2 por 2 es diagonal y sus elementos son _____.

5 Sea $\widehat{x}_k$ el promedio de $b_1, \ldots, b_k$. Llega una nueva medición b_{k+1}. La ecuación de actualización de Kalman (17) da el nuevo promedio $\widehat{x}_{k+1}$:

$$\textbf{Verifique que} \quad \widehat{x}_{k+1} = \widehat{x}_k + \frac{1}{k+1}\,(b_{k+1} - \widehat{x}_k) \quad \textit{es el promedio de} \quad b_1, \ldots, b_{k+1}.$$

También verifique la ecuación de actualización (18) para la varianza $W_{k+1} = \sigma^2/(k+1)$ de este promedio $\widehat{x}$, suponiendo que $W_k = \sigma^2/k$ y que b_{k+1} tiene varianza $V = \sigma^2$.

6 (**Modelo estable**) El Problema 5 era mínimos cuadrados *estáticos*. Todos los promedios muestrales $\widehat{x}_k$ eran estimaciones del mismo x. Para que el filtro de Kalman sea *dinámico* incluya también una *ecuación de estado* $x_{k+1} = Fx_k$ *con su propia varianza de error* s^2. El problema de mínimos cuadrados dinámico permite que x "se desplace" a medida que k aumenta:

$$\begin{bmatrix} 1 & \\ -F & 1 \\ & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} b_0 \\ 0 \\ b_1 \end{bmatrix} \text{ con varianzas } \begin{bmatrix} \sigma^2 \\ s^2 \\ \sigma^2 \end{bmatrix}.$$

Con $F = 1$, divida ambos lados de esas tres ecuaciones por σ, s, y σ. Encuentre $\widehat{x_0}$ y $\widehat{x_1}$ usando mínimos cuadrados, notando que le da más peso al b_1 reciente. El filtro de Kalman se desarrolla en *Algorithms for Global Positioning* (Borre y Strang).

V.6. Cadenas de Markov

¡Observemos las características clave sobre las cadenas de Markov usando un ejemplo sobre autos de alquiler ! Comenzamos con 100 autos en Chicago. Todos los meses, los autos se desplazan entre Chicago y Denver.

80 % de los autos de Chicago se quedan en Chicago **30 %** de los autos de Denver van a Chicago
20 % de los autos de Chicago van a Denver **70 %** de los autos de Denver se quedan en Denver

En lenguaje matricial, el desplazamiento de autos del mes n al $n+1$ está dado por $\boldsymbol{y_{n+1} = Py_n}$:

$$\boldsymbol{y_{n+1}} = \begin{bmatrix} \textbf{Autos de Chicago} \\ \textbf{Autos de Denver} \end{bmatrix}_{\boldsymbol{n+1}} = \begin{bmatrix} \mathbf{0.8} & \mathbf{0.3} \\ \mathbf{0.2} & \mathbf{0.7} \end{bmatrix} \begin{bmatrix} \textbf{Autos de Chicago} \\ \textbf{Autos de Denver} \end{bmatrix}_{\boldsymbol{n}} = \boldsymbol{Py_n} \tag{1}$$

Todos los meses multiplicamos por esa **"matriz de Markov"** $\boldsymbol{P}$. Ambas columnas suman 1. Después de n meses, la distribución de autos es $\boldsymbol{y_n = P^n y_0}$. En nuestro ejemplo $y_0 = (100, 0)$ ya que todos los autos comienzan en Chicago :

$$\boldsymbol{y_0} = \begin{bmatrix} 100 \\ 0 \end{bmatrix} \quad \boldsymbol{y_1} = \begin{bmatrix} 80 \\ 20 \end{bmatrix} \quad \boldsymbol{y_2} = \begin{bmatrix} 70 \\ 30 \end{bmatrix} \quad \boldsymbol{y_3} = \begin{bmatrix} 65 \\ 35 \end{bmatrix} \cdots \quad \boldsymbol{y_\infty} = \begin{bmatrix} \mathbf{60} \\ \mathbf{40} \end{bmatrix}.$$

Supongamos que 100 autos comienzan en Denver en lugar de Chicago :

$$\boldsymbol{y_0} = \begin{bmatrix} 0 \\ 100 \end{bmatrix} \quad \boldsymbol{y_1} = \begin{bmatrix} 30 \\ 70 \end{bmatrix} \quad \boldsymbol{y_2} = \begin{bmatrix} 45 \\ 55 \end{bmatrix} \quad \boldsymbol{y_3} = \begin{bmatrix} 52.5 \\ 47.5 \end{bmatrix} \cdots \quad \boldsymbol{y_\infty} = \begin{bmatrix} \mathbf{60} \\ \mathbf{40} \end{bmatrix}.$$

Ambas maneras llevan a la misma distribución límite 60-40. No importa dónde se encuentren los autos al comienzo. Ya que utilizamos las potencias P^n de la matriz P, este es un problema para estudiar usando los autovalores y autovectores de P :

$$\textbf{Autovalores} \quad \det \begin{bmatrix} .8-\lambda & .3 \\ .2 & .7-\lambda \end{bmatrix} = \lambda^2 - 1.5\lambda + 0.5 = (\lambda - \mathbf{1})(\lambda - \mathbf{0.5}) \quad \begin{matrix} \lambda = \mathbf{1} \\ \lambda = \mathbf{0.5} \end{matrix}$$

$$\textbf{Autovectores} \quad \begin{bmatrix} 0.8 & 0.3 \\ 0.2 & 0.7 \end{bmatrix} \begin{bmatrix} .60 \\ .40 \end{bmatrix} = \begin{bmatrix} \mathbf{.60} \\ \mathbf{.40} \end{bmatrix} \qquad \begin{bmatrix} 0.8 & 0.3 \\ 0.2 & 0.7 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \mathbf{1} \\ \mathbf{-1} \end{bmatrix}$$

Esos explican todo. La distribución límite 60-40 de autos es el **estado estacionario** : **autovalor $\boldsymbol{\lambda_1 = 1}$**. Así $\boldsymbol{y}_n = (60, 40)$ da $\boldsymbol{y}_{n+1} = (60, 40)$. Entonces $\lambda_2 = \frac{1}{2}$ significa que :

Cada mes la distancia al estado estacionario se multiplica por $\frac{1}{2}$.

Vemos que en los números mencionados arriba $100, 80, 70, 65$ correspondientes a Chicago se multiplicó la distancia a 60 (estado estacionario) por $\frac{1}{2}$ todos los meses. De manera similar en los números $0, 20, 30, 35$ correspondientes a Denver se divide por dos la distancia a 40 (estado estacionario). En notación matricial, P se diagonaliza como $X\Lambda X^{-1}$ usando sus autovectores y autovalores. Entonces $\boldsymbol{P^n} = (X\Lambda X^{-1}) \ldots (X\Lambda X^{-1}) = \boldsymbol{X\Lambda^n X^{-1}}$:

$$\boldsymbol{P^n} = X\Lambda^n X^{-1} = \begin{bmatrix} .6 & 1 \\ .4 & -1 \end{bmatrix} \begin{bmatrix} 1^n & \\ & \left(\frac{1}{2}\right)^n \end{bmatrix} \begin{bmatrix} 1 & 1 \\ .4 & -.6 \end{bmatrix} = \begin{bmatrix} \mathbf{.6} & \mathbf{.6} \\ \mathbf{.4} & \mathbf{.4} \end{bmatrix} + \left(\frac{\mathbf{1}}{\mathbf{2}}\right)^{\boldsymbol{n}} \begin{bmatrix} \mathbf{.4} & \mathbf{-.6} \\ \mathbf{-.4} & \mathbf{.6} \end{bmatrix}$$

Para $n = 1$ tenemos P. Para $n = \infty$ la matriz límite P^∞ tiene .6, .4 en ambas columnas. Los 100 autos pueden empezar en la columna de Chicago o en la columna de Denver, siempre se repartirán 60-40 cuando $n \to \infty$. Esta es la característica destacable de las matrices de Markov positivas como lo es la matriz P.

Los requisitos para una **matriz de Markov positiva** son:

Todo $p_{ij} > 0$ y cada columna de P suma 1 (no se pierde ningún auto). Así $\mathbf{1}^{\mathrm{T}} P = \mathbf{1}^{\mathrm{T}}$

La matriz P tiene $\lambda_1 = 1$ (mayor autovalor) y $\boldsymbol{x}_1 > 0$ (autovector positivo).

Es el teorema de Perron-Frobenius para matrices positivas el que garantiza $\lambda_1 > 0$ y $\boldsymbol{x}_1 > 0$. Así, el hecho de que las columnas sumen 1 nos dice que $P^{\mathrm{T}}\mathbf{1} = \mathbf{1}$ y $\lambda_1 = 1$. Esto produce el estado estacionario $\boldsymbol{y}_\infty$ como un múltiplo de $\boldsymbol{x}_1$. Recordemos que la fila 1 de X^{-1} es el autovector a izquierda $[\,\mathbf{1}\ \ \mathbf{1}\ \ \ldots\ \ \mathbf{1}\,]$:

$$\textbf{Convergencia}\quad P^n = X\Lambda^n X^{-1} = \begin{bmatrix} & & \\ \boldsymbol{x}_1 & \boldsymbol{x}_2 \cdots & \boldsymbol{x}_n \\ & & \end{bmatrix} \begin{bmatrix} 1 & & \\ & \lambda_2^n & \\ & & \ddots \end{bmatrix} \begin{bmatrix} & & \\ & X^{-1} & \\ & & \end{bmatrix} \to \begin{bmatrix} & & \\ \boldsymbol{x}_1 & \boldsymbol{x}_1 \cdots & \boldsymbol{x}_1 \\ & & \end{bmatrix}$$

A medida que $n \to \infty$, solo el 1 en esa matriz diagonal Λ^n sobrevivirá. Las columnas multiplicadas por las filas se vuelven **columna $\boldsymbol{x}_1$ de $\boldsymbol{X}$ multiplicada por** $[\,\mathbf{1}\ \ \mathbf{1}\ \ \ldots\ \ \mathbf{1}\,]$. ¡La matriz límite P^∞ tiene $\boldsymbol{x}_1$ en cada columna! El estado estacionario $\boldsymbol{y}_\infty = P^\infty \boldsymbol{y}_0$ tiene que ser un múltiplo del vector columna $\boldsymbol{x}_1$. El múltiplo era $(60, 40)$ en nuestro ejemplo porque comenzamos con 100 autos. Una cadena de Markov no destruye los autos viejos ni añade nuevos, eventualmente los distribuye de acuerdo al autovector principal $\boldsymbol{x}_1$ de P.

Consideremos ahora P como una **matriz de probabilidades**. Luego viene Perron-Frobenius.

Probabilidades de transición

Las cadenas de Markov son ejemplos perfectos de álgebra lineal dentro de la teoría de probabilidad. Los números fundamentales son las probabilidades p_{ij} de moverse del estado j en el tiempo n al estado i en el tiempo $n + 1$:

Probabilidad de transición p_{ij} = Probabilidad de que $x(n+1) = i$ si $x(n) = j$. (2)

Hay dos puntos clave ocultos en esa sencilla afirmación. Primero, la probabilidad p_{ij} no depende de n. Las reglas se mantienen iguales en todo momento. Segundo, *las probabilidades $\boldsymbol{y}_{n+1}$ para el nuevo estado $x(n+1)$ dependen solo del estado actual $x(n)$*, no de la historia previa.

Todavía nos queda una pregunta por responder: ¿Cuáles son los "estados" posibles de la cadena de Markov? En el ejemplo, los estados son Chicago y Denver (*técnicamente son ciudades*). He aquí tres opciones:

Cadena de Markov finita	Cada estado $x(n)$ es uno de los números $1, 2, \ldots, N$
Cadena de Markov de infinitos estados	Cada estado $x(n)$ es un entero $n = 0, 1, 2, \ldots$
Cadena de Markov continua	Cada estado $x(n)$ es un número real.

Consideraremos principalmente cadenas finitas con N estados posibles. El estado inicial $x(0)$ podría estar dado. O podemos solo conocer el **vector y_0 de probabilidades** para ese estado inicial. En contraste con las ecuaciones diferenciales, $x(0)$ *no determina* $x(1)$. Si $x(0) = j$, eso solo determina las N probabilidades en y_1 para el nuevo estado $x(1)$. Ese nuevo estado es un número del 1 al N.

Las probabilidades para esos estados nuevos son los números $p_{1j}, p_{2j}, \ldots, p_{Nj}$. **Esas probabilidades deben sumar 1** :

$$\textbf{Columna } j \textbf{ de } P \qquad p_{1j} + p_{2j} + \cdots + p_{Nj} = 1. \tag{3}$$

Esos números van naturalmente en una matriz de P = matriz de probabilidades p_{ij} de N por N. Esas son las probabilidades de transición y **P es la matriz de transición**. Nos dice todo lo que podemos saber (solo probabilidades, no hechos) sobre la transición del estado $x(0) = j$ al estado $x(1) = i$. Y esta misma matriz P se aplica a la transición de $x(n)$ a $x(n+1)$ en todo tiempo futuro.

Matriz de transición P
$y_{n+1} = Py_n$

$$y_{n+1} = \begin{bmatrix} \text{Prob}\{x(n+1)=1\} \\ \vdots \\ \text{Prob}\{x(n+1)=N\} \end{bmatrix} = \begin{bmatrix} p_{11} & \cdots & p_{1N} \\ \vdots & & \vdots \\ p_{N1} & \cdots & p_{NN} \end{bmatrix} \begin{bmatrix} \text{Prob}\{x(n)=1\} \\ \vdots \\ \text{Prob}\{x(n)=N\} \end{bmatrix} \tag{4}$$

Todos los elementos de la matriz P satisfacen $0 \le p_{ij} \le 1$. De la ecuación (3), vemos que cada columna suma 1 :

$$\mathbf{1}^{\mathrm{T}} = \textbf{vector fila de } N \textbf{ unos} \quad \mathbf{1}^{\mathrm{T}} P = \mathbf{1}^{\mathrm{T}} \text{ y } P^{\mathrm{T}}\mathbf{1} = \mathbf{1} \quad \mathbf{1} = \textbf{vector columna de } N \textbf{ unos}$$

Así P^{T} es una matriz no negativa con autovalor $\lambda = 1$ y autovector $\mathbf{1} = (1, 1, \ldots, 1)$. Y P también es una matriz no negativa con autovalor $\lambda = 1$. Pero debemos encontrar el autovector que satisface $Pv = v$:

Ejemplo 1 $\quad P = \begin{bmatrix} \mathbf{0.8} & \mathbf{0.3} \\ \mathbf{0.2} & \mathbf{0.7} \end{bmatrix} \quad \begin{bmatrix} 1 & 1 \end{bmatrix} P = \begin{bmatrix} 1 & 1 \end{bmatrix} \quad Pv = \begin{bmatrix} 0.8 & 0.3 \\ 0.2 & 0.7 \end{bmatrix} \begin{bmatrix} \mathbf{0.6} \\ \mathbf{0.4} \end{bmatrix} = v$

Por lo tanto $v = (0.6, 0.4)$. La traza es $0.8 + 0.7 = 1.5$, y luego el segundo autovalor es $\lambda_2 = \mathbf{0.5}$. El segundo autovector v_2 de P siempre es ortogonal al primer autovector $(1, 1)$ de P^{T}.

Regresemos a la ecuación de transición para las probabilidades $y_{n+1} = Py_n$. Esos vectores y_n e y_{n+1} contienen, a tiempo n y a tiempo $n+1$, las probabilidades para los N estados diferentes. En todo tiempo, **las columnas de P^n suman 1** (al igual que las probabilidades en y_n) :

$$\mathbf{1}^{\mathrm{T}} y_{n+1} = \mathbf{1}^{\mathrm{T}}(Py_n) = (\mathbf{1}^{\mathrm{T}}P)y_n = \mathbf{1}^{\mathrm{T}} y_n = \mathbf{1}. \tag{5}$$

He aquí la pregunta fundamental para una cadena de Markov. La matriz de transición P es fija y conocida. El estado inicial $x(0)$ o el vector y_0 de probabilidades para ese estado puede ser conocido o desconocido. Pregunta clave : **¿ Tienen los vectores de probabilidades $y_n = P^n y_0$ un límite y_∞ cuando $n \to \infty$?** Esperamos que y_∞ sea independiente de las probabilidades iniciales en y_0. Veremos que y_∞ a menudo existe, pero no para toda P. Cuando existe, **y_∞ nos dice con qué frecuencia esperamos que esté en cada estado**.

Ejemplo 2 La matriz de transición puede ser $P = \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} \end{bmatrix}$ = **matriz alternante.**

Esto significa que el sistema cambia su estado en cada paso. El estado 1 en el tiempo n lleva al estado 2 en el tiempo $n+1$. Si las probabilidades iniciales estaban en $\boldsymbol{y}_0 = \left(\frac{1}{3}, \frac{2}{3}\right)$ entonces $\boldsymbol{y}_1 = P\boldsymbol{y}_0 = \left(\frac{2}{3}, \frac{1}{3}\right)$. Las probabilidades van y vienen entre esos dos vectores. *No hay estado estacionario.*

Nuestras matrices P siempre tienen como autovalor a $\lambda = 1$. Su autovector sería un estado estacionario (nada cambia cuando multiplicamos por P). Pero esta P en particular también tiene el autovalor $\lambda = -1$. Su efecto no desaparecerá a medida que $n \to \infty$. Lo único estacionario que se podrá observar está en $\boldsymbol{y}_0 = \boldsymbol{y}_2 = \boldsymbol{y}_4 \ldots$ y por separado en $\boldsymbol{y}_1 = \boldsymbol{y}_3 = \boldsymbol{y}_5 \ldots$ Las potencias de esta matriz P oscilan entre P e I.

Otras matrices P tienen un estado estacionario: $\boldsymbol{P^n \to P^\infty}$ e $\boldsymbol{y}_n \to \boldsymbol{y}_\infty$. Observe que los estados actuales x continúan cambiando, segun las probabilidades p_{ij} en P. **El vector $\boldsymbol{y}_\infty$ nos dice la fracción $(\boldsymbol{y}_{1\infty}, \ldots, \boldsymbol{y}_{N\infty})$ de tiempo en que el sistema estará eventualmente en cada estado.**

P positiva o P no negativa

Hay una diferencia clara entre nuestros dos ejemplos: $P_1 > 0$ y $P_2 \geq 0$.

$P_1 = \begin{bmatrix} 0.8 & 0.3 \\ 0.2 & 0.7 \end{bmatrix}$ tiene autovalores $\mathbf{1}$ y $\frac{\mathbf{1}}{\mathbf{2}}$. Las potencias $\left(\frac{1}{2}\right)^n$ tienden a cero.

$P_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ tiene autovalores $\mathbf{1}$ y $\mathbf{-1}$. Las potencias $(-1)^n$ no tienden a cero.

Toda columna de P suma 1. Así $\lambda = 1$ es un autovalor, porque las filas de $P - I$ suman cero. *Y ningún elemento de P es negativo.* Las dos propiedades juntas aseguran que ningún autovalor verifica $|\lambda| > 1$. Pero hay una diferencia importante entre P_1 y P_2.

$\boldsymbol{P_2}$ tiene elementos nulos. Esto abre la posibilidad de que la magnitud de λ_2 pueda ser 1.

$\boldsymbol{P_1}$ tiene elementos estrictamente positivos. Esto garantiza que la magnitud de λ_2 verifica $|\lambda_2| < 1$.

El teorema de Perron-Frobenius para $\boldsymbol{P_1 > 0}$ (elementos estrictamente positivos) garantiza el éxito:

1. **El mayor autovalor $\boldsymbol{\lambda_1}$ de $\boldsymbol{P}$ y su autovector $\boldsymbol{v_1}$ son estrictamente positivos.**
2. **Todos los demás autovalores $\boldsymbol{\lambda_2, \ldots, \lambda_N}$ cumplen $\boldsymbol{|\lambda| < \lambda_1}$. El mayor autovalor de las matrices de Markov es $\boldsymbol{\lambda_1 = 1}$.**

Un tercer ejemplo P_3 muestra que los ceros en P no siempre arruinan la aproximación al estado estacionario:

Ejemplo 3 $\boldsymbol{P_3} = \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix}$ tiene $\lambda = 1, \frac{1}{2}$ con $\boldsymbol{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ y $\boldsymbol{v}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$.

Incluso con ese cero en P_3, todas las columnas de $(P_3)^n$ tienden a $\boldsymbol{v}_1$ = primer autovector:

$$(\boldsymbol{P_3})^{\boldsymbol{n}} = \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix}^n = \begin{bmatrix} 1 & 1 - \left(\frac{1}{2}\right)^n \\ 0 & \left(\frac{1}{2}\right)^n \end{bmatrix} \to \begin{bmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{6}$$

Este es el estado estacionario que queremos y esperamos. Así $\boldsymbol{y}_n = (P_3)^n \boldsymbol{y}_0$ aproxima el mismo autovector $\boldsymbol{v}_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}$. Esta cadena de Markov mueve todo al estado 1 a medida que $n \to \infty$.

Convergencia al estado estacionario a medida que $n \to \infty$

Para matrices de Markov estrictamente positivas la mejor manera de ver la convergencia $P^n \to P^\infty$ es diagonalizando P. *Supongamos por ahora* que p tiene n autovectores independientes. La matriz de autovalores Λ comienza con $\lambda_1 = 1$. Su matriz de autovectores X tiene el autovector principal $\boldsymbol{v}_1$ en su primera columna. A medida que n crece, ese autovector $\boldsymbol{v}_1$ irá apareciendo en cada columna de P^n.

$$P = X\Lambda X^{-1} \text{ implica que } P^n = (X\Lambda X^{-1})\dots(X\Lambda X^{-1}) = \boldsymbol{X\Lambda^n X^{-1}}$$

Las columnas de X son los autovectores $\boldsymbol{v}_1, \dots, \boldsymbol{v}_n$ de P. Las filas de X^{-1} son los autovectores de P^{T} (comenzando por el vector de todos unos $\mathbf{1}^{\mathrm{T}}$). Debido a que $\lambda_1 = 1$ y todos los demás autovalores verifican $|\lambda| < 1$, la matriz diagonal Λ^n aproximará Λ^∞ con un solo uno "1" en la esquina superior izquierda:

$$\boldsymbol{P^n = X\Lambda^n X^{-1}} \text{ se aproxima a } \boldsymbol{P^\infty} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots \end{bmatrix} \begin{bmatrix} \mathbf{1} & & \\ & \mathbf{0} & \\ & & \mathbf{0} \\ & & & \ddots \end{bmatrix} \begin{bmatrix} \mathbf{1}^{\mathrm{T}} \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_1 & \cdots \end{bmatrix}$$

He aquí $P, P^2, P^3, \dots$ convergiendo a la matriz de rango uno $P^\infty = \boldsymbol{v}_1\mathbf{1}^{\mathrm{T}}$ con $\boldsymbol{v}_1$ en todas las columnas:

$$\begin{bmatrix} .80 & .30 \\ .20 & .70 \end{bmatrix} \quad \begin{bmatrix} .70 & .45 \\ .30 & .55 \end{bmatrix} \quad \begin{bmatrix} .65 & .525 \\ .35 & .475 \end{bmatrix} \text{ se aproximan a } \begin{bmatrix} \mathbf{.60} & \mathbf{.60} \\ \mathbf{.40} & \mathbf{.40} \end{bmatrix} = \begin{bmatrix} \mathbf{.6} \\ \mathbf{.4} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{1} \end{bmatrix}$$

A continuación veremos el teorema de Perron-Frobenius. Demostraremos la parte de Perron (matrices estrictamente positivas). Frobenius permite ceros en P. Esto permite la *posibilidad* de que $|\lambda_2|$ sea igual a λ_1. En ese caso P^n no convergerá (a menos que $P = I$) a la $P^\infty = \boldsymbol{v}_1\mathbf{1}^{\mathrm{T}}$ usual.

Teorema de Perron-Frobenius

Un teorema matricial domina este tema. El teorema de Perron-Frobenius se aplica cuando todos los $a_{ij} \geq 0$. No hay un requisito de que todas las columnas sumen 1. Haremos la demostración más prolija que sirve cuando todo $a_{ij} > 0$. Así el mayor autovalor $\lambda_{\text{máx}}$ y también su autovector $\boldsymbol{x}$ son positivos.

Perron-Frobenius para $A > 0$
Todos los números que aparecen en $A\boldsymbol{x} = \lambda_{\max}\boldsymbol{x}$ son estrictamente positivos.

Prueba Comencemos con $A > 0$. La idea clave es mirar todos los números t tal que $A\boldsymbol{x} \geq t\boldsymbol{x}$ para algún vector no negativo $\boldsymbol{x}$ (aparte de $\boldsymbol{x} = \mathbf{0}$). Permitimos la desigualdad en $A\boldsymbol{x} \geq t\boldsymbol{x}$ para poder tener muchos candidatos t positivos pequeños. Para el **mayor valor $t_{\max}$** (que se alcanza), mostraremos que ***se cumple la igualdad***: $A\boldsymbol{x} = t_{\text{máx}}\,\boldsymbol{x}$. Así $t_{\text{máx}}$ es nuestro autovalor $\lambda_{\text{máx}}$ y $\boldsymbol{x}$ es el autovector. Lo demostraremos ahora.

Si $A\boldsymbol{x} \geq t_{\text{máx}}\boldsymbol{x}$ no es una igualdad, multiplicamos ambos lados por A. Como $A > 0$, esto produce una desigualdad estricta $A^2\boldsymbol{x} > t_{\text{máx}}A\boldsymbol{x}$. Por lo tanto, el vector positivo $\boldsymbol{y} = A\boldsymbol{x}$ satisface $A\boldsymbol{y} > t_{\text{máx}}\,\boldsymbol{y}$. Esto significa que $t_{\text{máx}}$ podría crecer. Esta contradicción fuerza la igualdad $A\boldsymbol{x} = t_{\text{máx}}\boldsymbol{x}$, y *tenemos así un autovalor*. Su autovector $\boldsymbol{x}$ es positivo porque del lado izquierdo de esa igualdad, $A\boldsymbol{x}$, sin duda será positivo.

Para ver que ningún autovalor puede ser más grande que $t_{\text{máx}}$, supongamos que $A\boldsymbol{z} = \lambda\boldsymbol{z}$. Ya que λ y $\boldsymbol{z}$ pueden tener números negativos o complejos, tomamos valores absolutos: $|\lambda||\boldsymbol{z}| = |A\boldsymbol{z}| \leq A|\boldsymbol{z}|$ por la "desigualdad triangular." Este $|\boldsymbol{z}|$ es un vector no negativo, entonces este $|\lambda|$ es uno de los candidatos t posibles. Por lo tanto, $|\lambda|$ no puede exceder $t_{\text{máx}}$, que debe ser $\lambda_{\text{máx}}$.

Sutilezas de la teoría de Markov

Regresamos a Markov, donde dejamos dos casos sin resolver al probar $P^n \to P^\infty = [\boldsymbol{v}_1\ \boldsymbol{v}_1 \ldots \boldsymbol{v}_1]$.

1. $P > 0$ podría ser estrictamente positiva, pero puede no tener n autovectores independientes.
2. $P \geq 0$ podría tener elementos nulos. Así $|\lambda_2| = 1$ se vuelve una posibilidad en lugar de una certeza.

El caso 1 es un problema técnico. El dato importante $P^n \to P^\infty$ sigue siendo cierto aunque no tengamos una matriz de autovectores invertible X. Tenemos una separación entre $\lambda_1 = 1$ y todos los otros autovalores. Siempre y cuando el autovector con $P\boldsymbol{v}_1 = \boldsymbol{v}_1$ vaya en la primera columna de X, la primera columna de $X^{-1}PX$ seguirá siendo $(1, 0, \ldots, 0)$. La submatriz A en las últimas filas y columnas de $X^{-1}PX$ tiene los otros autovalores de P. Todos satisfacen $|\lambda| < 1$ según Perron-Frobenius. **Demostraremos que $\boldsymbol{A^n \to 0}$.**

$X^{-1}PX$ siempre puede hacerse triangular (a menudo diagonal).

Si $|\lambda_2| < 1$ entonces $P^n \to P^\infty$

Queremos demostrar que $\boldsymbol{P^n \to P^\infty = [\ v_1\ \ v_1\ \ldots\ v_1\ ]}$ **cuando** $\boldsymbol{|\lambda_2| < 1}$. La matriz $P \geq 0$ podría contener ceros, como en el ejemplo 3. La matriz P puede no tener n autovectores independientes, de modo que no podemos diagonalizar P. Pero podemos separar $\lambda_1 = 1$ del resto de la matriz que tiene $|\lambda| < 1$. Esto será suficiente para demostrar que P^n se aproxima a P^∞:

$$X^{-1}PX = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & A \end{bmatrix} \text{ y los autovalores de } A \text{ satisfacen } |\lambda_2| < 1, \ldots, |\lambda_n| < 1.$$

Al aislar esa matriz A, obtenemos un resultado claro con muchas aplicaciones.

$$\boldsymbol{P^n} = X(X^{-1}PX)^nX^{-1} = \boldsymbol{X}\begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{A^n} \end{bmatrix}\boldsymbol{X^{-1}} \text{ converge a } \boldsymbol{v}_1\mathbf{1}^{\mathrm{T}} \text{ y } \boldsymbol{A^n \to 0}.$$

Si todos los autovalores de $\boldsymbol{A}$ satisfacen $\boldsymbol{|\lambda| < 1}$, entonces $\boldsymbol{A^n \to 0}$ cuando $\boldsymbol{n \to \infty}$

Paso 1 Encontrar una matriz triangular superior $S = M^{-1}AM$ con norma pequeña $\boldsymbol{||S|| < 1}$.

Paso 2 Entonces $A^n = (MSM^{-1})^n = MS^nM^{-1}$ satisfacen $||A^n|| \leq ||M||\,||S||^n\,||M^{-1}|| \to \mathbf{0}$.

Para completar la prueba necesitamos encontrar esa matriz triangular S. Ya que $S = M^{-1}AM$ es semejante a A, S tiene los mismos autovalores que A. Pero los autovalores de una matriz triangular se encuentran en su diagonal principal:

$$S = \begin{bmatrix} \lambda_2 & a & b \\ 0 & \lambda_3 & c \\ 0 & 0 & \lambda_4 \end{bmatrix} \text{ satisface } ||S|| < 1 \text{ si } a, b, c \text{ son muy pequeños}$$

Punto clave: ¡el autovalor más grande no es una norma! Si a, b, c son grandes, entonces las potencias S^2, S^3, S^4 comenzarán a crecer. Eventualmente $\lambda_2^n, \lambda_3^n, \lambda_4^n$ harán su parte y S^n retrocederá hacia la matriz cero. Si queremos garantizar que no haya crecimiento al principio, entonces queremos que la *norma* de S sea $||S|| < 1$. Así $||S^n||$ se mantendrá debajo de $||S||^n$ e irá directamente hacia cero.

Sabemos que $|\lambda_2| < 1, |\lambda_3| < 1, |\lambda_4| < 1$. Si a, b, c son pequeños, entonces la norma será menor que 1:

$$||S|| \leq ||\text{parte diagonal}|| + ||\text{parte fuera de la diagonal}|| < 1.$$

Alcanzamos esta $S = M^{-1}AM$ triangular en dos pasos. Primero, toda matriz cuadrada A es similar a una matriz triangular superior $T = Q^{-1}AQ$. Esto mismo dice el teorema de Schur, pero con una matriz ortogonal Q. Su prueba es simple, está en la página 343 de *Introduction to Linear Algebra*. Luego reducimos A, B, C en T utilizando una matriz diagonal D para alcanzar a, b, c pequeños en S:

$$D^{-1}TD = \begin{bmatrix} 1 & & \\ & 1/d & \\ & & 1/d^2 \end{bmatrix} \begin{bmatrix} \lambda_2 & A & B \\ 0 & \lambda_3 & C \\ 0 & 0 & \lambda_4 \end{bmatrix} \begin{bmatrix} 1 & & \\ & d & \\ & & d^2 \end{bmatrix} = \begin{bmatrix} \lambda_2 & dA & d^2B \\ 0 & \lambda_3 & dC \\ 0 & 0 & \lambda_4 \end{bmatrix} = S.$$

Para d pequeño, los números por fuera de la diagonal dA, d^2B y dC se vuelven tan pequeños como querramos. Entonces S es $D^{-1}(Q^{-1}AQ)D = M^{-1}AM$, como se necesita para el paso 1 y el paso 2.

Si $P \geq 0$ *no es estrictamente positiva*, todo depende de los autovalores de P. Nos enfrentamos a la posibilidad de que $|\lambda_2| = 1$ y las potencias P^n no converjan. He aquí ejemplos:

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0.5 \\ 0 & 0.5 & 0.5 \end{bmatrix} \quad (\lambda = 1, 1, 0) \qquad P = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (\lambda^3 = 1)$$

"La ruina del jugador"

Esta matriz P es un ejemplo clásico de Markov. Dos apostadores tienen en conjunto \$3. El sistema tiene cuatro estados $(3, 0), (2, 1), (1, 2)$, y $(0, 3)$. Los estados *absorbentes* son $(3, 0)$ y $(0, 3)$, cuando un jugador gana todo el dinero; aquí termina el juego y no hay manera de salir de esos estados. Con esos dos estados estacionarios podemos esperar $\lambda = 1$ dos veces.

Los estados transientes son $(2, 1)$ y $(1, 2)$, cuando un juego de \$1 se juega con probabilidad p de que el jugador 1 gane y probabilidad $q = 1 - p$ de que el jugador 2 gane. Cuando el jugador 1 tiene \$2 o \$1, la matriz de transición P de 4 por 4 tiene esos números $\boldsymbol{p}$ y $\boldsymbol{q}$ en sus columnas del medio.

$$\boldsymbol{P} = \begin{bmatrix} \mathbf{1} & \boldsymbol{p} & 0 & 0 \\ 0 & 0 & \boldsymbol{p} & 0 \\ 0 & \boldsymbol{q} & 0 & 0 \\ 0 & 0 & \boldsymbol{q} & \mathbf{1} \end{bmatrix}$$

Pregunta : ¿ Cuáles son los cuatro autovalores de P ?
Respuesta : $\lambda = 1, 1, \sqrt{pq}$, y $-\sqrt{pq}$. Así $|\lambda_2| = 1$ y no hay un estado estacionario único.
Pregunta : ¿ Cuál es la probabilidad de que el juego continúe para siempre sin ganador ?
Respuesta : **Cero**. El juego terminará con probabilidad 1.
Pregunta : **Si el juego comienza en (\$2, \$1), ¿ cuál es la probabilidad p^* de que el jugador 1 gane ?**
Respuesta : ¡ Buena pregunta ! El jugador 1 ganará inmediatamente en la primera ronda con probabilidad p. La probabilidad de que el jugador 2 gane en la primera ronda y cambie el estado a $(\$1, \$2)$ es $q = 1 - p$. Así p es la probabilidad de que el jugador 1 gane en la segunda ronda y se regrese al estado $(\$2, \$1)$. A partir de ahí el jugador 1 eventualmente ganará con probabilidad p^*. A partir de esto podemos encontrar p^* :

$$\boldsymbol{p^* = p + qpp^*} \quad \text{y} \quad \boldsymbol{p^* = \frac{p}{1 - qp}}$$

Ecuaciones maestras : procesos de Markov continuos

Las ecuaciones maestras fueron consagradas con un nombre espectacular. Son ecuaciones diferenciales lineales $\boldsymbol{dp/dt = Ap}$ para un vector de probabilidad $\boldsymbol{p}(t)$ (componentes no negativos que suman 1). La matriz A es especial : **negativa o cero** sobre la diagonal, **positiva o cero** fuera de la diagonal, **las columnas suman cero**. Este proceso de Markov continuo tiene probabilidades $\boldsymbol{e^{At}p}(0)$.

La probabilidad de estar en el estado j a tiempo t es $p_j(t)$. La probabilidad de cambiar del estado j al estado i en un intervalo breve dt es $a_{ij}\, dt$. Dado $\boldsymbol{p}(0)$, la solución se obtiene de la matriz exponencial $\boldsymbol{p}(t) = \boldsymbol{e^{tA}p}(0)$. Esa matriz e^{At} será una matriz de Markov.

Prueba. Si n es grande, $I + (tA/n)$ no es una matriz de Markov ordinaria. Sus columnas suman 1. Así $\left(I + \frac{tA}{n}\right)^n$ converge a $P = e^{tA}$ que también es Markov. Y a medida que $t \to \infty, e^{tA}$ converge de la forma usual a un límite P^∞.

Un ejemplo es la matriz A con diagonales $1, -2, 1$, excepto que $A_{11} = A_{NN} = -1$. Esto es el *laplaciano de grafo* sobre una recta de nodos. Aproximaciones por diferencias finitas a la ecuación de calor con condiciones de borde de Neumann usan esa matriz.

Esta A aparece en la ecuación maestra para la *reacción bimolecular* $\boldsymbol{A + B \to C}$. Una molécula de A se combina químicamente con una molécula de B para formar una molécula de C.

$$\boldsymbol{A} = \begin{array}{c} \\ \textcircled{0} \\ \textcircled{1} \\ \textcircled{2} \\ \textcircled{3} \\ \textcircled{4} \end{array} \begin{array}{c} \begin{array}{ccccc} \textcircled{0} & \textcircled{1} & \textcircled{2} & \textcircled{3} & \textcircled{4} \end{array} \\ \begin{bmatrix} -16 & 1 & 0 & 0 & 0 \\ 16 & -10 & 2 & 0 & 0 \\ 0 & 9 & -6 & 3 & 0 \\ 0 & 0 & 4 & -4 & 4 \\ 0 & 0 & 0 & 1 & -4 \end{bmatrix} \end{array}$$

Columnas de A suman cero
Columnas de $I + \frac{tA}{n}$ suman uno
Así $P = e^{At}$ es Markov

Problemas de la Sección V.6

1 Encuentre una matriz de Markov $P \geq 0$ con algún elemento nulo pero tal que P^2 sea estrictamente positiva. Las columnas de P suman 1 de manera que $\lambda_{\text{máx}} = 1$. ¿Cómo sabe que los otros autovalores de P verifican $|\lambda| < 1$? Así P^n aproxima P^∞ con $\boldsymbol{v}_1$ en cada columna.

2 Si A tiene todos sus elementos positivos, entonces $A^{\mathrm{T}}A$ y AA^{T} tienen todos sus elementos positivos. Use el teorema de Perron para mostrar que la matriz de rango 1, $\sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^{\mathrm{T}}$ más cercana a A también es positiva.

3 Estas matrices tienen $||A|| > 1$ y $||M|| > 1$. Encuentre matrices B y C de manera que $||BAB^{-1}|| < 1$ y $||CMC^{-1}|| < 1$. Esto seguro es posible porque los _____ de A y M son menores que 1 en valor absoluto. ¿Por qué es esto imposible si M es de Markov?

$$A = \begin{bmatrix} \frac{1}{2} & 1 \\ 0 & \frac{1}{2} \end{bmatrix} \qquad M = \begin{bmatrix} .8 & .1 \\ .8 & .1 \end{bmatrix}$$

4 ¿Por qué es $||BZB^{-1}|| \leq 1$ imposible para cualquier B, pero $||CYC^{-1}|| \leq 1$ es posible?

$$Z = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad Y = \begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix}$$

5 Si toma las potencias de A, ¿cuál es el límite de A^n cuando $n \to \infty$?

$$A = \begin{bmatrix} 2/3 & 1/3 \\ 1/3 & 2/3 \end{bmatrix} \quad \text{y también} \quad A = \frac{1}{4}\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

6 Supongamos que todos los años el 99 % de las personas en Nueva York y Florida van a Florida y Nueva York pero el 1 % muere. ¿Puede crear una matriz de Markov P de 3 por 3 que corresponda a los tres estados Nueva York-Florida-muertes? ¿Cuál es el límite de las matrices P^n a medida que $n \to \infty$?

Parte VI

Optimización

VI.1 Problemas de mínimos: convexidad y el método de Newton

VI.2 Multiplicadores de Lagrange = derivadas del costo

VI.3 Programación lineal, teoría de juego y dualidad

VI.4 Descenso por gradiente hacia el mínimo

VI.5 Descenso por gradiente estocástico y ADAM

Parte VI : Optimización

El objetivo de la optimización es minimizar una función $F(x_1, \ldots, x_n)$ con muchas variables. Debemos comenzar este tema con la ecuación más importante del cálculo : *Derivada = cero en el punto mínimo* $\boldsymbol{x^*}$. Con n variables, F tiene n derivadas parciales $\partial F/\partial x_i$. Si no hay "restricciones" que $\boldsymbol{x}$ deba satisfacer, tenemos n ecuaciones $\partial F/\partial x_i = 0$ con n incógnitas $x_1^*, \ldots, x_n^*$.

Al mismo tiempo, a menudo hay condiciones que el vector $\boldsymbol{x}$ debe satisfacer. Estas **restricciones sobre** $\boldsymbol{x}$ podrían ser ecuaciones $A\boldsymbol{x} = \boldsymbol{b}$ o desigualdades $\boldsymbol{x} \geq \boldsymbol{0}$. Las restricciones ingresan en las ecuaciones mediante multiplicadores de Lagrange $\lambda_1, \ldots, \lambda_m$. Ahora tenemos $m + n$ incógnitas (los x y los λ) y $m + n$ ecuaciones (derivadas $= 0$). Así este tema combina álgebra lineal con el cálculo multivariable. A menudo trabajamos en dimensiones altas.

Esta introducción termina con una afirmación clave de cálculo : la aproximación de $\boldsymbol{F(x + \Delta x)}$ por $\boldsymbol{F(x) + \Delta x^{\mathrm{T}} \nabla F + \frac{1}{2} \Delta x^{\mathrm{T}} H \Delta x}$. Vea esa página que es importante.

Evidentemente esto es parte de las matemáticas. Sin embargo, la optimización tiene tanto ideas como algoritmos propios. No siempre los departamentos de matemática la presentan como contenido esencial, pero las ciencias sociales y las ciencias físicas como economía, finanzas, psicología, sociología y todos los campos de ingeniería la usan y la enseñan porque la necesitan.

Organizamos este capítulo de manera que enfatice las ideas clave de la optimización :

VI.1 La importancia central de la **convexidad**, que reemplaza la linealidad. La convexidad usa las derivadas segundas: el gráfico de $F(\boldsymbol{x})$ se curvará hacia arriba. Una gran pregunta computacional es si podemos encontrar y usar todas esas derivadas parciales segundas $\partial^2 F/\partial x_i \partial x_j$. La elección será entre los **"métodos de Newton"** que las usan y los **"métodos de gradiente"** que no : los métodos de segundo orden o los métodos de primer orden.

Por lo general, las redes neuronales del aprendizaje profundo involucran muchísimas incógnitas. En este caso se eligen métodos de gradiente (primer orden). *A menudo F no es convexa*. Las últimas secciones de este capítulo describen esos algoritmos importantes, se desplazan a lo largo del gradiente (el vector de derivadas primeras) hacia el mínimo de $F(\boldsymbol{x})$.

VI.2 El significado de **los multiplicadores de Lagrange**, que incluyen las restricciones en la ecuación *derivada = cero*. Lo que es más importante, esos multiplicadores dan las **derivadas del costo con respecto a las restricciones**. Son los griegos de las matemáticas de finanzas.

VI.3 Los problemas clásicos **LP, QP, SDP** de la "programación matemática". Las incógnitas son vectores o matrices. Las desigualdades indican vectores no negativos $\boldsymbol{x} \geq \mathbf{0}$ o matrices semidefinidas positivas $\boldsymbol{X} \geq \mathbf{0}$. Cada problema de minimización tiene un **problema dual**, una *maximización*. Los multiplicadores en un problema se convierten en las incógnitas del problema dual. Ambos aparecen en un **juego con dos participantes**.

VI.4 Los algoritmos de primer orden comienzan con **descenso por gradiente**. La derivada del costo en la dirección de búsqueda es negativa. La elección de la dirección en la cual moverse y cuánto moverse son el arte de la optimización computacional. Verá que hay decisiones cruciales por tomar, como agregar "momento" para descender con mayor velocidad.

Levenberg-Marquardt combina el descenso por gradiente con el método de Newton. La idea es acercarse a $\boldsymbol{x}^*$ con un método de primer orden y luego converger rápidamente con (casi) segundo orden. Este es un favorito para los cuadrados mínimos no lineales.

VI.5 **Descenso por gradiente estocástico.** En redes neuronales, la función a minimizar es una suma de muchos términos, las pérdidas en todas las muestras de los datos de entrenamiento. La función de aprendizaje F depende de los "pesos". Calcular su gradiente es costoso. Entonces cada paso solo aprende un **minilote** de B muestras de entrenamiento elegidas al azar o "estocásticamente". Un paso considera una parte de los datos pero no todos. *Esperamos y deseamos que esa parte sea razonablemente típica en relación al todo.*

El descenso por gradiente estocástico, a menudo con términos de aceleración de "ADAM" para utilizar direcciones de pasos anteriores, se ha convertido en la bestia de carga del aprendizaje profundo. Las derivadas parciales que necesitamos de F se calculan con **retropropagación**. Esta idea clave será explicada por separado en el capítulo final del libro.

Al igual que gran parte de la matemática aplicada, la optimización tiene una forma discreta y una continua. Nuestras incógnitas son vectores y nuestras restricciones involucran matrices. Para el *cálculo de variaciones* las incógnitas son funciones y las restricciones involucran integrales. La ecuación vectorial "primera derivada = cero" se vuelve la ecuación diferencial de Euler-Lagrange "primera variación = cero".

Ese es un mundo de optimización paralelo (y continuo) pero no nos adentraremos allí.

La expresión "argmin"

El mínimo de la función $F(x) = (x-1)^2$ es cero : *min* $F(x) = 0$. Esto nos dice cuánto baja el gráfico de F. Pero no nos dice qué número $\boldsymbol{x}^*$ da el mínimo. En optimización, ese "argumento" $\boldsymbol{x}^*$ es el número que intentamos encontrar. El x que minimiza $F = (x-1)^2$ es $\boldsymbol{x}^* = \textbf{argmin}\, \boldsymbol{F(x)} = \mathbf{1}$.

$$\textbf{argmin}\, \boldsymbol{F(x)} = \textbf{ valor(es) de } \boldsymbol{x} \textbf{ donde } \boldsymbol{F} \textbf{ alcanza su mínimo.}$$

Para funciones estrictamente convexas, argmin $F(x)$ es **un punto** $\boldsymbol{x}^*$: un mínimo aislado.

Cálculo multivariable

El aprendizaje automático requiere de funciones $F(x_1, \ldots, x_n)$ de muchas variables. Necesitamos resultados básicos sobre las derivadas primeras y segundas de F. Estas son "derivadas parciales" cuando $n > 1$. Los resultados importantes están en las ecuaciones (1)-(2)-(3). No creo que necesite un curso entero (demasiado sobre integrales) para usar estos resultados al optimizar una función de aprendizaje profundo $F(x)$.

$$\boxed{\begin{array}{l}\textbf{Una función } \boldsymbol{F} \\ \textbf{Una variable } \boldsymbol{x}\end{array} \quad \boldsymbol{F(x + \Delta x) \approx F(x) + \Delta x \, \frac{dF}{dx}(x) + \frac{1}{2}(\Delta x)^2 \, \frac{d^2F}{dx^2}(x)}} \tag{1}$$

Estos son los primeros términos de una serie de Tayor, y a menudo no vamos más allá de ese último término de segundo orden. Los primeros términos $F(x) + (\Delta x)(dF/dx)$ dan una aproximación de *primer orden* para $F(x + \Delta x)$, con información correspondiente a x. Luego el término $(\Delta x)^2$ lo transforma en una aproximación de *segundo orden*.

La suma de los dos primeros términos (hasta el término de Δx) da un punto en la recta tangente, que es tangente al gráfico de $F(x)$. El término $(\Delta x)^2$ se mueve de la recta tangente a la "parábola tangente". La función F será **convexa**: su pendiente aumenta y el gráfico se curva hacia arriba, como en $y = x^2$, cuando la segunda derivada de $F(x)$ es positiva: $\boldsymbol{d^2F/dx^2 > 0}$. **La ecuación (2) es la versión n-dimensional de (1).**

$$\boxed{\begin{array}{l}\textbf{Una función } \boldsymbol{F} \\ \textbf{Variables } \boldsymbol{x_1} \textbf{ a } \boldsymbol{x_n}\end{array} \quad \boldsymbol{F(x + \Delta x) \approx F(x) + (\Delta x)^{\mathrm{T}} \nabla F + \frac{1}{2}(\Delta x)^{\mathrm{T}} H \, (\Delta x)}} \tag{2}$$

Esta es la fórmula importante. **El vector $\boldsymbol{\nabla F}$ es el gradiente de $\boldsymbol{F}$**, el vector columna de n derivadas parciales $\partial F/\partial x_1$ a $\partial F/\partial x_n$. **La matriz $\boldsymbol{H}$ es la matriz hessiana**. H es la matriz simétrica de derivadas segundas $H_{ij} = \partial^2 F/\partial x_i \, \partial x_j = \partial^2 F/\partial x_j \, \partial x_i$.

El gráfico de $y = F(x_1, \ldots, x_n)$ es ahora una superficie en un espacio de dimensión $(n + 1)$. La recta tangente se vuelve un plano tangente en $\boldsymbol{x}$. Cuando la matriz de derivadas segundas H es definida positiva, F es una *función estrictamente convexa*: *su gráfico se encuentra por encima de sus tangentes*. Una función convexa F tiene un **mínimo** en $\boldsymbol{x^*}$ si $\boldsymbol{f = \nabla F(x^*) = 0}$: n ecuaciones para $\boldsymbol{x^*}$. A veces encontramos m funciones diferentes $f_1(\boldsymbol{x})$ a $f_m(\boldsymbol{x})$: una *función vectorial* $\boldsymbol{f}$:

$$\boxed{\begin{array}{l}\boldsymbol{m} \textbf{ funciones } \boldsymbol{f = (f_1, \ldots, f_m)} \\ \boldsymbol{n} \textbf{ variables } \boldsymbol{x = (x_1, \ldots, x_n)}\end{array} \quad \boldsymbol{f(x + \Delta x) \approx f(x) + J(x) \, \Delta x}} \tag{3}$$

El símbolo $\boldsymbol{J(x)}$ representa la **matriz jacobiana de $\boldsymbol{f(x)}$**, de m por n en el punto $\boldsymbol{x}$. Las m filas de J contienen los vectores gradientes de las m funciones $f_1(\boldsymbol{x})$ a $f_m(\boldsymbol{x})$.

$$\textbf{Matriz jacobiana } \boldsymbol{J} = \begin{bmatrix} (\nabla f_1)^{\mathrm{T}} \\ \vdots \\ (\nabla f_m)^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \dfrac{\partial f_m}{\partial x_1} & \cdots & \dfrac{\partial f_m}{\partial x_n} \end{bmatrix} \tag{4}$$

¡ La **matriz hessiana $\boldsymbol{H}$ es la jacobiana $\boldsymbol{J}$ del gradiente $\boldsymbol{f = \nabla F}$**! El determinante de J (cuando $m = n$) aparece en integrales n-dimensionales. Es la r en la fórmula de área $\iint r \, dr \, d\theta$.

VI.1. Problemas de mínimos : convexidad y método de Newton

Esta parte del libro se enfoca en problemas de minimización de funciones $F(\boldsymbol{x})$ con muchas variables : $F(\boldsymbol{x}) = F(x_1, \ldots, x_n)$. A menudo habrá restricciones en los vectores $\boldsymbol{x}$:

Restricciones lineales	$\boldsymbol{Ax} = \boldsymbol{b}$	(el conjunto de estos $\boldsymbol{x}$ es convexo)
Restricciones de desigualdad	$\boldsymbol{x} \geq \boldsymbol{0}$	(el conjunto de estos $\boldsymbol{x}$ es convexo)
Restricciones binarias	**Cada x_i es 0 o 1**	(el conjunto de estos $\boldsymbol{x}$ no es convexo)

El planteo del problema puede ser no estructurado o muy estructurado. Así los algoritmos para encontrar el $\boldsymbol{x}$ que minimiza varían entre muy generales a muy específicos. He aquí unos ejemplos :

No estructurado	Minimizar $F(\boldsymbol{x})$ para $\boldsymbol{x}$ en un subconjunto K de $\mathbf{R}^n$
Estructurado	Minimizar un costo cuadrático $F(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}$ restringido a $A\boldsymbol{x} = \boldsymbol{b}$
	Minimizar un costo lineal $F(\boldsymbol{x}) = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ restringido a $A\boldsymbol{x} = \boldsymbol{b}$ y $\boldsymbol{x} \geq \boldsymbol{0}$
	Minimizar con una restricción binaria : cada x_i es 0 o 1

No podemos avanzar con estos problemas hasta que reconozcamos el rol crucial que ocupa la **convexidad**. Esperamos que la función $F(\boldsymbol{x})$ sea convexa. Esperamos que el conjunto de restricciones K sea convexo. En la teoría de optimización, debemos vivir sin linealidad. La **convexidad** tomará el control cuando se pierde la linealidad. He aquí la convexidad para una función $F(\boldsymbol{x})$ y un conjunto de restricciones K :

K es un conjunto convexo	Si $\boldsymbol{x}$ e $\boldsymbol{y}$ están en K, también lo está el segmento que une $\boldsymbol{x}$ e $\boldsymbol{y}$
F es una función convexa	El conjunto de puntos sobre y encima del gráfico de F es convexo
F es suave y convexa	$F(\boldsymbol{x}) \geq F(\boldsymbol{y}) + \boldsymbol{\nabla} F(\boldsymbol{y})^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{y})$

Esa última desigualdad establece que el gráfico de una F convexa *está por encima de sus rectas tangentes*.

Un triángulo en el plano es sin duda un conjunto convexo en $\mathbf{R}^2$. ¿Qué hay de la unión de dos triángulos ? Estas son dos maneras de lograr que la unión sea convexa. ¿Puede encontrar otras?

1) Un triángulo contiene al otro triángulo. La unión es el triángulo más grande.
2) Comparten un lado y su unión no tiene vértices que apunten hacia adentro.

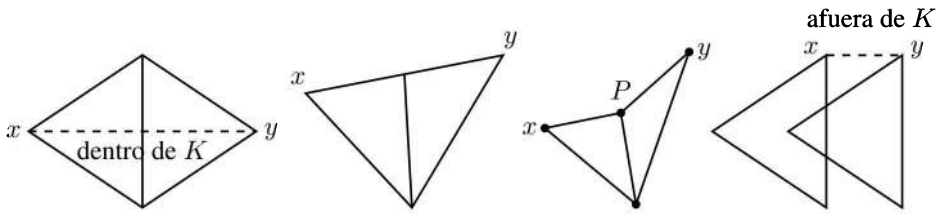


Figura VI.1: dos conjuntos convexos y dos no convexos en $\mathbf{R}^2$: apuntan hacia adentro en P.

Para las funciones F hay una forma directa de definir convexidad. Miremos todos los puntos $p\boldsymbol{x}+(1-p)\,\boldsymbol{y}$ entre $\boldsymbol{x}$ e $\boldsymbol{y}$. El gráfico de F está justo sobre o debajo de un segmento recto.

$$\boxed{\boldsymbol{F}\textbf{ es convexa}\quad \boldsymbol{F(px+(1-p)y)\ \le\ pF(x)+(1-p)F(y)}\ \text{ para }\ 0<p<1.} \tag{1}$$

Para una función estrictamente convexa esta desigualdad es estricta (reemplazamos $\le$ por $<$). Así el gráfico de F está estrictamente debajo de la cuerda que conecta el punto $(\boldsymbol{x}, F(\boldsymbol{x}))$ a $(\boldsymbol{y}, F(\boldsymbol{y}))$.

Es interesante que el gráfico esté arriba de sus rectas tangentes y debajo de sus cuerdas.

He aquí tres ejemplos de funciones convexas. Solo F_2 es estrictamente convexa :

$$F_1 = ax+b \qquad F_2 = x^2 \text{ (pero no } -x^2) \qquad F_3 = \max(F_1, F_2)$$

La convexidad de esa función F_3 es un dato importante. Aquí es donde la linealidad falla, pero la convexidad funciona. El máximo de dos o mas funciones lineales es raramente lineal. Pero **la $\boldsymbol{F(x)}$ que es el máximo de dos o más funciones convexas $\boldsymbol{F_i(x)}$ siempre es convexa**. Para $\boldsymbol{z} = p\boldsymbol{x}+(1-p)\boldsymbol{y}$ entre $\boldsymbol{x}$ e $\boldsymbol{y}$, cada función F_i satisface

$$\boxed{F_i(\boldsymbol{z}) \le pF_i(\boldsymbol{x})+(1-p)F_i(\boldsymbol{y}) \le p\,F(\boldsymbol{x})+(1-p)\,F(\boldsymbol{y}).} \tag{2}$$

Esto es cierto para cada i. Así $F(\boldsymbol{z}) = \max F_i(\boldsymbol{z}) \le pF(\boldsymbol{x})+(1-p)F(\boldsymbol{y})$, como se quería probar.

El máximo de cualquier familia de funciones convexas (en particular cualquier familia de funciones lineales) es una función convexa. Supongamos que tenemos *todas las funciones lineales que se mantienen debajo de una función convexa* F. El máximo de esas funciones lineales que están debajo de F es exactamente igual a F.

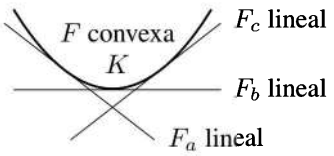


Figura VI.2: **una función convexa $\boldsymbol{F}$ es el máximo de todas sus funciones tangentes.**

Un conjunto convexo K es la intersección de todos los semiplanos que lo contienen. Y la *intersección* de cualquier familia de conjuntos convexos es un conjunto convexo. Pero la *unión* de conjuntos convexos no siempre es un conjunto convexo.

De manera similar, el *máximo* de cualquier familia de funciones convexas será una función convexa. Pero el *mínimo* de dos funciones convexas generalmente no será una función convexa, ya que puede tener un "pozo doble".

He aquí dos datos útiles sobre las matrices, basado en que def pos + def pos = def pos :

El conjunto de *matrices definidas positivas de n por n* es convexo.

El conjunto de *matrices semidefinidas positivas de n por n* es convexo.

El primer conjunto es "abierto". El segundo conjunto es "cerrado". Contiene sus puntos límites semidefinidos.

La matriz de derivadas segundas

Una función ordinaria $f(x)$ es convexa si $d^2f/dx^2 \geq 0$. Motivo: la pendiente df/dx es creciente. *La curva dobla hacia arriba* (como la parábola $f = x^2$ con derivada segunda $= 2$). La extensión a n variables involucra la **matriz $H(x)$ de derivadas segundas de n por n**. Si $F(x)$ es una función suave, entonces tenemos un criterio casi perfecto para la convexidad:

$F(x_1, \ldots, x_n)$ es convexa si y solo si su matriz de derivadas segundas $H(x)$ es **semidefinida positiva en todo x**. Esa **matriz hessiana** es simétrica porque $\partial^2 F/\partial x_i \partial x_j = \partial^2 F/\partial x_j \partial x_i$. La función F es **estrictamente** convexa si $H(x)$ es **definida positiva en todo x**.

$$H(x) = \begin{bmatrix} \partial^2 F/\partial x_1^2 & \partial^2 F/\partial x_1 \partial x_2 & \cdot \\ \partial^2 F/\partial x_2 \partial x_1 & \partial^2 F/\partial x_2^2 & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \qquad H_{ij} = \frac{\partial^2 F}{\partial x_i \partial x_j} = H_{ji}$$

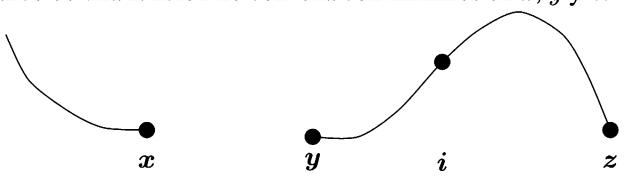
Una función lineal $F = c^{\mathrm{T}}x$ es convexa (pero no estrictamente convexa). Por arriba de su gráfico hay un semiespacio, con un borde plano. Su matriz de derivadas segundas es $H = 0$ (muy semidefinida).

Una F cuadrática $= \frac{1}{2}x^{\mathrm{T}}Sx$ tiene gradiente Sx. Su matriz simétrica de derivadas segundas es S. Cuando S es definida positiva la parte de arriba de su gráfico es un bol. Esta función F es estrictamente convexa.

La convexidad impide dos mínimos locales

Minimizamos una función convexa $F(x)$ para x en un conjunto convexo K. Esa convexidad doble trae un efecto favorable: *no habrá dos soluciones aisladas*. Si x e y están en K y dan el mismo mínimo, entonces todos los puntos z sobre el segmento recto entre ellos también están en K y dan ese mínimo. La convexidad evita la situación peligrosísima que ocurre cuando F tiene su valor mínimo en una cantidad desconocida de puntos separados de K.

Esta contribución de la convexidad ya es clara para las funciones ordinarias $F(x)$ de una variable x. He aquí el gráfico de una función no convexa con mínimos en x, y y z.



F no es convexa. Es cóncava a partir del punto de inflexión i, donde $\partial^2 F/\partial x^2$ se vuelve negativa. Además F no está definida en un conjunto convexo K (debido al hueco entre x e y). Para arreglar ambos problemas, podríamos conectar x a y con un segmento de recta y terminar el gráfico en i.

Para que un problema convexo tenga múltiples soluciones x e y, todos los puntos del intervalo son soluciones. Nunca dos mínimos aislados, generalmente es un solo punto. **El conjunto de puntos minimizantes x en un problema convexo es convexo.**

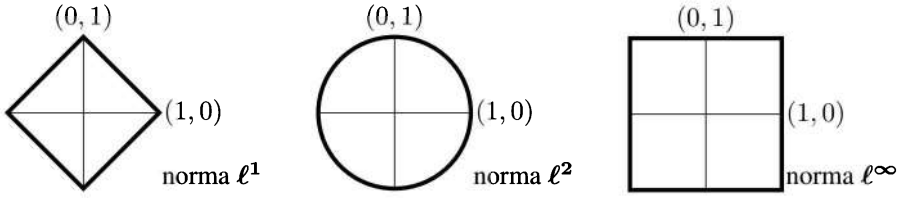
El sistema **CVX** ofrece software MATLAB para "problemas convexos disciplinados". El usuario elige la programación cuadrática, lineal y de mínimos cuadrados,... Vea **cvxr.com/cvx**.

Las normas ℓ^1, ℓ^2 y ℓ^∞ de x

Las **normas $F(x) = \|x\|$ son funciones convexas de x. La bola unitaria, es decir el conjunto donde $\|x\| \le 1$ es un conjunto convexo K de vectores x.** Esta primera afirmación es exactamente la desigualdad triangular :

$$\textbf{Convexidad de } \|x\| \quad \|p\,x + (1-p)\,y\| \le p\|x\| + (1-p)\|y\|$$

Son tres las normas vectoriales favoritas $\ell^1, \ell^2, \ell^\infty$. Dibujamos las bolas unitarias $\|x\| \le 1$ en $\mathbf{R}^2$:



$\|x\|_1 = |x_1| + |x_2|$ $\quad$ $\|x\|_2 = \sqrt{x_1^2 + x_2^2}$ $\quad$ $\|x\|_\infty = \max(|x_1|, |x_2|)$

Figura VI.3: para todas las normas, la "bola unitaria" convexa $\|x\| \le 1$ está centrada en $x = 0$.

El Método de Newton

Buscamos el punto x^* donde $F(x)$ tiene un mínimo y su gradiente $\nabla F(x^*)$ es el vector cero. Hemos alcanzado un punto cercano x_k. Queremos movernos a un punto x_{k+1} nuevo que esté más cerca que x_k de $x^* = \operatorname{argmin} F(x)$. ¿Cuál es un paso apropiado $x_{k+1} - x_k$ para alcanzar ese punto nuevo x_{k+1} ?

El cálculo ofrece una solución. Cerca de nuestro punto x_k, el gradiente ∇F a menudo se estima bien con sus derivadas primeras, que son las derivadas segundas de $F(x)$. Esas derivadas segundas $\partial^2 F/\partial x_i \partial x_j$ se encuentran en la matriz hessiana H :

$$\nabla F(x_{k+1}) \approx \nabla F(x_k) + H(x_k)(x_{k+1} - x_k). \tag{3}$$

Queremos que ese lado izquierdo sea cero. Entonces, la elección natural para x_{k+1} se obtiene cuando el *lado derecho es cero* : tenemos n ecuaciones lineales para el paso $\Delta x_k = x_{k+1} - x_k$:

$$\boxed{\textbf{El método de Newton} \qquad H(x_k)(\Delta x_k) = -\nabla F(x_k) \quad \text{y} \quad x_{k+1} = x_k + \Delta x_k} \tag{4}$$

El método de Newton da en cada paso también el minimizador de una función cuadrática construida a partir de F, sus derivadas ∇F y sus derivadas segundas H en el punto x_k :

$$x_{k+1} \text{ minimiza } F(x_k) + \nabla F(x_k)^{\mathrm{T}}(x - x_k) + \frac{1}{2}(x - x_k)^{\mathrm{T}} H(x_k)(x - x_k). \tag{5}$$

El método de Newton es de *segundo orden*, usa las derivadas segundas (en H). Todavía habrá un error en el nuevo $\boldsymbol{x}_{k+1}$. Pero ese error es proporcional al **cuadrado** del error en $\boldsymbol{x}_k$:

$$\boxed{\textbf{Convergencia cuadrática} \qquad ||\boldsymbol{x}_{k+1}-\boldsymbol{x}^*|| \le C\,||\boldsymbol{x}_k-\boldsymbol{x}^*||^2.} \tag{6}$$

Si $\boldsymbol{x}_k$ está cerca de $\boldsymbol{x}^*$, entonces $\boldsymbol{x}_{k+1}$ estará mucho más cerca. Un ejemplo es el cálculo de $\boldsymbol{x}^* = \sqrt{4} = 2$ en una dimensión. Newton resuelve $\boldsymbol{x^2 - 4 = 0}$:

$$\textbf{Minimizar} \qquad \boldsymbol{F(x) = \tfrac{1}{3}x^3 - 4x} \quad \textbf{con} \quad \boldsymbol{\nabla F(x) = x^2 - 4} \quad \textbf{y} \quad \boldsymbol{H(x) = 2x}$$

Un paso del método de Newton : $H(x_k)(\Delta x_k) = 2x_k(x_{k+1} - x_k) = -x_k^2 + 4.$

Entonces $2x_k x_{k+1} = x_k^2 + 4.$ Así Newton elige $\boldsymbol{x_{k+1} = \frac{1}{2}\left(x_k + \frac{4}{x_k}\right)}$.

Adivinemos la raíz cuadrada, dividámosla por 4 *y promediemos los dos números*. Podemos empezar con 2.5 :

$$x_0 = \mathbf{2.5} \qquad x_1 = \mathbf{2.05} \qquad x_2 = \mathbf{2.0006} \qquad x_3 = \mathbf{2.000000009}$$

El decimal incorrecto está al doble de distancia en cada paso. **El error $\boldsymbol{x_k - 2}$ se eleva al cuadrado** :

$$\boxed{x_{k+1} - 2 = \frac{1}{2}\left(x_k + \frac{4}{x_k}\right) - 2 = \frac{1}{2x_k}\,(x_k - 2)^{\mathbf{2}}} \qquad \boxed{||\boldsymbol{x}_{k+1} - \boldsymbol{x^*}|| \approx \frac{1}{4}||\boldsymbol{x}_k - \boldsymbol{x^*}||^2}$$

El error al cuadrado explica la velocidad del método de Newton, *siempre y cuando $\boldsymbol{x}_k$ esté cerca.*

¿ Qué tan bien funciona el método de Newton en la práctica ? Al principio, $\boldsymbol{x}_0$ puede no estar cerca de $\boldsymbol{x^*}$. No podemos confiar que la derivada segunda en $\boldsymbol{x}_0$ sea útil. Entonces calculamos el paso de Newton $\Delta \boldsymbol{x}_0 = \boldsymbol{x}_1 - \boldsymbol{x}_0$, pero luego permitimos el **retroceso** (backtracking) :

> Elegimos $\alpha < \frac{1}{2}$ y $\beta < 1$ y reducimos el paso $\Delta \boldsymbol{x}$ por el factor β hasta que sepamos que el $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + t\Delta \boldsymbol{x}$ nuevo produce suficiente reducción de $F(\boldsymbol{x})$:

$$\textbf{Reducir } \boldsymbol{t} \textbf{ hasta que la reducción en } \boldsymbol{F} \textbf{ satisfaga} \quad F(\boldsymbol{x}_k + t\Delta \boldsymbol{x}) \le F(\boldsymbol{x}_k) + \alpha\, t\, \boldsymbol{\nabla F}^{\mathrm{T}} \Delta \boldsymbol{x}. \tag{7}$$

Retomaremos el retroceso en la Sección VI.4. Después de establecer la dirección de búsqueda desde $\boldsymbol{x}_k$, el retroceso es un paso que permite obtener una elección segura de $\boldsymbol{x}_{k+1}$. O podemos fijar un tamaño de paso pequeño, este sería el hiperparámetro, y no hacer la búsqueda durante el entrenamiento de una gran red neuronal.

En resumen El método de Newton es eventualmente rápido porque usa las derivadas segundas de $F(\boldsymbol{x})$. Pero esas pueden ser muy costosas de calcular, sobre todo en dimensiones altas. Los *métodos de Quasi-Newton* en la Sección III.1 permiten que la matriz hessiana H se construya gradualmente a partir de información obtenida a medida que el algoritmo continúa. A menudo las redes neuronales son simplemente demasiado grandes para usar H. El *descenso por gradiente* es el algoritmo preferido y se desarrolla en VI.4-5.

Las siguientes dos páginas describen una solución de compromiso que mejora a medida que nos acercamos a $\boldsymbol{x^*}$.

Levenberg-Marquardt para cuadrados mínimos no lineales

Cuadrados mínimos comienza con un conjunto de m puntos de datos (t_i, y_i) y busca aproximar esos m puntos de la mejor manera posible eligiendo los parámetros $\boldsymbol{p} = (p_1, \ldots, p_n)$ en una función de ajuste $\widehat{Y}(t, \boldsymbol{p})$. Supongamos que esos parámetros son los $\boldsymbol{p} = (C, D)$ usuales para una recta de ajuste dada por $\widehat{y} = C + Dt$. Así la suma de los errores al cuadrado depende de C y D :

$$E(C, D) = (y_1 - C - Dt_1)^2 + \cdots + (y_m - C - Dt_m)^2. \tag{8}$$

El error mínimo E se encuentra en los valores $\widehat{C}$ y $\widehat{D}$ donde $\partial E/\partial C = 0$ y $\partial E/\partial D = 0$. Esas ecuaciones son lineales porque C y D aparecen linealmente en la ecuación de ajuste $\widehat{y} = C + Dt$. Estamos buscando la mejor solución de cuadrados mínimos a m ecuaciones lineales $\boldsymbol{Jp} = \boldsymbol{y}$:

m ecuaciones 2 incógnitas sin solución	$J \begin{bmatrix} C \\ D \end{bmatrix} = \begin{bmatrix} 1 & t_1 \\ \cdot & \cdot \\ \cdot & \cdot \\ 1 & t_m \end{bmatrix} \begin{bmatrix} C \\ D \end{bmatrix} = \begin{bmatrix} y_1 \\ \cdot \\ \cdot \\ y_m \end{bmatrix} = \boldsymbol{y}$
2 ecuaciones 2 incógnitas	$J^{\mathrm{T}} J \begin{bmatrix} \widehat{C} \\ \widehat{D} \end{bmatrix} = J^{\mathrm{T}} \boldsymbol{y}$ para los parámetros óptimos $\widehat{\boldsymbol{p}} = \begin{bmatrix} \widehat{C} \\ \widehat{D} \end{bmatrix}$.

Esto es cuadrados mínimos lineales. La función de ajuste $\widehat{\boldsymbol{y}}$ es lineal en C y D. A J normalmente la llamaríamos A. Pero para mínimos cuadrados **no lineales** la función de ajuste $\widehat{\boldsymbol{y}}(\boldsymbol{p})$ depende de manera no lineal de los n parámetros $\boldsymbol{p} = (p_1, \ldots, p_n)$. Cuando minimizamos el error total $E =$ suma de los cuadrados, esperamos encontrarnos con $\boldsymbol{n}$ **ecuaciones no lineales** para determinar los parámetros óptimos :

$$\begin{aligned} \boldsymbol{E}(\boldsymbol{p}) &= \sum_{i=1}^{m} (y_i - \widehat{y}_i)^2 = (\boldsymbol{y} - \widehat{\boldsymbol{y}}(\boldsymbol{p}))^{\mathrm{T}} (\boldsymbol{y} - \widehat{\boldsymbol{y}}(\boldsymbol{p})) \\ &= \boldsymbol{y}^{\mathrm{T}} \boldsymbol{y} - 2\boldsymbol{y}^{\mathrm{T}} \widehat{\boldsymbol{y}}(\boldsymbol{p}) + \widehat{\boldsymbol{y}}(\boldsymbol{p})^{\mathrm{T}} \widehat{\boldsymbol{y}}(\boldsymbol{p}). \end{aligned} \tag{9}$$

Esta es la función error de la "pérdida cuadrática" cuyo mínimo se alcanza cuando elegimos los mejores parámetros $\widehat{\boldsymbol{p}}$.

Las aplicaciones pueden incluir una matriz ponderada o una matriz de blanqueamiento W. A menudo W es una matriz diagonal con las inversas de las varianzas $1/\sigma_1^2, \ldots, 1/\sigma_m^2$. Estas aparecen en el error total $\boldsymbol{E} = (\boldsymbol{y} - \widehat{\boldsymbol{y}})^{\mathrm{T}} W (\boldsymbol{y} - \widehat{\boldsymbol{y}})$. Por lo tanto aparecen en las ecuaciones normales $J^{\mathrm{T}} W J \widehat{\boldsymbol{p}} = J^{\mathrm{T}} W \boldsymbol{y}$. Esto le da más peso a los datos con varianza más pequeña σ_i : los datos que son más confiables. Por simplicidad procedemos con $W = I$: pesos unitarios.

Nuestro problema es minimizar $\boldsymbol{E}(\boldsymbol{p})$ en la ecuación (9). Entonces calculamos su vector gradiente $\partial \boldsymbol{E}/\partial \boldsymbol{p} = \boldsymbol{\nabla E}$. Este gradiente es constante para mínimos cuadrados lineales, pero $\boldsymbol{\nabla E}$ depende de $\boldsymbol{p}$ para nuestro problema no lineal. La página siguiente describe un algoritmo para minimizar E aproximando Newton pero evitando las derivadas segundas de E.

$$\boxed{\nabla \boldsymbol{E} = 2\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{y} - \widehat{\boldsymbol{y}}\,(\boldsymbol{p}_n)) = \boldsymbol{0} \text{ con matriz jacobiana de } m \text{ por } n \;\; \boldsymbol{J} = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{p}} \text{ en } \widehat{\boldsymbol{p}}.} \tag{10}$$

J era una matriz constante de m por 2 cuando la función de ajuste $\widehat{\boldsymbol{y}} = C + Dt$ era lineal en los parámetros $\boldsymbol{p} = (C, D)$. *La ecuación de cuadrados mínimos para el error mínimo es* $\nabla \boldsymbol{E} = \boldsymbol{0}$. En el caso lineal esto era $J^{\mathrm{T}} J\widehat{\boldsymbol{p}} = J^{\mathrm{T}}\widehat{\boldsymbol{y}}$. En el caso no lineal, tenemos un método de primer orden (**decenso por gradiente**) y un **método de Newton aproximado** para resolver (10) :

$$\textbf{Descenso por gradiente} \qquad \boldsymbol{p}_{n+1} - \boldsymbol{p}_n = -\boldsymbol{s}\boldsymbol{J}^{\mathrm{T}}\left(\boldsymbol{y} - \widehat{\boldsymbol{y}}\,(\boldsymbol{p}_n)\right) \tag{11}$$

$$\textbf{Newton (Aproximado)} \qquad \boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}\left(\boldsymbol{p}_{n+1} - \boldsymbol{p}_n\right) = \boldsymbol{J}^{\mathrm{T}}\left(\boldsymbol{y} - \widehat{\boldsymbol{y}}\,(\boldsymbol{p}_n)\right) \tag{12}$$

Esa matriz simétrica $J^{\mathrm{T}}J$ es una aproximación de la matriz de derivadas segundas $\frac{1}{2}H$ (la hessiana de la función E). Para verlo, sustituimos la aproximación de primer orden $\widehat{\boldsymbol{y}}\,(\boldsymbol{p}+\Delta\boldsymbol{p}) \approx \widehat{\boldsymbol{y}}\,(\boldsymbol{p}) + J\Delta\boldsymbol{p}$ en la función de pérdida E en (9) :

$$E(\boldsymbol{p} + \boldsymbol{\Delta p}) \approx (\boldsymbol{y} - \widehat{\boldsymbol{y}}\,(\boldsymbol{p}) - J\boldsymbol{\Delta p})^{\mathrm{T}}(\boldsymbol{y} - \widehat{\boldsymbol{y}}\,(\boldsymbol{p}) - J\boldsymbol{\Delta p}). \tag{13}$$

El término de segundo orden es $\Delta\boldsymbol{p}^{\mathrm{T}} J^{\mathrm{T}} J\Delta\boldsymbol{p}$. Así $2J^{\mathrm{T}}J$ funciona como una matriz hessiana.

La idea de Levenberg y Marquardt fue combinar el descenso por gradiente y las reglas de actualización de Newton (11)-(12). Tiene un parámetro λ. Valores pequeños de λ dan una regla más similar a Newton, valores grandes de λ dan una regla más similar a descenso por gradiente. He aquí un método favorito (y flexible) para problemas de mínimos cuadrados no lineales :

$$\boxed{\textbf{Levenberg-Marquardt} \qquad (\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J} + \boldsymbol{\lambda}\boldsymbol{I})\left(\boldsymbol{p}_{n+1} - \boldsymbol{p}_n\right) = \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{y} - \widehat{\boldsymbol{y}}\,(\boldsymbol{p}_n)).} \tag{14}$$

Comenzamos con un λ bastante grande. El $\boldsymbol{p}_0$ inicial probablemente no esté cerca de la mejor elección $\boldsymbol{p}^*$. En este caso, cuando el problema es no lineal y la hessiana de E depende de $\boldsymbol{p}$, no podemos confiar en la precisión de $J^{\mathrm{T}}J$.

A medida que las aproximaciones $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots$ se acercan al valor correcto $\boldsymbol{p}^*$, la matriz $J^{\mathrm{T}}J$ se vuelve más confiable. Entonces hacemos más pequeño a λ. El objetivo es disfrutar de la convergencia rápida (casi Newton) a la solución de $\nabla \boldsymbol{E}\,(\boldsymbol{p}^*) = \boldsymbol{0}$ y al mínimo de $E(\boldsymbol{p})$.

Una variante útil es multiplicar a λI en (14) por la matriz diagonal diag $(J^{\mathrm{T}}J)$. Esto hace que λ sea adimensional. Al igual que con todos los descensos por gradiente, el código debe verificar que el error E disminuya en cada paso y ajustar el tamaño del paso según sea necesario. Un buen decrecimiento indica que λ puede reducirse y su siguiente iteración es más similar a una iteración de Newton.

Cuando $\lambda = 0$ *¿ Es exactamente Newton ? Lo lamento, pero no creo que lo sea. Volvamos a* (13).

El término cuadrático sugiere que la matriz de segundas derivadas (hessiana) es $2J^{\mathrm{T}}J$. Pero (13) es solo una aproximación de primer orden. Para cuadrados mínimos lineales, el primer orden era exacto. Para un problema no lineal, eso no puede ser cierto. El nombre oficial para esto es *Gauss-Newton*. En otras palabras, no podemos calcular una derivada segunda elevando al cuadrado una derivada primera.

Sin embargo Levenberg-Marquardt es un método de primer orden mejorado, muy útil para cuadrados mínimos no lineales. Es una manera de entrenar redes neuronales de tamaño moderado.

Problemas de la Sección VI.1

1 ¿ Cuándo es la unión de dos discos circulares un conjunto convexo ? ¿ O dos cuadrados ?

2 El texto propone solo dos maneras de unir dos triángulos en $\mathbf{R}^2$ para que sea convexa la unión. ¿ Es correcto que solo existen esos dos casos ? ¿ Y si los triángulos están en $\mathbf{R}^3$?

3 La **"cápsula convexa"** de un conjunto S en $\mathbf{R}^n$ es el menor conjunto convexo K que contiene a S. A partir del conjunto S, ¿ cómo construiría su cápsula convexa K ?

4 (a) Explique por qué la intersección $K_1 \cap K_2$ de dos conjuntos convexos es un conjunto convexo.

(b) ¿ Cómo se deduce entonces que el máximo F_3 de dos funciones convexas F_1 y F_2 es una función convexa ? Use la definición : F es convexa cuando el conjunto de puntos sobre y por encima de su gráfico es convexo. ¿ Cuál conjunto se encuentra sobre el gráfico de F_3 ?

5 Sea K convexo y $F(x) = 1$ para x en K y $F(x) = 0$ para x fuera de K. ¿ Es F una función convexa ? ¿ Y si se invierten el 0 y el 1 ?

6 A partir de sus derivadas segundas, muestre que estas funciones son convexas :

(a) Entropía $x \log x$

(b) $\log\,(e^x + e^y)$

(c) norma ℓ^p $||\boldsymbol{x}||_p = (|x_1|^p + |x_2|^p)^{1/p}$, $p \geq 1$

(d) $\lambda_{\text{máx}}\,(S)$ como función de las matrices simétricas S

7 $R(\boldsymbol{x}) = \dfrac{\boldsymbol{x}^{\mathrm{T}} S \boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}} \boldsymbol{x}} = \dfrac{x^2 + 2y^2}{x^2 + y^2}$ *no* es convexa. Tiene tanto un máximo como un mínimo. En el punto máximo, la matriz de derivadas segundas H es _____ .

8 Este capítulo incluye afirmaciones de la forma $\min\limits_x \max\limits_y K(x,y) = \max\limits_y \min\limits_x K(x,y)$. ¡ Pero minimax = maximin no siempre es cierto ! Explique este ejemplo :

$$\min_x \max_y (x+y) \quad y \quad \max_y \min_x (x+y) \quad \text{son} \quad +\infty \quad \text{y} \quad -\infty.$$

9 Sea $f(x,y)$ una función convexa suave y $f(0,0) = f(1,0) = f(0,1) = 0$.

(a) ¿ Qué sabe sobre $f\left(\frac{1}{2}, \frac{1}{2}\right)$?

(b) ¿ Qué sabe sobre las derivadas $a = \partial^2 f/\partial x^2$, $b = \partial^2 f/\partial x \partial y$, $c = \partial^2 f/\partial y^2$?

10 ¿ Podría una función suave cualquiera $f(x,y)$ en el círculo $x^2 + y^2 \leq 1$ escribirse como la diferencia $g(x,y) - h(x,y)$ de dos funciones convexas g y h ? Probablemente sí.

Los siguientes cuatro problemas tratan sobre el método de Newton.

11 Muestre que la ecuación (5) es correcta : **Δx de Newton minimiza esa cuadrática**.

12 ¿ Qué produce el método de Newton al intentar resolver $x^2 + 1 = 0$? Como no hay una solución (real), el método no puede converger. (Las iteraciones dan un lindo ejemplo de "caos".)

13 ¿ Qué produce el método de Newton al intentar resolver $\operatorname{sen} x = 0$? Como tiene muchas soluciones, puede ser difícil predecir el límite x^* de las iteraciones de Newton.

14 A partir de $\boldsymbol{x}_0 = (u_0, v_0)$, encuentre $\boldsymbol{x}_1 = (u_1, v_1)$ en el método de Newton para las ecuaciones $u^3 - v = 0$ y $v^3 - u = 0$. Newton converge a la solución $(0, 0)$, $(1, 1)$, $(-1, -1)$ o va al infinito. Si usa cuatro colores para los puntos iniciales (u_0, v_0) que llevan a esos cuatro límites, la imagen que se forma es hermosa.

15 Si f es una función convexa, sabemos que $f(x/2 + y/2) \leq \frac{1}{2}f(x) + \frac{1}{2}f(y)$. Si esta "prueba del punto medio" se cumple para todo x e y, muestre que también se cumple la "prueba de punto cuarto" $f(3x/4 + y/4) \leq \frac{3}{4}f(x) + \frac{1}{4}f(y)$. Esta es una prueba de punto medio entre x y $x/2 + y/2$. Así dos pruebas de punto medio dan la prueba de punto cuarto.

El mismo razonamiento eventualmente da $f(px + (1-p)y) \leq p\,f(x) + (1-p)\,f(y)$ para cualquier fracción $p = m/2^n \leq 1$. Estas fracciones son densas en todo el intervalo $0 \leq p \leq 1$. Si f es una función continua, entonces la prueba de mitad de camino para todo x, y lleva a la prueba de $px + (1-p)y$ para todo $0 \leq p \leq 1$. *Así la prueba del punto medio muestra que f es convexa.*

16 Realice el gráfico de una función estrictamente convexa $f(x)$.

Dibuje la cuerda entre dos puntos cualesquiera del gráfico.

Dibuje las rectas tangentes en esos dos puntos.

Entre x e y, verifique que las *rectas tangentes* $<$ $f(x)$ $<$ *cuerda*.

VI.2. Multiplicadores de Lagrange = derivadas del costo

Los problemas no estructurados tratan con funciones convexas $F(x)$ en conjuntos convexos K. Esta sección comienza con problemas muy estructurados, para ver cómo los multiplicadores de Lagrange lidian con las restricciones. Queremos mostrar el significado de los multiplicadores $\lambda_1, \ldots, \lambda_m$. Es un grave error descartarlos luego de introducirlos y utilizarlos.

Nuestro primer ejemplo es en dos dimensiones. La función F es cuadrática. El conjunto K es lineal.

$$\textbf{Minimizar } \boldsymbol{F(x) = x_1^2 + x_2^2} \textbf{ sobre la recta } \boldsymbol{K}\textbf{: } \boldsymbol{a_1 x_1 + a_2 x_2 = b}$$

En la recta K, buscamos el punto que se encuentre más cerca de $(0, 0)$. El costo $F(x)$ es la distancia al cuadrado. En la Figura VI.4, la recta que da la restricción es **tangente al círculo** en el punto ganador $\boldsymbol{x^*} = (x_1^*, x_2^*)$. Descubrimos esto a partir de un cálculo simple, *después de sustituir la ecuación de restricción* $a_1 x_1 + a_2 x_2 = b$ *en la función* $F = x_1^2 + x_2^2$.

Esta fue la bella idea de Lagrange.

Multiplicar $\boldsymbol{a_1 x_1 + a_2 x_2 - b}$ por un multiplicador desconocido $\boldsymbol{\lambda}$ y sumarlo a $\boldsymbol{F(x)}$

Lagrangiano $$\begin{aligned} \boldsymbol{L(x, \lambda)} &= F(x) + \lambda(a_1 x_1 + a_2 x_2 - b) \\ &= \boldsymbol{x_1^2 + x_2^2 + \lambda(a_1 x_1 + a_2 x_2 - b)} \end{aligned} \tag{1}$$

Igualar las derivadas $\boldsymbol{\partial L/\partial x_1}$, $\boldsymbol{\partial L/\partial x_2}$ y $\boldsymbol{\partial L/\partial \lambda}$ a cero.

Resolver esas tres ecuaciones para hallar $\boldsymbol{x_1, x_2, \lambda}$.

$$\boldsymbol{\partial L/\partial x_1 = 2x_1 + \lambda a_1 = 0} \tag{2a}$$

$$\boldsymbol{\partial L/\partial x_2 = 2x_2 + \lambda a_2 = 0} \tag{2b}$$

$$\boldsymbol{\partial L/\partial \lambda} \ \boldsymbol{= a_1 x_1 + a_2 x_2 - b = 0} \quad \text{(¡ la restricción !)} \tag{2c}$$

Las primeras ecuaciones dan $x_1 = -\frac{1}{2}\lambda a_1$ y $x_2 = -\frac{1}{2}\lambda a_2$. Reemplazamos en $a_1 x_1 + a_2 x_2 = b$:

$$-\frac{1}{2}\lambda a_1^2 - \frac{1}{2}\lambda a_2^2 = b \ \text{ y } \ \boldsymbol{\lambda = \frac{-2b}{a_1^2 + a_2^2}}. \tag{3}$$

Reemplazar λ en (2a) y (2b) revela el punto más cercano (x_1^*, x_2^*) y el costo mínimo $(x_1^*)^2 + (x_2^*)^2$:

$$\boldsymbol{x_1^* = -\frac{1}{2}\lambda a_1 = \frac{a_1 b}{a_1^2 + a_2^2}} \qquad \boldsymbol{x_2^* = -\frac{1}{2}\lambda a_2 = \frac{a_2 b}{a_1^2 + a_2^2}} \qquad \boldsymbol{(x_1^*)^2 + (x_2^*)^2 = \frac{b^2}{a_1^2 + a_2^2}}$$

La derivada del costo mínimo con respecto a b es menos el multiplicador de Lagrange:

$$\boxed{\frac{d}{db}\left(\frac{b^2}{a_1^2 + a_2^2}\right) = \frac{2b}{a_1^2 + a_2^2} = -\lambda.} \tag{4}$$

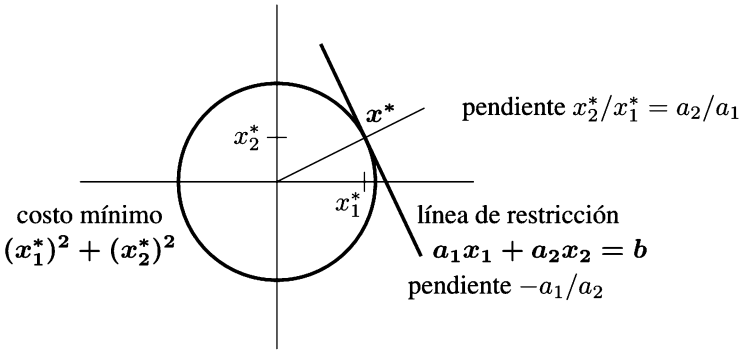


Figura VI.4: la recta de restricción es tangente al círculo de costo mínimo en la solución $\boldsymbol{x^*}$.

Minimizar una cuadrática con restricciones lineales

Pasamos de ese ejemplo del plano $\mathbf{R}^2$ al espacio $\mathbf{R}^n$. En lugar de una restricción en $\boldsymbol{x}$, tenemos $\boldsymbol{m}$ **restricciones** $\boldsymbol{A^{\mathrm{T}}x = b}$. **La matriz** $\boldsymbol{A^{\mathrm{T}}}$ **será de** $\boldsymbol{m}$ **por** $\boldsymbol{n}$. Tendremos m multiplicadores de Lagrange $\lambda_1, \ldots, \lambda_m$: uno para cada restricción. La función costo $F(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}$ permite cualquier matriz S simétrica definida positiva.

$$\textbf{Problema : minimizar } \boldsymbol{F = \tfrac{1}{2}x^{\mathrm{T}}Sx} \textbf{ sujeto a } \boldsymbol{A^{\mathrm{T}}x = b.} \tag{5}$$

Con m restricciones, tendremos m multiplicadores de Lagrange $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_m)$. Estos incluyen las restricciones $A^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{b}$ en el lagrangiano $L(\boldsymbol{x}, \boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x} + \boldsymbol{\lambda}^{\mathrm{T}}(A^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{b})$. Las $n + m$ derivadas de L dan $n + m$ ecuaciones para las componentes de los vectores $\boldsymbol{x}$ de $\mathbf{R}^n$ y $\boldsymbol{\lambda}$ de $\mathbf{R}^m$:

$$\boxed{\begin{array}{ll} \text{derivadas con respecto a } x \text{ de } L: & S\boldsymbol{x} + A\boldsymbol{\lambda} = \mathbf{0} \\ \text{derivadas con respecto a } \lambda \text{ de } L: & A^{\mathrm{T}}\boldsymbol{x} \qquad = \boldsymbol{b} \end{array}} \tag{6}$$

Las primeras ecuaciones dan $\boldsymbol{x} = -S^{-1}A\boldsymbol{\lambda}$. Luego las segundas ecuaciones dan $-A^{\mathrm{T}}S^{-1}A\boldsymbol{\lambda} = \boldsymbol{b}$. Esto determina el $\boldsymbol{\lambda}^*$ óptimo, y por lo tanto el $\boldsymbol{x}^*$ óptimo :

$$\textbf{Solución } \boldsymbol{\lambda^*, x^*} \qquad \boldsymbol{\lambda^* = -(A^{\mathrm{T}}S^{-1}A)^{-1}b} \qquad \boldsymbol{x^* = S^{-1}A(A^{\mathrm{T}}S^{-1}A)^{-1}b.} \tag{7}$$

Costo mínimo $F^* = \dfrac{1}{2}(\boldsymbol{x}^*)^{\mathrm{T}}S\boldsymbol{x}^* = \dfrac{1}{2}\boldsymbol{b}^{\mathrm{T}}(A^{\mathrm{T}}S^{-1}A)^{-1}A^{\mathrm{T}}S^{-1}SS^{-1}A(A^{\mathrm{T}}S^{-1}A)^{-1}\boldsymbol{b}$. ¡ Esto se simplifica muchísimo !

$$\boxed{\begin{array}{l} \textbf{Costo mínimo } \boldsymbol{F^* = \dfrac{1}{2}b^{\mathrm{T}}(A^{\mathrm{T}}S^{-1}A)^{-1}b} \\ \textbf{Gradiente del costo } \boldsymbol{\dfrac{\partial F^*}{\partial b} = (A^{\mathrm{T}}S^{-1}A)^{-1}b = -\lambda^*} \end{array}} \tag{8}$$

Este es un verdadero problema modelo. Cuando las restricciones cambian a una desigualdad $A^{\mathrm{T}}\boldsymbol{x} \leq \boldsymbol{b}$, los multiplicadores se vuelven $\lambda_i \geq 0$ y los problemas se vuelven más difíciles.

Regresemos a la "matriz de punto de ensilladura" o la "matriz KKT" de la ecuación (6) :

$$M\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{S} & \boldsymbol{A} \\ \boldsymbol{A^{\mathrm{T}}} & \boldsymbol{0} \end{bmatrix}\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{b} \end{bmatrix}. \tag{9}$$

Esa matriz M no es definida positiva ni definida negativa. Supongamos que multiplicamos el primer bloque fila $[S \;\; A]$ por $A^{\mathrm{T}}S^{-1}$ para obtener $[A^{\mathrm{T}} \;\; A^{\mathrm{T}}S^{-1}A]$. Restamos del segundo bloque fila para tener un bloque cero :

$$\begin{bmatrix} \boldsymbol{S} & \boldsymbol{A} \\ \boldsymbol{0} & \boldsymbol{-A^{\mathrm{T}}S^{-1}A} \end{bmatrix}\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{b} \end{bmatrix}. \tag{10}$$

Esto es tan solo eliminación en la matriz bloque M de 2 por 2. A ese nuevo bloque en la posición $2,2$ lo llamamos el **complemento de Schur** (así nombrado en honor al mejor algebrista lineal que jamás existió).

Llegamos a la misma ecuación que antes: $-A^{\mathrm{T}}S^{-1}A\boldsymbol{\lambda} = \boldsymbol{b}$. La eliminación es solo una manera organizada de resolver ecuaciones lineales. Los primeros n pivotes son positivos porque S es definida **positiva**. Ahora tendremos m pivotes negativos porque $-A^{\mathrm{T}}S^{-1}A$ es definida **negativa**. Esta es una señal inconfundible de un **punto de ensilladura en el lagrangiano $\boldsymbol{L(x,\lambda)}$**.

¡ Esa función $L = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x} + \boldsymbol{\lambda}^{\mathrm{T}}(A^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{b})$ es **convexa en $\boldsymbol{x}$ y cóncava en $\boldsymbol{\lambda}$** !

Minimax = Maximin

Tenemos más para aprender de este problema. La derivada con respecto a $\boldsymbol{x}$ de L y la derivada con respecto a $\boldsymbol{\lambda}$ de L se igualaron a cero en la ecuación (6). Resolvimos esas dos ecuaciones para $\boldsymbol{x}^*$ y $\boldsymbol{\lambda}^*$. Ese par $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ es un punto de ensilladura de L en la Ecuación (7). Al resolver (7) encontramos el costo mínimo y su derivada en (8).

Supongamos que separamos esto en dos problemas : un problema de mínimo y uno de máximo. Primero minimizamos $L(\boldsymbol{x}, \boldsymbol{\lambda})$ para cada λ fijo. El $\boldsymbol{x}^*$ minimizante depende de $\boldsymbol{\lambda}$. Luego encontramos $\boldsymbol{\lambda}^*$ que maximiza $L(\boldsymbol{x}^*(\boldsymbol{\lambda}), \boldsymbol{\lambda})$.

Minimizar L en $\boldsymbol{x}^* = -S^{-1}A\boldsymbol{\lambda}$ En ese punto $\boldsymbol{x}^*$, $\min L = -\dfrac{1}{2}\boldsymbol{\lambda}^{\mathrm{T}}A^{\mathrm{T}}S^{-1}A\boldsymbol{\lambda} - \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{b}$

Maximizar ese mínimo $\boldsymbol{\lambda}^* = -(A^{\mathrm{T}}S^{-1}A)^{-1}\boldsymbol{b}$ da $L = \dfrac{1}{2}\boldsymbol{b}^{\mathrm{T}}(A^{\mathrm{T}}S^{-1}A)^{-1}\boldsymbol{b}$

$$\boxed{\max_{\boldsymbol{\lambda}} \min_{\boldsymbol{x}} L = \frac{1}{2}\,\boldsymbol{b}^{\mathrm{T}}(A^{\mathrm{T}}S^{-1}A)^{-1}\,\boldsymbol{b}}$$

Este *maximin* fue primero en $\boldsymbol{x}$ y luego en $\boldsymbol{\lambda}$. El orden inverso es *minimax* : primero en $\boldsymbol{\lambda}$, luego en $\boldsymbol{x}$.

El máximo en $\boldsymbol{\lambda}$ de $L(\boldsymbol{x}, \boldsymbol{\lambda}) = \dfrac{1}{2}\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x} + \boldsymbol{\lambda}^{\mathrm{T}}(A^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{b})$ es $\begin{cases} +\infty \text{ si } A^{\mathrm{T}}\boldsymbol{x} \neq \boldsymbol{b} \\ \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x} \text{ si } A^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{b} \end{cases}$

El mínimo en $\boldsymbol{x}$ de ese máximo en $\boldsymbol{\lambda}$ es nuestra respuesta $\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}(A^{\mathrm{T}}S^{-1}A)^{-1}\,\boldsymbol{b}$.

$$\boxed{\min_{\boldsymbol{x}} \max_{\boldsymbol{\lambda}} L = \frac{1}{2}\,\boldsymbol{b}^{\mathrm{T}}(A^{\mathrm{T}}S^{-1}A)^{-1}\,\boldsymbol{b}}$$

En el punto de ensilladura $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ tenemos $\dfrac{\boldsymbol{\partial L}}{\boldsymbol{\partial x}} = \dfrac{\boldsymbol{\partial L}}{\boldsymbol{\partial \lambda}} = \boldsymbol{0}$ **y** $\max\limits_{\boldsymbol{\lambda}} \min\limits_{\boldsymbol{x}} \boldsymbol{L} = \min\limits_{\boldsymbol{x}} \max\limits_{\boldsymbol{\lambda}} \boldsymbol{L}$.

Problemas duales en la ciencia y la ingeniería

Minimizar una cuadrática $\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}$ con una restricción lineal $A^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{b}$ no es solo un ejercicio abstracto, sino que es el problema central de la matemática aplicada a la física: cuando se discretiza una ecuación diferencial. He aquí dos ejemplos importantes.

1 **Ecuaciones de redes para circuitos eléctricos**

Incógnitas : voltajes en los nodos, corrientes a lo largo de las aristas

Ecuaciones : las leyes de Kirchhoff y la ley de Ohm

Matrices : $A^{\mathrm{T}}S^{-1}A$ es la **matriz de conductancia**.

2 **Método de elementos finitos para estructuras**

Incógnitas : desplazamientos en los nodos, tensiones en la estructura

Ecuaciones : balance de fuerzas y relaciones fuerza-deformación

Matrices : $A^{\mathrm{T}}S^{-1}A$ es la **matriz de rigidez**.

La lista completa se extendería a todos los campos de la ingeniería. La matriz de rigidez y la matriz de conductancia son simétricas definidas positivas. Normalmente las restricciones son ecuaciones de igualdad y no desigualdades. Así la matemática ofrece tres modos de abarcar el modelado del problema físico :

(i) Ecuaciones lineales con la matriz de rigidez o matriz de conductancia o matriz del sistema

(ii) Minimización donde las incógnitas $\boldsymbol{x}$ son las corrientes o las tensiones

(iii) Maximización donde las incógnitas $\boldsymbol{\lambda}$ son los voltajes o desplazamientos

En definitiva, las ecuaciones lineales (i) son la opción más popular. Reducimos la ecuación (9) a la ecuación (10). Esas ecuaciones de redes explican por qué Kirchhoff y Ohm aparecen juntos. Las ecuaciones estructurales explican el balance de la fuerza y las propiedades del material. Todas las leyes eléctricas y mecánicas están incorporadas en el sistema final.

Para problemas de mecánica de fluidos, ese sistema de ecuaciones a menudo se encuentra en forma de punto de ensilladura. Las incógnitas $\boldsymbol{x}$ y $\boldsymbol{\lambda}$ son velocidades y presiones. El análisis numérico está bien desarrollado en *Finite Elements and Fast Iterative Solvers* de Elman, Silvester y Wathen.

Para ecuaciones de redes y ecuaciones de elementos finitos que llevan a matrices de conductancia y matrices de rigidez $A^{\mathrm{T}}CA$, una referencia es mi libro *Computational Science and Engineering*, que trata sobre la ciencia computacional y la ingeniería. Las clases en formato vídeo para 18.085 están publicadas en **ocw.mit.edu**.

En estadística y cuadrados mínimos (regresión lineal), la matriz $A^{\mathrm{T}}\Sigma^{-1}A$ incluye a $\Sigma =$ matriz de covarianza. Dividimos por las varianzas σ^2 para blanquear el ruido.

Para problemas no lineales, la energía ya no es más una cuadrática $\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}$. Las *no linealidades geométricas* aparecen en la matriz A. Las *no linealidades materiales* (a menudo más simples) aparecen en la matriz C. Desplazamientos grandes y tensiones grandes son una fuente típica de no linealidad.

Problemas de la Sección VI.2

1 Minimice $F(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x} = \frac{1}{2}x_1^2 + 2x_2^2$ sujeto a $A^{\mathrm{T}}\boldsymbol{x} = x_1 + 3x_2 = b$.

(a) ¿ Cuál es el lagrangiano $L(\boldsymbol{x}, \lambda)$ para este problema ?

(b) ¿ Cuáles son las tres ecuaciones "derivada de L = cero" ?

(c) Resuelva esas ecuaciones para encontrar $\boldsymbol{x}^* = (x_1^*, x_2^*)$ y el multiplicador λ^*.

(d) Dibuje la Figura VI.4 para este problema con la recta de restricción tangente al círculo del costo.

(e) Verifique que la derivada del costo mínimo es $\partial F^*/\partial b = -\lambda^*$.

2 Minimice $F(\boldsymbol{x}) = \frac{1}{2}\left(x_1^2 + 4x_2^2\right)$ sujeta a $2x_1 + x_2 = 5$. Encuentre y resuelva las tres ecuaciones $\partial L/\partial x_1 = 0$, $\partial L/\partial x_2 = 0$ y $\partial L/\partial \lambda = 0$. Dibuje la recta de restricción $2x_1 + x_2 = 5$ tangente a la elipse $\frac{1}{2}\left(x_1^2 + 4x_2^2\right) = F_{\min}$ en el punto mínimo (x_1^*, x_2^*).

3 La matriz de punto de ensilladura del Problema 1 es

$$M = \begin{bmatrix} S & A \\ A^{\mathrm{T}} & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 4 & 3 \\ 1 & 3 & 0 \end{bmatrix}.$$

Reduzca M a una matriz triangular U por eliminación y verifique que

$$U = \begin{bmatrix} S & A \\ 0 & -A^{\mathrm{T}}S^{-1}A \end{bmatrix}.$$

¿ Cuántos pivotes positivos tiene M ? ¿ Cuántos autovalores positivos tiene M ?

4 Para una matriz simétrica invertible S cualquiera, *el número de pivotes positivos equivale al número de autovalores positivos*. Los pivotes aparecen en $S = LDL^{\mathrm{T}}$ (L triangular). Los autovalores aparecen en $S = Q\Lambda Q^{\mathrm{T}}$ (Q orthogonal). Una prueba linda envía L y Q a I o $-I$ *sin volverse singular en el camino* (vea los Problemas de la sección III.2). Los autovalores se mantienen reales y no cruzan el cero. Así sus signos son los mismos en D y Λ.

Demuestre esta "ley de inercia" para toda matriz simétrica invertible S de 2 por 2: si $\boldsymbol{S}$ tiene **0, 1 o 2** pivotes positivos **$\boldsymbol{S}$ tiene 0, 1 o 2 autovalores positivos.**

1. Tome el determinante de $LDL^{\mathrm{T}} = Q\Lambda Q^{\mathrm{T}}$ para mostrar que $\det D$ y $\det \Lambda$ tienen el mismo signo. Si el determinante es negativo, entonces S tiene __ autovalores positivos en Λ y __ pivotes positivos en D.
2. Si el determinante es positivo, entonces S podría ser definido positivo o negativo. Muestre que ambos pivotes son positivos cuando ambos autovalores son positivos.

5 Encuentre el valor mínimo de $F(\boldsymbol{x}) = \frac{1}{2}\left(x_1^2 + x_2^2 + x_3^2\right)$ con una restricción $x_1 + x_2 + x_3 = 3$ y luego con una restricción adicional $x_1 + 2x_2 + 3x_3 = 12$. El segundo valor mínimo debería ser mayor que el primer valor mínimo: ¿ Por qué ? El primer problema tiene una _____ tangente a una esfera en $\mathbf{R}^3$. El segundo problema tiene una _____ tangente a una esfera en $\mathbf{R}^3$.

VI.3. Programación lineal, teoría de juegos y dualidad

Esta sección trata sobre problemas de optimización altamente estructurados. Primero viene la programación lineal: costo lineal y restricciones lineales (incluyendo desigualdades). Desde un punto de vista histórico, también fue la primera que surgió, cuando Dantzig inventó el algoritmo simplex para encontrar la solución óptima. Nuestro enfoque será ver la "dualidad" entre un problema de mínimo y uno de máximo: *dos programas lineales que se resuelven al mismo tiempo.*

Una restricción de desigualdad $\boldsymbol{x_k \geq 0}$ tiene dos estados: activo e inactivo. Si la solución minimizante termina con $x_k^* > 0$, entonces ese requisito estaba inactivo, no cambiaba nada. Su multiplicador de Lagrange tendrá $\lambda_k^* = 0$. El costo mínimo no se ve afectado por esa restricción en x_k. Pero si la restricción $x_k \geq 0$ fuerza activamente a que el mejor $\boldsymbol{x^*}$ tenga $x_k^* = 0$, entonces el multiplicador tendrá $\lambda_k^* > 0$. Así la condición óptima es $\boldsymbol{x_k^* \lambda_k^* = 0}$ para cada k.

Otro aspecto de la programación lineal: resuelve todos los **juegos de suma cero de 2 personas**. La ganancia para un jugador significa pérdida para el otro. Las estrategias óptimas producen un punto de ensilladura.

Las restricciones de desigualdad siguen presentes en la programación cuadrática (**QP**) y la programación semidefinida (**SDP**). Las restricciones en SDP involucran matrices simétricas. La desigualdad $S \geq 0$ significa que la matriz es semidefinida (o definida) positiva. Si la mejor S es en realidad definida positiva, entonces la restricción $S \geq 0$ no estaba activa y el multiplicador de Lagrange (ahora también una matriz) será cero.

Programación lineal

La programación lineal comienza con un vector de costo $\boldsymbol{c} = (c_1, \ldots, c_n)$. El problema es minimizar el costo $F(\boldsymbol{x}) = c_1 x_1 + \cdots + c_n x_n = \boldsymbol{c}^{\mathrm{T}} \boldsymbol{x}$. Las restricciones son m ecuaciones lineales $A\boldsymbol{x} = \boldsymbol{b}$ y n desigualdades $x_1 \geq 0, \ldots, x_n \geq 0$. Tan solo escribimos $\boldsymbol{x} \geq 0$ para incluir todas las n componentes:

Programación lineal

$$\boxed{\textbf{Minimizar } \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \textbf{ sujeto a } \boldsymbol{Ax = b} \textbf{ y } \boldsymbol{x \geq 0}} \tag{1}$$

Si A es de 1 por 3, $A\boldsymbol{x} = \boldsymbol{b}$ da un plano de la forma $x_1 + x_2 + 2x_3 = 4$ en un espacio tridimensional. Ese plano se verá recortado por las restricciones $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$. Esto nos deja un triángulo en el plano, con vértices en $(x_1, x_2, x_3) = (4, 0, 0)$, $(0, 4, 0)$ y $(0, 0, 2)$. Nuestro problema es encontrar el punto $\boldsymbol{x^*}$ en este triángulo que minimiza el costo $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$.

Debido a que el costo es lineal, **su mínimo se alzanza en uno de esos vértices**. La programación lineal tiene que encontrar ese vértice de costo mínimo. Calcular todos los vértices es exponencialmente impráctico cuando m y n son grandes. Así el *método simplex* encuentra un vértice inicial que satisface $A\boldsymbol{x} = \boldsymbol{b}$ y $\boldsymbol{x} \geq \boldsymbol{0}$. Luego se desplaza a lo largo de una arista del conjunto de restricciones K a otro vértice (de menor costo). El costo $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ disminuye en cada paso.

Encontrar la arista de pendiente máxima y el siguiente vértice (donde esa arista termina) es un problema de álgebra lineal. El método simplex repite este paso muchas veces, de vértice a vértice.

Nuevo vértice inicial, nueva arista de pendiente máxima, nuevo vértice de menor costo en el método simplex. En la práctica, la cantidad de pasos es polinomial (pero en teoría podría ser exponencial).

Nuestro interés aquí es identificar el **problema dual**, un problema de máximo para $\boldsymbol{y}$ en $\mathbf{R}^m$. Es usual utilizar $\boldsymbol{y}$ en lugar de $\boldsymbol{\lambda}$ para las incógnitas duales: los multiplicadores de Lagrange.

$$\textbf{Problema dual} \qquad \boxed{\textbf{Maximizar } \boldsymbol{y}^{\mathrm{T}}\boldsymbol{b} \textbf{ sujeto a } \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} \leq \boldsymbol{c}.} \tag{2}$$

Este es otro programa lineal para resolver por el método simplex. Tiene los mismos elementos $A, \boldsymbol{b}, \boldsymbol{c}$ que antes. Cuando la matriz A es de m por n, la matriz A^{T} es de n por m. Entonces $A^{\mathrm{T}}\boldsymbol{y} \leq \boldsymbol{c}$ tiene n restricciones. Un dato bello: $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{b}$ en el problema de máximo nunca es mayor que $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ en el problema de mínimo.

$$\textbf{Dualidad débil} \quad \boldsymbol{y}^{\mathrm{T}}\boldsymbol{b} = \boldsymbol{y}^{\mathrm{T}}(A\boldsymbol{x}) = (A^{\mathrm{T}}\boldsymbol{y})^{\mathrm{T}}\boldsymbol{x} \leq \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \quad \textbf{máximo (2)} \leq \textbf{mínimo (1)}$$

Maximizar empuja $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{b}$ hacia arriba. Minimizar empuja $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ hacia abajo. El gran teorema de la dualidad (**el teorema minimax**) establece que se encuentran en el mejor $\boldsymbol{x}^*$ y el mejor $\boldsymbol{y}^*$.

$$\boxed{\textbf{Dualidad} \qquad \textbf{El máximo de } \boldsymbol{y}^{\mathrm{T}}\boldsymbol{b} \textbf{ es igual al mínimo de } \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}.}$$

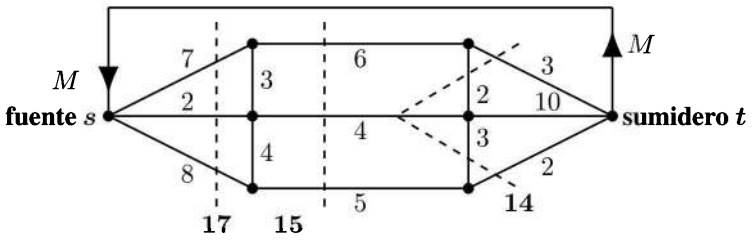
El método simplex resolverá ambos problemas a la vez. Por muchos años, no ha habido métodos que compitieran con este. Ahora ya no es así. Existen algoritmos nuevos que *viajan por dentro* del conjunto de $\boldsymbol{x}$ permitidos en lugar de desplazarse por sus aristas (de vértice a vértice). Los *métodos de puntos interiores* son competitivos porque usan el cálculo para lograr el descenso más pronunciado.

La situación actual indica que cualquiera de estos dos métodos podría ganar: moviéndose por las aristas o por el interior.

Fujo máximo-corte mínimo

He aquí un programa lineal especial. La matriz A será la matriz de incidencia de un grafo. Esto significa que en cada nodo, el flujo que ingresa es igual al flujo que sale. Cada arista del grafo tiene una *capacidad* M_j, que el flujo y_j a lo largo de esa arista no puede exceder.

El problema de máximo es enviar el mayor flujo posible desde el nodo fuente s al nodo sumidero t. Este flujo regresa de t a s en una arista especial con capacidad ilimitada, que está dibujada en la siguiente página. Las restricciones en $\boldsymbol{y}$ son: la ley de corriente de Kirchhoff $A^{\mathrm{T}}\boldsymbol{y} = 0$ y las cotas de capacidad $|y_j| \leq M_j$ en cada arista del grafo. La belleza de este ejemplo es que podemos resolverlo con sentido común (para un grafo pequeño). En el proceso, descubrimos y resolvemos el problema de mínimo dual: **el corte mínimo**.



Problema de flujo máximo
Maximizar M
con flujo $y_j \leq M_j$
La capacidad M_j
se muestra en cada arista

Figura VI.5: el flujo máximo M está acotado por la capacidad de cualquier corte (línea de puntos). *Por dualidad, la capacidad del* ***corte mínimo*** *es igual al* ***flujo máximo*** : $M = \mathbf{14}$.

Comenzamos enviando flujo desde el origen. Las tres aristas de s tienen capacidad $7 + 2 + 8 = \mathbf{17}$. ¿ Existe una cota más ajustada que $M \leq 17$?

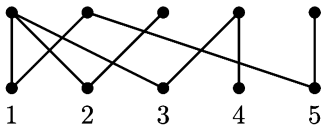
Sí, un corte a través de las tres aristas del medio solo tiene capacidad $6 + 4 + 5 = \mathbf{15}$. Por lo tanto 17 no pueden alcanzar el sumidero. ¿ Existe una cota más ajustada que $M \leq 15$?

Sí, un corte a través de las cinco aristas posteriores solo tiene capacidad $3+2+4+3+2 = \mathbf{14}$. El flujo total M no puede exceder 14. ¿ Se puede lograr ese flujo de 14 y es este el corte más ajustado ?

Sí, este es el corte mínimo (*es un problema* ℓ^1) y por dualidad 14 es el flujo máximo.

Wikipedia presenta una lista creciente de algoritmos cada vez más rápidos para resolver este importante problema. Tiene muchas aplicaciones. Si las capacidades M_j son enteras, los flujos óptimos y_j son enteros. Normalmente los programas enteros son muy difíciles, pero no es así en este caso.

Un problema de flujo máximo particular tiene todas las capacidades $M_j = 1$ o 0. El grafo es **bipartito** (todas las aristas van de un nodo en la parte 1 a un nodo en la parte 2). Relacionamos personas en la parte 1 con tareas en la parte 2 (a lo sumo una persona por tarea y viceversa). Así la relación máxima es $M =$ flujo máximo en el grafo $=$ número máximo de asignaciones.



Este grafo bipartito permite una relación perfecta : $M = 5$. Quitamos la arista que baja de 2 a 1. Ahora solo son posibles $M = 4$ asignaciones, porque 2 y 5 solo están calificados para una tarea (5).

Para grafos bipartitos, flujo máximo $=$ corte mínimo es el teorema de König y el teorema del matrimonio de Hall.

Juegos de dos jugadores

Los juegos con tres o más jugadores son muy difíciles de resolver. Grupos de jugadores se unen contra otros y esas alianzas son inestables. A menudo se formarán equipos nuevos. Fue John Nash quien logró hacer un avance importante, y por esto recibió un premio Nobel (¡ en economía !). Pero los juegos de suma cero de dos jugadores fueron resueltos por completo por von Neumann. Veremos su relación estrecha con la programación lineal y la dualidad.

> Los jugadores son X e Y. Tenemos la matriz de pago A. En cada turno, el jugador X elige una fila de A y el jugador Y elige una columna. El número en esa fila y columna de A es el pago, y luego los jugadores juegan otro turno.

Para relacionarlo con la programación lineal, haré que el pago vaya del jugador X al jugador Y. Así X quiere minimizar e Y quiere maximizar. He aquí una matriz de pago muy pequeña. Tiene dos filas para que X elija y tres columnas para Y.

Matriz de pago

	y_1	y_2	y_3
x_1	**1**	**0**	**2**
x_2	**3**	**−1**	**4**

A Y le gustan esos números grandes en la columna 3. X nota que el menor número en esa columna es 2 (en la fila 1). No hay motivos por los cuales estos jugadores se desviaríen de esta estrategia sencilla de columna 3 para Y y fila 1 para X. El pago de 2 es de X a Y:

2 es el más pequeño de su columna y el mayor de su fila

Esto es un **punto de ensilladura**. Y no puede esperar ganar más que 2. X no puede esperar perder menos que 2. Todas las jugadas serán las mismas porque ningún jugador está incentivado a cambiar. Las estrategias óptimas $\boldsymbol{x^*}$ e $\boldsymbol{y^*}$ son claras: fila 1 para X y columna 3 para Y.

Pero un cambio en la columna 3 requerirá que los jugadores reconsideren.

Nueva matriz de pago

	y_1	y_2	y_3
x_1	**1**	**0**	**4**
x_2	**3**	**−1**	**2**

A X le gustan esos números pequeños y favorables en la columna 2. Pero Y jamás eligiría esa columna. La columna 3 es la que mejor (más grande) se ve para Y, y X debería contrarrestar eligiendo la fila 2 (para evitar pagar 4). Pero entonces la columna 1 se vuelve mejor que la columna 3 para Y, porque ganar 3 en la columna 1 es mejor que ganar 2.

Vemos que Y todavía quiere la columna 3 pero a veces debe elegir la columna 1. De manera similar X debe usar una *estrategia mixta*: elegir las filas 1 y 2 con probabilidades x_1 y x_2. La elección en cada turno debe ser impredecible o el otro jugador tomará ventaja. Entonces la elección para X tiene dos probabilidades $x_1 \geq 0$ y $x_2 \geq 0$ que suman $x_1 + x_2 = 1$. La matriz de pago ahora tiene una fila nueva gracias a esta estrategia mixta:

fila 1	1	0	4
fila 2	3	−1	2
$\boldsymbol{x_1}$**(fila 1)** $+ \boldsymbol{x_2}$**(fila 2)**	$\boldsymbol{x_1 + 3x_2}$	$\boldsymbol{-x_2}$	$\boldsymbol{4x_1 + 2x_2}$

X elegirá fracciones x_1 y x_2 para que el peor (*mayor*) pago sea lo más pequeño posible. Si recordamos que $x_2 = 1 - x_1$, esto ocurrirá cuando los dos mayores pagos sean iguales:

$\boldsymbol{x_1 + 3x_2 = 4x_1 + 2x_2}$ **significa que** $\boldsymbol{x_1 + 3(1 - x_1) = 4x_1 + 2(1 - x_1)}$.

Esa ecuación da $\boldsymbol{x_1^* = \frac{1}{4}}$ **y** $\boldsymbol{x_2^* = \frac{3}{4}}$**. La nueva fila mixta es** $\boldsymbol{2.5, -.75, 2.5}$.

De manera similar Y elegirá las columnas $1, 2, 3$ con probabilidades y_1, y_2, y_3. De nuevo suman 1. Esta estrategia mixta combina las tres columnas de A en una nueva columna para Y.

columna 1	columna 2	columna 3	**mezcla** $\boldsymbol{1, 2, 3}$
1	0	4	$\boldsymbol{y_1 + 4y_3}$
3	−1	2	$\boldsymbol{3y_1 - y_2 + 2y_3}$

Y elegirá las fracciones $y_1 + y_2 + y_3 = 1$ para obtener el peor (*menor*) pago tan grande como sea posible. Esto ocurre cuando $y_2 = 0$ e $y_3 = 1 - y_1$. Los dos pagos mixtos son iguales :

$$y_1 + 4(1 - y_1) = 3y_1 + 2(1 - y_1) \quad \text{da} \quad -3y_1 + 4 = y_1 + 2 \quad \text{y} \quad \boldsymbol{y_1^* = y_3^* = \frac{1}{2}}.$$

La nueva columna mixta tiene 2.5 en ambas componentes. Estas estrategias óptimas identifican 2.5 como el valor del juego. Con la estrategia mixta $x_1^* = \frac{1}{4}$ y $x_2^* = \frac{3}{4}$, está garantizado que el jugador X no pagará más de 2.5. Está garantizado que el jugador Y no recibirá menos de 2.5. Hemos encontrado el punto de ensilladura (mejores estrategias mixtas, con pago minimax de $X =$ pago maximín a $Y = \mathbf{2.5}$) de este juego de dos jugadores.

	y_1	y_2	y_3	$\frac{1}{2}$**col 1** $+ \frac{1}{2}$**col 2**
fila 1	1	0	4	**2.5**
fila 2	3	−1	2	**2.5**
$\frac{1}{4}$ **fila 1** $+ \frac{3}{4}$**fila 2**	**2.5**	−.75	**2.5**	

El **teorema minimax** de Von Neumann para juegos da una solución para cada matriz de pago. Es equivalente al teorema de dualidad *min* $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} =$ *max* $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{b}$ de la programación lineal.

Programación semidefinida (SDP)

El costo a minimizar sigue siendo $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$: costo lineal. Pero ahora las restricciones en $\boldsymbol{x}$ usan matrices simétricas S. Dadas S_0 a S_n, se requiere que $S(\boldsymbol{x}) = S_0 + x_1S_1 + \cdots + x_nS_n$ sea *semidefinida (o definida) positiva*. Por suerte este es un conjunto convexo de $\boldsymbol{x}$: el promedio de dos matrices semidefinidas es semidefinido. (Tan solo promediamos las dos energías $\boldsymbol{v}^{\mathrm{T}}S\boldsymbol{v} \geq 0$.)

Ahora el conjunto de $\boldsymbol{x}$ permitidos podría tener lados curvos en lugar de planos :

$$S_0 + x_1S_1 + x_2S_2 = \begin{bmatrix} x_1 & 1 \\ 1 & x_2 \end{bmatrix} \text{ es semidefinida positiva cuando } \boldsymbol{x_1 \geq 0} \text{ y } \boldsymbol{x_1x_2 \geq 1}.$$

Minimizar el máximo autovalor de $S(\boldsymbol{x})$ se considera utilizando una variable adicional t :

Minimizar t de manera que $tI - S(\boldsymbol{x})$ sea semidefinida positiva.

Y la SDP podría también minimizar el mayor valor singular, la norma L^2 de $S(\boldsymbol{x})$:

$$\text{Minimizar } t \text{ de manera que } \begin{bmatrix} tI & S(\boldsymbol{x}) \\ S(\boldsymbol{x})^{\mathrm{T}} & tI \end{bmatrix} \text{ sea semidefinida positiva.}$$

Para la mayoría de los problemas semidefinidos, los métodos de punto interior son los mejores. No viajamos por la frontera del conjunto de restricciones (de vértice a vértice, como el método simplex en los programas lineales). En cambio, viajamos por el interior del conjunto. Esencialmente resolvemos un problema de mínimos cuadrados en cada iteración: generalmente entre 5 y 50 iteraciones.

Al igual que en la programación lineal, hay un **problema dual** (una maximización). El valor de este dual está siempre por *debajo* del valor $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ del original. Cuando maximizamos en el dual y minimizamos $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ en el primal, esperamos lograr que esos valores sean iguales. Pero esto podría no ocurrir para programas semidefinidos con desigualdades matriciales.

La SDP da un método de resolución a problemas matriciales que antes parecían demasiado difíciles.

Problemas de la Sección VI.3

1 ¿ Se necesita la restricción $\boldsymbol{x} \geq \mathbf{0}$ en la ecuación (3) para una dualidad débil ? ¿ La desigualdad $A^{\mathrm{T}}\boldsymbol{y} \leq \boldsymbol{c}$ ya es suficiente para mostrar que $(A^{\mathrm{T}}\boldsymbol{y})^{\mathrm{T}}\boldsymbol{x} \leq \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$? No lo creo.

2 Consideremos las restricciones $x_1 + x_2 + 2x_3 = 4$ y $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$. Encuentre los tres vértices de este triángulo en $\mathbf{R}^3$. ¿ Cuál vértice minimiza el costo $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} = 5x_1 + 3x_2 + 8x_3$?

3 ¿ Cuál problema de máximo para y es dual al Problema 2 ? Una restricción en el problema primal significa una incógnita y en el problema dual. Resuelva este problema dual.

4 Consideremos las restricciones $\boldsymbol{x} \geq \mathbf{0}$, $x_1 + 2x_3 + x_4 = 4$ y $x_2 + x_3 - x_4 = 2$. Tenemos dos restricciones de igualdad con cuatro incógnitas, entonces un vértice como $\boldsymbol{x} = (0, 6, 0, 4)$ tiene $4 - 2 = 2$ ceros. Encuentre otro vértice con $\boldsymbol{x} = (x_1, x_2, 0, 0)$ y muestre que cuesta más que el primer vértice.

5 Encuentre la estrategia óptima (minimizante) para que X elija filas. Encuentre la estrategia óptima (maximizante) para que Y elija columnas. ¿ Cuál es el pago de X a Y en este punto minimax óptimo $\boldsymbol{x}^*, \boldsymbol{y}^*$?

$$\textbf{Matrices de pago} \qquad \begin{bmatrix} 1 & 2 \\ 4 & 8 \end{bmatrix} \qquad \begin{bmatrix} 1 & 4 \\ 8 & 2 \end{bmatrix}$$

6 Si $A^{\mathrm{T}} = -A$ (matriz de pago antisimétrica), ¿ por qué resulta un juego justo para X e Y con pago minimax igual a cero ?

7 Suponga que la matriz de pago es una matriz diagonal Σ con elementos $\sigma_1 > \sigma_2 > \cdots > \sigma_n$. ¿ Qué estrategias son óptimas para X e Y ?

8 Convierta $||(x_1, x_2, x_3)||_1 \leq 2$ en la **norma $\boldsymbol{\ell^1}$** en ocho desigualdades lineales $A\boldsymbol{x} \leq \boldsymbol{b}$. La restricción $||\boldsymbol{x}||_\infty \leq 2$ en la **norma $\boldsymbol{\ell^\infty}$** también produce ocho desigualdades lineales.

9 En la **norma $\boldsymbol{\ell^2}$**, $||\boldsymbol{x}|| \leq 2$ es una desigualdad cuadrática $x_1^2 + x_2^2 + x_3^2 \leq 4$. Pero en programación semidefinida (SDP) se vuelve una desigualdad matricial $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} \leq 4I$.

¿ Por qué la restricción $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} \leq 4I$ es equivalente a $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} \leq 4$?

Nota La dualidad ofrece una opción importante : resolver el problema primal *o el dual*. Esto se aplica a la optimización en aprendizaje automático, como expresa este artículo :

F. Bach, *Duality between subgradient and conditional gradient methods*, SIAM Journal of Optimization **25** (2015) 115-129; arXiv : 1211.6302.

VI.4. Descenso por gradiente hacia el mínimo

Esta sección del libro trata un problema fundamental: **minimizar una función $f(x_1, \ldots, x_n)$.** El cálculo nos enseña que todas las derivadas primeras $\partial f/\partial x_i$ son cero en el mínimo (cuando f es suave). Si tenemos $n = 20$ incógnitas (un número pequeño en aprendizaje profundo), entonces minimizar una función f se traduce a 20 ecuaciones $\partial f/\partial x_i = 0$. *El "descenso por gradiente" usa las derivadas $\partial f/\partial x_i$ para encontrar una dirección que reduzca $f(\boldsymbol{x})$.* La dirección de mayor pendiente, en la que $f(\boldsymbol{x})$ decrece con mayor velocidad, está dada por el gradiente $-\nabla \boldsymbol{f}$:

$$\boxed{\textbf{Descenso por gradiente} \qquad \boldsymbol{x_{k+1} = x_k - s_k \nabla f(x_k)}} \tag{1}$$

El símbolo $\nabla \boldsymbol{f}$ representa el vector de n derivadas parciales de primer orden de f: su **gradiente**. Así (1) es una ecuación vectorial en cada paso $k = 1, 2, 3, \ldots$ y s_k es el *tamaño del paso* o la *tasa de aprendizaje*. Esperamos movernos hacia el punto $\boldsymbol{x^*}$ donde el gráfico de $f(\boldsymbol{x})$ toca fondo.

Estamos dispuestos a suponer por ahora que las 20 derivadas primeras existen y pueden ser calculadas. No estamos dispuestos a suponer que esas 20 funciones también tienen 20 derivadas $\partial/\partial x_j(\partial f/\partial x_i)$. Estas son las 210 **derivadas segundas** de f, que se colocan en una matriz simétrica H de 20 por 20. (La simetría reduce $n^2 = 400$ a $\frac{1}{2}n^2 + \frac{1}{2}n = 210$ cálculos). Las derivadas segundas suelen contener información adicional muy útil, pero en muchos problemas debemos trabajar sin ellas.

Debería saber que 20 derivadas primeras y 210 derivadas segundas no multiplican el costo de cálculo por 20 y 210. Esta linda idea de **diferenciación automática**, redescubierta y popularizada como **retropropagación** en el aprendizaje automático, reduce significativamente los costos en la práctica. La idea está descrita en la Sección VII.2.

Regresemos por un momento a la ecuación (1). El paso $-s_k \nabla \boldsymbol{f}(\boldsymbol{x_k})$ incluye un signo menos (para descender) y un factor s_k (para controlar el tamaño del paso) y el vector gradiente $\nabla \boldsymbol{f}$ (que contiene las derivadas primeras de f calculadas en el punto actual $\boldsymbol{x_k}$). Se ha pensado mucho y obtenido mucha experiencia computacional sobre la elección del tamaño de paso y dirección de búsqueda.

Comenzamos con los aspectos principales del cálculo de derivadas y vectores gradientes $\nabla \boldsymbol{f}$.

La derivada de $f(x)$: $n = 1$

La derivada de $f(x)$ conlleva un *límite*, esta es la diferencia clave entre cálculo y álgebra. Comparamos los valores de f en dos puntos cercanos x y $x + \Delta x$, a medida que Δx se acerca a cero. Para ser más precisos, observamos la pendiente $\Delta f/\Delta x$ entre dos puntos en el gráfico de $f(x)$:

$$\textbf{Derivada de } \boldsymbol{f} \textbf{ en } \boldsymbol{x} \qquad \frac{\boldsymbol{df}}{\boldsymbol{dx}} = \textbf{límite de } \frac{\boldsymbol{\Delta f}}{\boldsymbol{\Delta x}} = \textbf{límite de } \left[\frac{\boldsymbol{f(x + \Delta x) - f(x)}}{\boldsymbol{\Delta x}}\right]. \tag{2}$$

Es una diferencia adelantada cuando Δx es positivo y una diferencia atrasada cuando $\Delta x < 0$. Cuando tenemos el mismo límite desde ambos lados, ese número es la pendiente del gráfico en x.

La función rampa $\text{ReLU}(x) = f(x) = \max(0, x)$ juega un papel importante en el aprendizaje profundo (vea VII.1). Tiene pendientes diferentes en $x = 0$, **1** a la derecha y **0** a la izquierda. Luego la derivada df/dx no existe en ese vértice del gráfico. Para $n = 1$, df/dx es el gradiente $\nabla \boldsymbol{f}$.

$$\text{ReLU} = \begin{matrix} \boldsymbol{x} \text{ para } x \geq 0 \\ \mathbf{0} \text{ para } x \leq 0 \end{matrix} \qquad \textbf{pendiente } \frac{\Delta f}{\Delta x} = \frac{f(0+\Delta x) - f(0)}{\Delta x} = \begin{matrix} \boldsymbol{\Delta x/\Delta x = 1} \text{ si } \Delta x > 0 \\ \boldsymbol{0/\Delta x = 0} \text{ si } \Delta x < 0 \end{matrix}$$

Para la función suave $\boldsymbol{f(x) = x^2}$, el cociente $\Delta f/\Delta x$ se acercará con seguridad a la derivada df/dx desde ambos lados. Pero el acercamiento podría ser lento (solo de primer orden). Observemos de nuevo el punto $x = 0$, donde la derivada exacta $df/dx = 2x$ ahora es cero :

El cociente $\dfrac{\Delta f}{\Delta x}$ en $x = 0$ es $\dfrac{f(\Delta x) - f(0)}{\Delta x} = \dfrac{(\Delta x)^2 - 0}{\Delta x} = \boldsymbol{\Delta x}$. Así límite = pendiente = 0.

En este caso y en casi todos los casos, obtenemos un mejor cociente (más cercano a la pendiente límite df/dx) promediando la **diferencia adelantada** (donde $\Delta x > 0$) con la **diferencia atrasada** (donde $\Delta x < 0$). El promedio es una **diferencia centrada** más precisa.

$$\textbf{Centrada en } \boldsymbol{x} \quad \frac{1}{2}\left[\frac{f(x+\Delta x) - f(x)}{\Delta x} + \frac{f(x-\Delta x) - f(x)}{-\Delta x}\right] = \frac{\boldsymbol{f(x+\Delta x) - f(x-\Delta x)}}{\boldsymbol{2\,\Delta x}}$$

Para el ejemplo $f(x) = x^2$ este centramiento producirá la derivada exacta $df/dx = 2x$. En el gráfico promediamos las pendientes con Δx positivo y negativo para obtener la pendiente correcta 0 en $x = 0$. Para todas las funciones suaves, las diferencias centradas reducen el error a un tamaño $(\Delta x)^2$. Esta es una gran mejora sobre el error de tamaño Δx de las diferencias no centradas $f(x + \Delta x) - f(x)$.

Figura VI.6: función ReLU = rampa del aprendizaje profundo. La pendiente centrada de $f = x^2$ es exacta.

En la mayoría de las aproximaciones por diferencias finitas se utilizan diferencias centradas para mayor precisión. Pero todavía dividimos por un número pequeño $2\,\Delta x$. Y para una función multivariada $F(x_1, x_2, \ldots, x_n)$ necesitaremos cocientes $\Delta F/\Delta x_i$ en n direcciones distintas, posiblemente para valores grandes de n. Esos cocientes aproximan las n derivadas parciales que van en el **vector gradiente grad** $\boldsymbol{F} = \boldsymbol{\nabla F}$.

El gradiente de $\boldsymbol{F(x_1, \ldots, x_n)}$ es el vector columna $\boldsymbol{\nabla F} = \left(\dfrac{\boldsymbol{\partial F}}{\boldsymbol{\partial x_1}}, \ldots, \dfrac{\boldsymbol{\partial F}}{\boldsymbol{\partial x_n}}\right)$.

Sus componentes son las n derivadas parciales primeras de F. $\boldsymbol{\nabla F}$ apunta hacia la dirección de mayor pendiente. Los ejemplos 1-3 mostrarán la importancia de la notación vectorial ($\boldsymbol{\nabla F}$ siempre es un vector columna).

Ejemplo 1 Para un vector constante $\boldsymbol{a} = (a_1, \ldots, a_n)$, $F(\boldsymbol{x}) = \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}$ tiene gradiente $\boldsymbol{\nabla F = a}$. Las derivadas parciales de $F = a_1x_1 + \cdots + a_nx_n$ son los números $\partial F/\partial x_k = a_k$.

Ejemplo 2 Para una matriz simétrica S, el gradiente de $F(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x}$ es $\boldsymbol{\nabla F = 2\,S x}$. Para ver esto, escriba la función $F(x_1, x_2)$ para $n = 2$. La matriz S es de 2 por 2 :

$$F = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} a & b \\ b & c \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{matrix} ax_1^2 + cx_2^2 \\ +2bx_1x_2 \end{matrix} \quad \begin{bmatrix} \partial f/\partial x_1 \\ \partial f/\partial x_2 \end{bmatrix} = 2\begin{bmatrix} ax_1 + bx_2 \\ bx_1 + cx_2 \end{bmatrix} = \boldsymbol{2S}\begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \end{bmatrix}.$$

Ejemplo 3 Para una matriz simétrica definida positiva S, el mínimo de la cuadrática $F(\boldsymbol{x}) = \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x} - \boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}$ es el número negativo $\boldsymbol{F_{\text{min}} = -\frac{1}{2}\,a^{\mathrm{T}}S^{-1}a}$ en $\boldsymbol{x^* = S^{-1}a}$.

¡ Este es un ejemplo importante ! El costo mínimo ocurre donde las derivadas primeras de F son cero :

$$\boxed{\boldsymbol{\nabla F} = \begin{bmatrix} \boldsymbol{\partial F/\partial x_1} \\ \boldsymbol{:} \\ \boldsymbol{\partial F/\partial x_n} \end{bmatrix} = \boldsymbol{Sx^* - a = 0} \text{ en } \boldsymbol{x^* = S^{-1}a = \arg\min F.}} \tag{3}$$

Como siempre, **esa notación arg mín $\boldsymbol{F}$ representa el punto $\boldsymbol{x^*}$ donde se alcanza el mínimo de $\boldsymbol{F(x) = \frac{1}{2}x^{\mathrm{T}}Sx - a^{\mathrm{T}}x}$.** A menudo nos interesa más este $\boldsymbol{x^*}$ minimizante que el valor mínimo $F_{\text{min}} = F(\boldsymbol{x^*})$ en ese punto :

$$\boldsymbol{F_{\text{mín}}} \text{ es } \frac{1}{2}(S^{-1}\boldsymbol{a})^{\mathrm{T}}S(S^{-1}\boldsymbol{a}) - \boldsymbol{a}^{\mathrm{T}}(S^{-1}\boldsymbol{a}) = \frac{1}{2}\boldsymbol{a}^{\mathrm{T}}S^{-1}\boldsymbol{a} - \boldsymbol{a}^{\mathrm{T}}S^{-1}\boldsymbol{a} = \boldsymbol{-\frac{1}{2}a^{\mathrm{T}}S^{-1}a}.$$

El gráfico de F es un bol que pasa por cero en $\boldsymbol{x = 0}$ y alcanza su mínimo en $\boldsymbol{x^*}$.

Ejemplo 4 El **determinante** $F(\boldsymbol{x}) = \boldsymbol{\det X}$ es una función de todas las n^2 variables x_{ij}. En la fórmula para $\det \boldsymbol{X}$, cada x_{ij} a lo largo de una fila es multiplicada por su "cofactor" C_{ij}. Este cofactor es un determinante de tamaño $n - 1$, que usa todas las filas de $\boldsymbol{X}$ excepto la fila i y todas las columnas excepto la columna j, multiplicado por $(-1)^{i+j}$:

$$\text{Las derivadas parciales } \frac{\partial(\det X)}{\partial x_{ij}} = C_{ij} \text{ en la matriz de cofactores de } X \text{ conforman } \boldsymbol{\nabla F}.$$

Ejemplo 5 El **logaritmo del determinante** es una función muy destacable :

$$\boldsymbol{L(X) = \log(\det X)} \text{ tiene derivadas parciales } \frac{\partial L}{\partial x_{ij}} = \frac{C_{ij}}{\det X} = \textbf{elemento } \boldsymbol{j, i} \textbf{ de } \boldsymbol{X^{-1}}.$$

La regla de la cadena para $L = \log F$ es $(\partial L/\partial F)(\partial F/\partial x_{ij}) = (1/F)(\partial F/\partial x_{ij}) = (1/\det X)\, C_{ij}$. Así este cociente de cofactores sobre el determinante da los elementos j, i de la matriz inversa X^{-1}.

Es muy práctico que X^{-1} contenga las n^2 derivadas primeras de $L = \log\det X$. Las derivadas segundas de L también se destacan. Tenemos n^2 variables x_{ij} y n^2 derivadas primeras en $\boldsymbol{\nabla L} = (X^{-1})^{\mathrm{T}}$. ¡ Esto significa n^4 derivadas segundas ! Lo más increíble es que la matriz de derivadas segundas es **definida negativa** cuando $X = S$ es simétrica definida positiva. Entonces invertimos el signo de L : **derivadas segundas definidas positivas $\Rightarrow$ función convexa**.

$-\log(\det S)$ es una función convexa de las componentes en la matriz definida positiva S.

La geometría del vector gradiente ∇f

Comenzamos con una función $f(x,y)$ de $n=2$ variables. Su gradiente es $\nabla \boldsymbol{f}=(\boldsymbol{\partial f/\partial x}, \boldsymbol{\partial f/\partial y})$. Este vector cambia de longitud a medida que movemos el punto x, y donde se calculan las derivadas :

$$\nabla \boldsymbol{f} = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right) \textbf{ Longitud} = ||\nabla \boldsymbol{f}|| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} = \text{pendiente más pronunciada de } f$$

Esa longitud $||\nabla \boldsymbol{f}||$ nos indica la inclinación es el gráfico de $z = f(x,y)$. El gráfico es normalmente una superficie curva, como una montaña o un valle en el espacio xyz . En cada punto hay una pendiente $\partial f/\partial x$ en la dirección x y una pendiente $\partial f/\partial y$ en la dirección y. **La pendiente más pronunciada es en la dirección $\nabla \boldsymbol{f} = \textbf{grad } \boldsymbol{f}$. La magnitud de esa pendiente máxima es $||\nabla \boldsymbol{f}||$.**

Ejemplo 6 El gráfico de una función lineal $f(x,y) = ax + by$ es el plano $z = ax + by$. El gradiente es el vector $\nabla \boldsymbol{f} = \begin{bmatrix} a \\ b \end{bmatrix}$ de derivadas parciales. La longitud de ese vector es $||\nabla \boldsymbol{f}|| = \sqrt{a^2+b^2} =$ pendiente del techo. La pendiente es máxima en la dirección de $\nabla \boldsymbol{f}$.

Esa dirección de máxima pendiente es perpendicular a la recta de nivel. La recta de nivel $z =$ constante satisface $ax+by =$ constante. Es una dirección segura para recorrer, perpendicular a $\nabla \boldsymbol{f}$. La componente de $\nabla \boldsymbol{f}$ en esa dirección plana es cero. La Figura VI.7 muestra las dos direcciones perpendiculares (nivel y de máxima pendiente) en el plano $z = x + 2y = f(x,y)$.

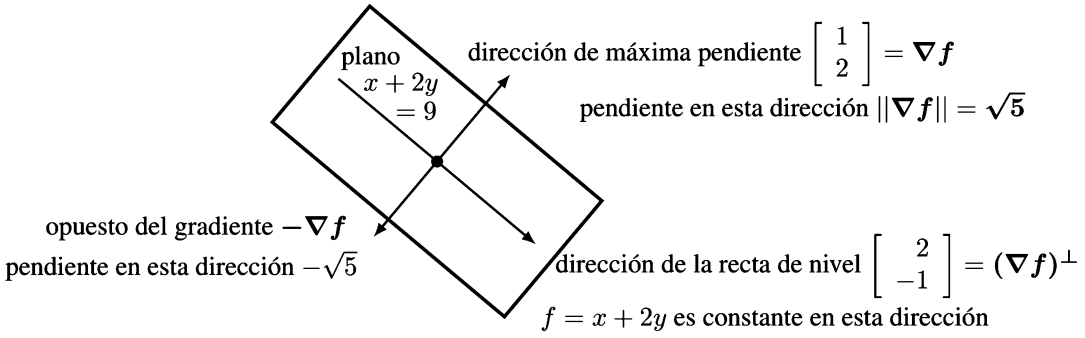
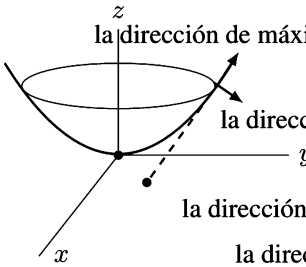


Figura VI.7: el opuesto del gradiente $-\nabla \boldsymbol{f}$ da la dirección de máximo descenso.

Para la función no lineal $f(x,y) = ax^2 + by^2$, el gradiente es $\nabla \boldsymbol{f} = \begin{bmatrix} 2ax \\ 2by \end{bmatrix}$. Esto nos dice que la dirección de máxima pendiente cambia de punto a punto. Nos encontramos con una superficie curva (un bol que se abre hacia arriba). El fondo del bol está en $x = y = 0$ donde el vector gradiente es cero. La pendiente en la dirección más empinada es $||\nabla \boldsymbol{f}||$. En el mínimo, $\nabla \boldsymbol{f} = (2ax, 2by) = (0,0)$ y la *pendiente = cero*.

La curva de nivel satisface $z = ax^2 + by^2 =$ altura constante. Ese plano $z =$ constante corta el bol en una curva de nivel. En este ejemplo, la curva de nivel $ax^2 + by^2 = c$ es una elipse. La dirección de la elipse (dirección de nivel) es perpendicular al vector gradiente (dirección de máxima pendiente). Pero hay una dificultad seria en los descensos más pronunciados :

¡ La dirección más inclinada cambia a medida que descendemos ! El gradiente no apunta hacia el mínimo.



la dirección de máxima pendiente ∇f hacia arriba y hacia abajo del bol $ax^2 + by^2 = z$

la dirección de nivel $(\nabla f)^\perp$ a lo largo de la elipse $ax^2 + by^2 =$ constante

la dirección más empinada es perpendicular a la dirección de nivel pero

la dirección más empinada no apunta hacia el punto mínimo

Figura VI.8: **el descenso más pronunciado se desplaza a lo largo del bol en la dirección del gradiente** $\begin{bmatrix} -2ax \\ -2by \end{bmatrix}$.

Reitero, en este punto (x_0, y_0) la dirección del gradiente para $f = ax^2 + by^2$ es a lo largo de $\nabla f = (2ax_0, 2by_0)$. La recta más inclinada en (x_0, y_0) es $2ax_0(y - y_0) = 2by_0(x - x_0)$. Pero así el punto más bajo $(x, y) = (0, 0)$ no yace sobre la línea. **No encontraremos ese punto mínimo en un paso del "descenso por gradiente". La dirección más empinada no lleva al fondo del bol**, excepto cuando $b = a$ y el bol es circular.

El agua cambia de dirección a medida que baja por la montaña. Tarde o temprano también debemos cambiar de dirección. En la práctica seguimos avanzando en dirección del gradiente y nos detenemos cuando la función de costo f no disminuye rápidamente. En ese punto termina el paso 1 y recalculamos el gradiente ∇f. Esto da una nueva dirección de descenso para el Paso 2.

Un ejemplo importante con zigzag

El ejemplo $f(x, y) = \frac{1}{2}(x^2 + by^2)$ es muy útil cuando $0 < b \leq 1$. Su gradiente ∇f tiene dos componentes $\partial f/\partial x = x$ y $\partial f/\partial y = by$. El valor mínimo de f es cero. Ese mínimo se alcanza en el punto $(x^*, y^*) = (0, 0)$. Lo mejor de todo, el descenso con la pendiente más pronunciada con la búsqueda lineal exacta produce una fórmula sencilla para cada punto (x_k, y_k) en el lento progreso descendiendo del bol hacia $(0, 0)$. A partir de $(x_0, y_0) = (b, 1)$ encontramos estos puntos :

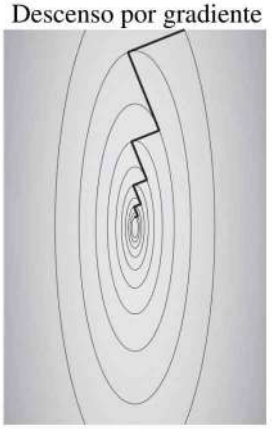
$$\boxed{x_k = b\left(\frac{b-1}{b+1}\right)^k \quad y_k = \left(\frac{1-b}{1+b}\right)^k \quad f(x_k, y_k) = \left(\frac{1-b}{1+b}\right)^{2k} f(x_0, y_0)} \tag{4}$$

Si $b = 1$, inmediatamente logramos el éxito en un paso. El punto (x_1, y_1) es $(0, 0)$. El bol es perfectamente circular con $f = \frac{1}{2}(x^2 + y^2)$. La dirección de menos el gradiente pasa exactamente por $(0, 0)$. Así el primer paso del descenso por gradiente encuentra el punto minimizante correcto cuando $f = 0$.

El verdadero propósito de este ejemplo puede verse cuando b es pequeño. El cociente crucial en la ecuación (4) es $r = (b - 1)/(b + 1)$. Para $b = \frac{1}{10}$ este cociente es $r = -9/11$. Para $b = \frac{1}{100}$ el cociente es $-99/101$. El cociente se aproxima a -1 y el progreso hacia $(0, 0)$ prácticamente se detiene cuando b es muy pequeño.

La Figura VI.9 muestra el frustante patrón en zigzag de los pasos hacia $(0, 0)$. Cada paso es corto y el progreso es muy lento. Este es un caso donde el tamaño del paso s_k en $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - s_k \nabla \boldsymbol{f}(\boldsymbol{x}_k)$ fue elegido con exactitud para minimizar f (una búsqueda lineal exacta). Pero la dirección de $-\nabla \boldsymbol{f}$, incluso si es la más pronunciada, apunta lejos de la respuesta final $(x^*, y^*) = (0, 0)$.

El bol se ha vuelto un valle angosto cuando b es pequeño, inútilmente cruzamos el valle en lugar de descender por el valle hasta el fondo.



El primer paso del descenso por gradiente comienza perpendicular al conjunto de nivel. A medida que cruza y atraviesa los conjuntos de nivel inferiores, la función $f(x, y)$ decrece. **Eventualmente su camino es tangencial al conjunto de nivel $\boldsymbol{L}$.** El descenso se ha detenido. Continuar avanzando aumentará f. Termina el primer paso. El siguiente paso es perpendicular a L. Así el camino zigzagueante dio un giro de 90 °.

Figura VI.9: la convergencia lenta en un camino zigzagueante al mínimo de $f = x^2 + by^2$.

Para b cerca de 1, este descenso por gradiente es más rápido. Una convergencia de primer orden significa que la distancia hasta $(x^*, y^*) = (0, 0)$ se ve reducida por el factor constante $(1 - b)/(1 + b)$ en cada paso. El siguiente análisis mostrará que la convergencia lineal se extiende a todas las funciones estrictamente convexas f, primero cuando la búsqueda de cada recta es exacta, y luego (más realista) cuando la búsqueda en cada paso está cerca de ser exacta.

El aprendizaje automático a menudo usa el **descenso por gradiente estocástico**. La siguiente sección describirá esta variación (muy útil cuando n es grande y $\boldsymbol{x}$ tiene muchas componentes). Recordemos que el **método de Newton** usa las derivadas segundas para producir una convergencia cuadrática, el factor de reducción $(1 - b)/(1 + b)$ decrece a cero y el error se eleva al cuadrado en cada paso. (Nuestro problema modelo de minimizar una cuadrática $\frac{1}{2}\boldsymbol{x}^{\mathrm{T}} S \boldsymbol{x}$ se resuelve en un paso.) Este es el estándar de oro que los algoritmos de aproximación a menudo no logran en la práctica.

Análisis de convergencia para el descenso más pronunciado

En esta página seguimos la presentación de Boyd y Vandenberghe del libro *Convex Optimization* (publicado por Cambridge University Press). De esta elección específica de $f(x,y) = \frac{1}{2}(x^2 + by^2)$, generalizamos a una $f(\boldsymbol{x})$ cualquiera en n dimensiones que sea fuertemente convexa. La convexidad se prueba verificando que la matriz simétrica $H = \boldsymbol{\nabla^2 f}$ de derivadas segundas (la matriz hessiana) es definida positiva. En una dimensión este es el número d^2f/dx^2 :

fuertemente convexa $\boldsymbol{m > 0}$ $\quad \boldsymbol{H_{ij} = \dfrac{\partial^2 f}{\partial x_i \partial x_j}}$ tiene autovalores entre $\boldsymbol{m \leq \lambda \leq M}$ para todo $\boldsymbol{x}$

La cuadrática $f = \frac{1}{2}\left(x^2 + by^2\right)$ tiene derivadas segundas 1 y b. La derivada mixta $\partial^2 f/\partial x \partial y$ es cero. Así la matriz es diagonal y sus dos autovalores son $m = b$ y $M = 1$. Ahora veremos que **el cociente m/M controla la velocidad del descenso más pronunciado**.

El paso del descenso por gradiente es $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - s\boldsymbol{\nabla f}_k$. Estimamos f con su serie de Taylor :

$$f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}_k) + \boldsymbol{\nabla f}^{\mathrm{T}}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) + \frac{M}{2}||\boldsymbol{x}_{k+1} - \boldsymbol{x}_k||^2 \tag{5}$$

$$= f(\boldsymbol{x}_k) - s||\boldsymbol{\nabla f}||^2 + \frac{Ms^2}{2}||\boldsymbol{\nabla f}||^2 \tag{6}$$

El mejor s minimiza el lado izquierdo (búsqueda lineal exacta). El mínimo del lado derecho está en $s = 1/M$. Reemplazando ese número por s, el siguiente punto $\boldsymbol{x}_{k+1}$ satisface

$$f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}_k) - \frac{1}{2M}||\boldsymbol{\nabla f}(\boldsymbol{x}_k)||^2. \tag{7}$$

Un argumento paralelo usa m en lugar de M para invertir el signo en (5).

$$f(\boldsymbol{x}^*) \geq f(\boldsymbol{x}_k) - \frac{1}{2m}||\boldsymbol{\nabla f}(\boldsymbol{x}_k)||^2. \tag{8}$$

Multiplicamos (7) por M y (8) por m y restamos para eliminar $||\boldsymbol{\nabla f}(\boldsymbol{x}_k)||^2$. Reescribimos el resultado así :

Descenso estable en $\boldsymbol{f}$

$$\boxed{f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}^*) \leq \left(1 - \frac{m}{M}\right)\left(f(\boldsymbol{x}_k) - f(\boldsymbol{x}^*)\right)} \tag{9}$$

Esto establece que cada paso reduce la altura con respecto al fondo del valle por un factor de al menos $c = 1 - \frac{m}{M}$. Esto es una **convergencia lineal : muy lenta cuando $b = m/M$ es pequeño**.

Nuestro ejemplo zigzagueante es con $m = b$ y $M = 1$. La cota (9) garantiza que la altura $f(\boldsymbol{x}_k)$ por encima de $f(\boldsymbol{x}^*) = 0$ se reduce por un factor de al menos $1 - b$. Esta fórmula exacta, en ese problema completamente calculable, produce el factor de reducción $(1-b)^2/(1+b)^2$. Cuando b es pequeño esto es aproximadamente $1 - 4b$. Así, en efecto, la mejora fue tan solo 4 veces más que la estimación aproximada $1 - b$ en (9). **Esto nos da mucha fe de que (9) es realista**.

Búsqueda lineal inexacta y retroceso (*backtracking*)

Ese cociente m/M aparece a lo largo de la teoría de aproximación y del álgebra lineal numérica. Esta es la esencia del análisis matemático: encontrar números como m/M que controlan la velocidad de descenso al valor mínimo $f(\boldsymbol{x}^*)$.

Hasta ahora todas las búsquedas lineales fueron exactas: $\boldsymbol{x}_{k+1}$ minimizaba exactamente $f(\boldsymbol{x})$ a lo largo de la recta $\boldsymbol{x} = \boldsymbol{x}_k - s\nabla \boldsymbol{f}_k$. La elección de s es una minimización en una variable. La recta pasa por $\boldsymbol{x}_k$ en la dirección del descenso más pronunciado. Pero no podemos esperar una fórmula exacta para minimizar una función general $f(\boldsymbol{x})$, incluso a lo largo de una recta. Entonces necesitamos una forma rápida y sensata de encontrar un mínimo aproximado (y el análisis necesita una cota de este error adicional).

Una manera sensata es la **vuelta atrás** o **retroceso** (*backtracking*). Comienza con el paso completo ($s = 1$) a $\boldsymbol{X} = \boldsymbol{x}_k - \nabla \boldsymbol{f}_k$.

Prueba Si $f(\boldsymbol{X}) \le f(\boldsymbol{x}_k) - \dfrac{s}{3}||\nabla \boldsymbol{f}_k||^2$, con $s = 1$, detener y aceptar $\boldsymbol{X}$ como $\boldsymbol{x}_{k+1}$.

De lo contrario retrocedemos: reducimos s a $\frac{1}{2}$ y realizamos la prueba con $\boldsymbol{X} = \boldsymbol{x}_k - \frac{1}{2}\nabla \boldsymbol{f}_k$.

Si la prueba falla de nuevo, probamos un paso de tamaño $s = \frac{1}{4}$. Como $\nabla \boldsymbol{f}$ es una dirección descendente, eventualmente se tiene éxito con la prueba. Los factores $\frac{1}{3}$ y $\frac{1}{2}$ podrían reemplazarse por los números $\alpha < \frac{1}{2}$ y $\beta < 1$, respectivamente.

Boyd y Vandenberghe muestran que el análisis de convergencia para la búsqueda lineal exacta se extiende también a esta búsqueda con retroceso. Por supuesto el factor de reducción garantizado $1 - (m/M)$ para cada paso hacia el mínimo ahora ya no es tan grande. Pero el nuevo factor $1 - \min(2m\alpha, 2m\alpha\beta/M)$ sigue estando por debajo de 1. El descenso más pronunciado con la búsqueda con retroceso todavía tiene convergencia lineal: con un factor constante (o mejor) en cada paso.

Momento y la trayectoria de una bola pesada

La lenta trayectoria zigzag del descenso más pronunciado es un problema real. Debemos mejorarlo. Nuestro ejemplo modelo $f = \frac{1}{2}(x^2 + by^2)$ tiene solo dos variables x, y y su matriz de derivadas segundas H es diagonal con componentes constantes $f_{xx} = 1$ y $f_{yy} = b$. Pero muestra el problema del zigzag muy claramente cuando $\boldsymbol{b = \lambda_{\min}/\lambda_{\max} = m/M}$ **es pequeño**.

Idea clave: el zigzag no ocurriría con una bola pesada que rueda colina abajo. Su momento la lleva por el valle angosto, golpeando los costados pero principalmente avanzando. Entonces **sumamos momento agregando un coeficiente $\boldsymbol{\beta}$ al gradiente** (una idea importante de Polyak). Esto nos da una de las ideas más convenientes y poderosas para el aprendizaje profundo.

La dirección $\boldsymbol{z}_k$ del nuevo paso es parecida a la dirección anterior $\boldsymbol{z}_{k-1}$.

Descenso con momento
$$\boxed{\boldsymbol{x_{k+1} = x_k - sz_k} \text{ con } \boldsymbol{z_k = \nabla f(x_k) + \beta z_{k-1}}} \tag{10}$$

Ahora tenemos que elegir dos coeficientes: el tamaño del paso $\boldsymbol{s}$ y también $\boldsymbol{\beta}$. Más importante, **el paso a $\boldsymbol{x}_{k+1}$ en la ecuación (10) usa $\boldsymbol{z}_{k-1}$**. El momento convirtió un método de un paso (descenso por gradiente) a un método de dos pasos. Para regresar a un paso, tenemos que reescribir la ecuación (10) como **dos ecuaciones acopladas** (una ecuación vectorial) para el estado en el tiempo $k + 1$:

Descenso con momento
$$\boxed{\begin{array}{rl} \boldsymbol{x}_{k+1} & = \boldsymbol{x}_k - \boldsymbol{s z}_k \\ \boldsymbol{z}_{k+1} - \nabla \boldsymbol{f}(\boldsymbol{x}_{k+1}) & = \quad \boldsymbol{\beta z}_k \end{array}} \tag{11}$$

Con esas dos ecuaciones hemos recuperado un método de un paso. Esto es exactamente como reducir una ecuación diferencial de segundo orden a un sistema de dos ecuaciones de primer orden. El segundo orden se reduce a primer orden cuando dy/dt se transforma en una segunda incógnita.

Ecuación de 2do orden $\dfrac{d^2y}{dt^2} + b\dfrac{dy}{dt} + ky = 0$ se transforma en $\dfrac{d}{dt}\begin{bmatrix} y \\ dy/dt \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k & -b \end{bmatrix}\begin{bmatrix} y \\ dy/dt \end{bmatrix}$.
Sistema de 1er orden

Es interesante que este b reduce la velocidad mientras que β le agrega momento para intensificarla.

El modelo cuadrático

Cuando $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}$ es cuadrático, su gradiente $\boldsymbol{\nabla f} = S\boldsymbol{x}$ es lineal. Este es el problema modelo que nos ayuda a comprender: S es simétrica definida positiva, y $\boldsymbol{\nabla f}(\boldsymbol{x}_{k+1})$ se transforma en $S\boldsymbol{x}_{k+1}$ en la ecuación (11). Nuestro supermodelo de 2 por 2 está incluido cuando la matriz S es diagonal con elementos 1 y b. Para una matriz S más grande, veremos que sus autovalores más grande y más pequeño determinan las mejores elecciones de β y del tamaño del paso s, así el caso de 2 por 2 contiene la esencia de todo el problema.

Para seguir los pasos del descenso acelerado, rastreamos cada autovector de S. Si $S\boldsymbol{q} = \lambda\boldsymbol{q}$, $\boldsymbol{x}_k = c_k\,\boldsymbol{q}$, $\boldsymbol{z}_k = d_k\,\boldsymbol{q}$ y $\boldsymbol{\nabla f}_k = S\boldsymbol{x}_k = \lambda c_k\,\boldsymbol{q}$, entonces nuestra ecuación (11) conecta los números c_k y d_k en el paso k a c_{k+1} y d_{k+1} en el paso $k+1$.

Siguiendo el autovector $\boldsymbol{q}$
$$\begin{aligned} c_{k+1} &= c_k - s\,d_k \\ -\lambda\,c_{k+1} + d_{k+1} &= \beta\,d_k \end{aligned} \qquad \begin{bmatrix} \mathbf{1} & 0 \\ \boldsymbol{-\lambda} & \mathbf{1} \end{bmatrix}\begin{bmatrix} c_{k+1} \\ d_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \boldsymbol{-s} \\ 0 & \boldsymbol{\beta} \end{bmatrix}\begin{bmatrix} c_k \\ d_k \end{bmatrix} \tag{12}$$

Finalmente invertimos la primera matriz ($-\lambda$ se convierte en $+\lambda$) para ver con claridad cada paso de descenso:

Un paso de descenso multiplica por $\boldsymbol{R}$
$$\boxed{\begin{bmatrix} c_{k+1} \\ d_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \lambda & 1 \end{bmatrix}\begin{bmatrix} 1 & -s \\ 0 & \beta \end{bmatrix}\begin{bmatrix} c_k \\ d_k \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \boldsymbol{-s} \\ \boldsymbol{\lambda} & \boldsymbol{\beta - \lambda s} \end{bmatrix}\begin{bmatrix} c_k \\ d_k \end{bmatrix} = \boldsymbol{R}\begin{bmatrix} c_k \\ d_k \end{bmatrix}} \tag{13}$$

Luego de k pasos, el vector inicial se multiplica por R^k. Para una convergencia rápida a cero (que es el mínimo de $f = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}$) queremos que ambos autovalores e_1 y e_2 de R sean lo más pequeños posibles. Claramente esos autovalores de R dependen del autovalor λ de S. Ese autovalor λ puede encontrarse en cualquier lugar entre $\lambda_{\text{mín}}(S)$ y $\lambda_{\text{máx}}(S)$. Nuestro problema es:

Elegir s y β para minimizar $\max\left[|\boldsymbol{e_1(\lambda)}|, |\boldsymbol{e_2(\lambda)}|\right]$ para $\lambda_{\text{mín}}(S) \le \lambda \le \lambda_{\text{máx}}(S)$. (14)

Parece un milagro que este problema tenga una solución hermosa. El s y el β óptimos son

$$\boxed{s = \left(\frac{2}{\sqrt{\lambda_{\text{máx}}} + \sqrt{\lambda_{\text{mín}}}}\right)^2 \qquad \text{y} \qquad \beta = \left(\frac{\sqrt{\lambda_{\text{máx}}} - \sqrt{\lambda_{\text{mín}}}}{\sqrt{\lambda_{\text{máx}}} + \sqrt{\lambda_{\text{mín}}}}\right)^2.} \tag{15}$$

Pensemos en el supermodelo de 2 por 2, cuando S tiene autovalores $\lambda_{\text{máx}} = 1$ y $\lambda_{\text{mín}} = b$:

$$s = \left(\frac{2}{1+\sqrt{b}}\right)^2 \qquad \text{y} \qquad \beta = \left(\frac{1-\sqrt{b}}{1+\sqrt{b}}\right)^2 \tag{16}$$

Estas elecciones de tamaño de paso y momento dan una velocidad de convergencia que se parece a la tasa de la ecuación (4) para el descenso más pronunciado ordinario (sin momento). Pero hay una diferencia crucial : **b se reemplaza por $\sqrt{b}$.**

Factor de descenso ordinario	$\left(\dfrac{1-b}{1+b}\right)^2$	**Factor de descenso acelerado**	$\left(\dfrac{1-\sqrt{b}}{1+\sqrt{b}}\right)^2$

(17)

Tan similares pero a la vez tan diferentes. La prueba real llega cuando b es muy pequeño. Así el factor de descenso ordinario es esencialmente $1-4b$, muy cercano a 1. El factor de descenso acelerado es esencialmente $1-4\sqrt{b}$, *mucho mas lejos de* 1.

Para enfatizar la mejora que trae el momento, consideremos $b = 1/100$. Así $\sqrt{b} = 1/10$ (diez veces más grande que b). los factores de convergencia en la ecuación (17) son

Descenso más pronunciado $\left(\dfrac{.99}{1.01}\right)^2 = .96$ **Descenso acelerado** $\left(\dfrac{.9}{1.1}\right)^2 = .67$

Diez pasos de descenso ordinario multiplican el error inicial por 0.67. Esto es equivalente a un paso de momento. Diez pasos con el término del momento multiplican el error por 0.018.

Observemos que $\lambda_{\text{máx}}/\lambda_{\text{mín}} = 1/b = \kappa$ es el **número de condición de S**. Esto controla todo. Para los problemas no cuadráticos estudiados a continuación, el número de condición sigue siendo la clave. Ese número κ se transforma en L/μ como se verá.

Un comentario breve *Esto no es parte del libro de texto expositivo.* Se trata de la velocidad de convergencia del descenso por gradiente. Sabemos que los métodos de un paso (calcular $\boldsymbol{x}_{k+1}$ a partir de $\boldsymbol{x}_k$) puede multiplicar el error por $1-O\left(1/\kappa\right)$. Los métodos de dos pasos que usan $\boldsymbol{x}_{k-1}$ en el término del momento pueden lograr $1-O(\sqrt{1/\kappa})$. El número de condición es $\kappa = \lambda_{\text{máx}}/\lambda_{\text{mín}}$ para el modelo cuadrático convexo $\boldsymbol{f} = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}S\boldsymbol{x}$. En nuestro ejemplo $\boldsymbol{1/\kappa = b}$.

Es natural esperar poder lograr $\boldsymbol{1 - c\kappa^{-1/n}}$ con n valores conocidos $\boldsymbol{x}_k, \boldsymbol{x}_{k-1}, \ldots, \boldsymbol{x}_{k-n+1}$. *Puede que esto sea imposible*, incluso cuando es un paralelo exacto a las fórmulas de diferencias finitas para $\boldsymbol{d\boldsymbol{x}/dt = \boldsymbol{f}(\boldsymbol{x})}$. Ahí la estabilidad requiere una cota para el tamaño del paso Δt. La estabilidad del descenso requiere una cota en el tamaño del paso s. Para ecuaciones diferenciales rígidas (*stiff*) a menudo se elige el código ODE15S de MATLAB y ODE45 es el caballo de batalla para soluciones más suaves. Esas son combinaciones de predictor-corrector, que no son importantes por ahora en optimización.

La aceleración causada por el momento es como la "**sobrerrelajación**" de la década del 1950. La tesis de David Young nos dio la misma mejora para métodos iterativos aplicados a una clase de ecuaciones lineales $A\boldsymbol{x} = \boldsymbol{b}$. En esos primeros días del análisis numérico, A se separaba en $S-T$ y las iteraciones eran $S\boldsymbol{x}_{k+1} = T\boldsymbol{x}_k + \boldsymbol{b}$. El proceso tardaba una eternidad. Ahora casi hemos olvidado la sobrerrelajación ya que fue reemplazada por métodos más rápidos (como multigrilla). ¿ El descenso más pronunciado acelerado dará lugar a ideas completamente diferentes para minimizar $f(\boldsymbol{x})$?

Esto es posible, pero difícil de imaginar.

La aceleración de Nesterov

Otra manera de introducir $\boldsymbol{x}_{k-1}$ en la fórmula para $\boldsymbol{x}_{k+1}$ se la debemos a Yurii Nesterov. En lugar de evaluar el gradiente $\boldsymbol{\nabla f}$ en $\boldsymbol{x}_k$, él desplazó ese punto de evaluación a $\boldsymbol{x}_k + \gamma_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$. Y eligiendo $\gamma = \beta$ (el coeficiente del momento) combina ambas ideas.

Descenso por gradiente	Paso $\boldsymbol{s}$	$\boldsymbol{\beta = 0}$	$\boldsymbol{\gamma = 0}$
Bola pesada	Paso $\boldsymbol{s}$	Momento $\boldsymbol{\beta}$	$\boldsymbol{\gamma = 0}$
Aceleración de Nesterov	Paso $\boldsymbol{s}$	Momento $\boldsymbol{\beta}$	desplazar $\nabla \boldsymbol{f}$ por $\gamma \Delta \boldsymbol{x}$

El descenso acelerado requiere de los tres parámetros $\boldsymbol{s}, \boldsymbol{\beta}, \boldsymbol{\gamma}$:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{\beta}\,(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) - \boldsymbol{s}\,\boldsymbol{\nabla f}\,(\boldsymbol{x}_k + \boldsymbol{\gamma}\,(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})) \tag{18}$$

Para analizar la velocidad de convergencia de Nesterov con $\gamma = \beta$, reducimos (18) a primer orden:

$$\textbf{Nesterov} \qquad \boldsymbol{x}_{k+1} = \boldsymbol{y}_k - s\boldsymbol{\nabla f}(\boldsymbol{y}_k) \;\text{ y }\; \boldsymbol{y}_{k+1} = \boldsymbol{x}_{k+1} + \beta(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k). \tag{19}$$

Sean $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}$, $\boldsymbol{\nabla f} = S\boldsymbol{x}$ y $S\boldsymbol{q} = \lambda \boldsymbol{q}$ como antes. Para rastrear este conjunto de autovectores hacemos $\boldsymbol{x}_k = c_k \boldsymbol{q}$, $\boldsymbol{y}_k = d_k \boldsymbol{q}$ y $\boldsymbol{\nabla f}(\boldsymbol{y}_k) = \lambda d_k \boldsymbol{q}$ en (19):

$$c_{k+1} = (1 - s\lambda)d_k \quad \text{y} \quad d_{k+1} = (1+\beta)c_{k+1} - \beta c_k = (1+\beta)\,(1 - s\lambda)d_k - \beta c_k$$

se transforma en

$$\begin{bmatrix} c_{k+1} \\ d_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 - s\lambda \\ -\beta & (1+\beta)(1-s\lambda) \end{bmatrix} \begin{bmatrix} c_k \\ d_k \end{bmatrix} = R \begin{bmatrix} c_k \\ d_k \end{bmatrix} \tag{20}$$

Cada paso de Nesterov es una multiplicación por $\boldsymbol{R}$. Supongamos que R tiene autovalores e_1 y e_2, dependiendo de s, β y λ. Queremos que el más grande entre $|e_1|$ y $|e_2|$ sea lo más pequeño posible para todos los λ entre $\lambda_{\text{mín}}(S)$ y $\lambda_{\text{máx}}(S)$. Estas elecciones para s y β dan e pequeños:

$$s = \frac{1}{\lambda_{\text{máx}}} \;\text{ y }\; \beta = \frac{\sqrt{\lambda_{\text{máx}}} - \sqrt{\lambda_{\text{mín}}}}{\sqrt{\lambda_{\text{máx}}} + \sqrt{\lambda_{\text{mín}}}} \;\text{ dan }\; \text{máx}(|e_1|, |e_2|) = \frac{\sqrt{\lambda_{\text{máx}}} - \sqrt{\lambda_{\text{mín}}}}{\sqrt{\lambda_{\text{máx}}}} \tag{21}$$

Cuando S es la matriz de 2 por 2 con autovalores $\lambda_{\text{máx}} = 1$ y $\lambda_{\text{mín}} = b$, ese factor de convergencia (el mayor autovalor de R) es $\boldsymbol{1 - \sqrt{b}}$.

Esto muestra la misma importantísima mejora (*de b a $\sqrt{b}$*) que la fórmula del momento (bola pesada). El análisis completo de Lessard, Recht y Packard descubrió que las elecciones de Nesterov para s y β pueden mejorarse un poco. Su, Boyd y Candès desarrollaron una relación más profunda entre una optimización de Nesterov en particular y esta ecuación:

$$\textbf{Modelo para el descenso} \qquad \frac{d^2y}{dt^2} + \frac{3}{t}\frac{dy}{dt} + \nabla f(t) = 0.$$

Funciones a minimizar : el panorama general

La función $f(x)$ puede ser estrictamente convexa, apenas convexa o no convexa. Su gradiente puede ser lineal o no lineal. He aquí una lista de tipos de funciones en orden creciente de dificultad.

1. $\boldsymbol{f(x,y) = \frac{1}{2}(x^2 + by^2)}$. Esta tiene solo 2 variables. Su gradiente $\boldsymbol{\nabla f} = (x, by)$ es lineal. Su hessiana $\boldsymbol{H}$ es una matriz diagonal de 2 por 2 con componentes 1 y b. Esos son los autovalores de H. El número de condición es $\boldsymbol{\kappa = 1/b}$ cuando $0 < b < 1$. *Estrictamente convexa.*

2. $\boldsymbol{f(x_1, \ldots, x_n) = \frac{1}{2}x^{\mathrm{T}}Sx - c^{\mathrm{T}}x}$. Aquí S es una matriz simétrica definida positiva. El gradiente $\boldsymbol{\nabla f = Sx - c}$ es lineal. La hessiana es $\boldsymbol{H = S}$. Sus autovalores son λ_1 a λ_n. El número de condición es $\boldsymbol{\kappa = \lambda_{\max}/\lambda_{\min}}$. *Estrictamente convexa.*

3. $\boldsymbol{f(x_1, \ldots, x_n)}$ **= función suave y estrictamente convexa**. Su hessiana $\boldsymbol{H(x)}$ es definida positiva en todo $\boldsymbol{x}$ (H varía con $\boldsymbol{x}$). Los autovalores de H son $\lambda_1(\boldsymbol{x})$ a $\lambda_n(\boldsymbol{x})$, siempre positivos. El número de condición es el máximo en $\boldsymbol{x}$ de $\lambda_{\text{máx}}/\lambda_{\text{mín}}$.

   Un número de condición esencialmente equivalente es el cociente $L/\lambda_{\text{mín}}(\boldsymbol{x})$:

$$\boldsymbol{L} = \textbf{“constante de Lipschitz”} \quad \text{en} \quad ||\boldsymbol{\nabla f}(\boldsymbol{x}) - \boldsymbol{\nabla f}(\boldsymbol{y})|| \;\leq\; \boldsymbol{L}\,||\boldsymbol{x} - \boldsymbol{y}||. \qquad (22)$$

   Esto permite vértices en el gradiente $\boldsymbol{\nabla f}$ y saltos en la matriz de derivadas segundas H.

4. $\boldsymbol{f(x_1, \ldots, x_n)}$ **= convexa pero no estrictamente convexa**. La hessiana puede ser solo semidefinida, con $\boldsymbol{\lambda_{\min} = 0}$. Un pequeño ejemplo es la función rampa $f = \text{ReLU}(x) = \text{máx}\,(0, x)$. El gradiente $\boldsymbol{\nabla f}$ se transforma en un “**subgradiente**” que tiene múltiples valores en un punto esquina. El subgradiente de ReLU en $x = 0$ contiene todos los valores entre 0 y 1. *Las rectas que pasan por* $(0,0)$ *con esas pendientes se mantienen por debajo de la función rampa ReLU* $(\boldsymbol{x})$.

Se permite H definida positiva pero también $\lambda_{\text{mín}} = 0$. El número de condición puede ser infinito. El ejemplo más sencillo con $\lambda_{\text{mín}} = 0$ tiene su mínimo a lo largo de toda la recta $x + y = 0$:

$$f(x,y) = (\boldsymbol{x+y})^2 = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \text{ con una } \textbf{matriz semidefinida } \boldsymbol{S}$$

Esta degeneración es muy típica del aprendizaje profundo. La cantidad de pesos que usa la red a menudo excede la cantidad de muestras de entrenamiento que determinan esos pesos. (El conjunto de datos “MNIST” puede tener $60,000$ muestras de entrenamiento y $300,000$ pesos.) Esos pesos están subdeterminados pero aun así el descenso por gradiente tiene éxito. *¿ Por qué sus pesos generalizan bien (dan buenas respuestas para datos de testeo no vistos)* ?

Cuando la convexidad estricta se pierde (Cap 4), una prueba de convergencia es aún posible. Pero el número de condición es infinito y la velocidad de convergencia puede volverse sublineal.

Observación. Hablamos de convergencia lineal cuando el error $\boldsymbol{x}_k - \boldsymbol{x}^*$ (la distancia al punto minimizante) se reduce en cada paso por aproximadamente un factor constante $C < 1$:

$$\textbf{Convergencia lineal} \qquad ||\boldsymbol{x}_{k+1} - \boldsymbol{x}^*|| \approx C\,||\boldsymbol{x}_k - \boldsymbol{x}^*||. \tag{23}$$

Esto significa que el error se reduce **exponencialmente** (como C^k o $e^{k \log C}$ con $\log C < 0$). Exponencial da la impresión de rapidez, pero puede ser muy lento cuando C es cercano a 1.

Al minimizar funciones cuadráticas **los métodos no adaptativos generalmente convergen a soluciones de norma mínima**. Esas soluciones (como $\boldsymbol{x}^+ = \boldsymbol{A}^+\boldsymbol{b}$ con la pseudoinversa A^+) tienen componente cero en el espacio fila de A.

He aquí unos buenos libros de referencia para el descenso por gradiente, que incluyen la versión estocástica que se avecina en la Sección VI.5 :

1. D. Bertsekas, *Convex Analysis and Optimization*, Athena Scientific (2003).
2. S. Boyd y L. Vandenberghe, *Convex Optimization*, Cambridge Univ. Press (2004).
3. Y. Nesterov, *Introductory Lectures on Convex Optimization*, Springer (2004).
4. J. Nocedal y S. Wright, *Numerical Optimization*, Springer (1999).

Cuatro artículos escritos por Ben Recht y colaboradores han proporcionado nuevas ideas esenciales al análisis de métodos de gradientes. Los artículos 1 y 2 estudian el descenso acelerado (esta sección). Los artículos 3 y 4 estudian el descenso estocástico y el descenso adaptativo. El vídeo 5 es excelente.

1. L. Lessard, B. Recht y A. Packard. *Analysis and design of optimization algorithms via integral quadratic constraints*, arXiv : 1408.3595v7, 28 de octubre de 2015.
2. A. C. Wilson, B. Recht y M. Jordan, *A Lyapunov analysis of momentum methods in optimization*, arXiv : 1611.02635v3, 31 de diciembre de 2016.
3. A. C. Wilson, R. Roelofs, M. Stern , N. Srebro y B. Recht, *The marginal value of adaptive gradient methods in machine learning*, arXiv :1705.08292v1, 23 de mayo de 2017.
4. C. Zhang, S. Bengio, M. Hardt, B. Recht y O. Vinyals, *Understanding deep learning requires rethinking generalization*, arXiv :1611.03530v2, 26 de febrero de 2017, International Conference on Learning Representations (2017).
5. **https://simons.berkeley.edu/talks/ben-recht-2013-09-04**

La Sección VI.5 sobre el descenso por gradiente estocástico vuelve sobre estos artículos para dar el siguiente mensaje : los métodos adaptativos pueden converger, posiblemente, a pesos indeseables en aprendizaje profundo. El descenso por gradiente a partir de $\boldsymbol{x}_0 = \boldsymbol{0}$ (y SGD) encuentra la solución de norma mínima al problema de cuadrados mínimos.

Esos métodos adaptativos (variantes de *Adam*) son populares. La fórmula para $\boldsymbol{x}_{k+1}$ que solo llegaba hasta $\boldsymbol{x}_{k-1}$ irá mucho más hacia atrás, para incluir *todos los puntos anteriores* comenzando en $\boldsymbol{x}_0$. En muchos problemas eso conduce a un entrenamiento más rápido. Como con el momento, *la memoria puede ayudar.*

Cuando hay más pesos por determinar que muestras para usar (problemas subdeterminados) podemos tener múltiples puntos mínimos para $f(\boldsymbol{x})$ y múltiples soluciones de $\boldsymbol{\nabla f} = \boldsymbol{0}$. Una pregunta crucial en VI.5 es si los métodos adaptativos mejorados encuentran buenas soluciones.

Restricciones y puntos proximales

¿Cómo lidia el descenso más pronunciado con una restricción de que $\boldsymbol{x}$ pertenece a un conjunto convexo K? Los métodos del **gradiente proyectado** y del **gradiente proximal** usan cuatro ideas fundamentales.

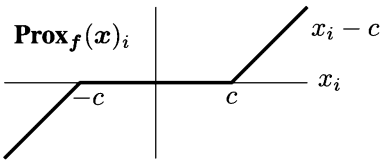
1 Proyección sobre K *La proyección $\boldsymbol{\Pi x}$ de $\boldsymbol{x}$ sobre $\boldsymbol{K}$ es el punto en $\boldsymbol{K}$ más cercano a $\boldsymbol{x}$.* Si K es curva entonces Π no es lineal. Pensemos en proyecciones sobre la bola unitaria $||\boldsymbol{x}|| \leq 1$. En este caso $\boldsymbol{\Pi x} = \text{proy}_K(\boldsymbol{x}) = \boldsymbol{x}/||\boldsymbol{x}||$ para puntos por fuera de la bola. Una propiedad clave es que **$\boldsymbol{\Pi}$ es una contracción**. Proyectar dos puntos sobre K reduce la distancia entre ellos:

$$\textbf{Proyección } \boldsymbol{\Pi} = \textbf{proy}_{\boldsymbol{K}} \qquad ||\boldsymbol{\Pi}\,\boldsymbol{x} - \boldsymbol{\Pi}\,\boldsymbol{z}|| \leq ||\boldsymbol{x} - \boldsymbol{z}|| \quad \text{para todo } \boldsymbol{x} \text{ y } \boldsymbol{z} \text{ en } \mathbf{R}^n. \tag{24}$$

2 El mapeo proximal $\textbf{Prox}_{\boldsymbol{f}}(\boldsymbol{x})$ *es el vector $\boldsymbol{z}$ que minimiza* $\frac{1}{2}||\boldsymbol{x} - \boldsymbol{z}||^2 + f(\boldsymbol{z})$. Si $f = 0$ dentro de K y $f = \infty$ fuera de K, $\text{Prox}_f(\boldsymbol{x})$ es exactamente la proyección $\boldsymbol{\Pi x}$.

Ejemplo importante Si $f(\boldsymbol{x}) = c||\boldsymbol{x}||_1$, entonces Prox_f es la **función de contracción** (*shrinkage*) en estadística.

La i-ésima componente de $\boldsymbol{x}$ lleva a
la i-ésima componente de $\text{Prox}_f(\boldsymbol{x})$
Este es el **umbral suave $\boldsymbol{S(x)}$**
$S(x_i) = \text{sign}\,(x_i) \cdot \max\,(|x_i| - c,\, 0)$



Estamos suprimiendo el ruido (*denoising*) y regularizando, como en la construcción LASSO $\boldsymbol{\ell^1}$ de la Sección III.4. El gráfico muestra cómo las componentes pequeñas son forzadas a ser cero (producir ceros es el efecto que la norma $\boldsymbol{\ell^1}$ ya logra, en comparación con $\boldsymbol{\ell^2}$).

3 El descenso por gradiente proyectado realiza un paso descendente normal (que puede ir por fuera del conjunto de restricciones K). Luego proyecta el resultado nuevamente sobre K: una idea básica.

$$\boxed{\textbf{Descenso proyectado} \qquad \boldsymbol{x}_{k+1} = \text{proy}_K(\boldsymbol{x}_k - s_k \boldsymbol{\nabla f}(\boldsymbol{x}_k))} \tag{25}$$

4 El descenso por gradiente proximal también comienza con un paso normal. Ahora se reemplaza la proyección sobre K con el **mapeo proximal** para determinar $s = s_k$: una idea sutil.

$$\boxed{\textbf{Descenso proximal} \qquad \boldsymbol{x}_{k+1} = \text{prox}_s(\boldsymbol{x}_k - s\boldsymbol{\nabla f}(\boldsymbol{x}_k))} \tag{26}$$

La función LASSO a minimizar es $f(\boldsymbol{x}) = \frac{1}{2}||\boldsymbol{b} - A\boldsymbol{x}||_2^2 + \lambda||\boldsymbol{x}||_1$. El mapeo proximal decide el umbral suave y el tamaño del paso s en $\boldsymbol{x}$ mediante

$$\begin{aligned} \text{prox}_s(\boldsymbol{x}) &= \underset{z}{\text{argmin}}\ \frac{1}{2s}||\boldsymbol{x} - \boldsymbol{z}||^2 + \lambda||\boldsymbol{x}||_1 \\ &= \underset{z}{\text{argmin}}\ \frac{1}{2}||\boldsymbol{x} - \boldsymbol{z}||^2 + \lambda s||\boldsymbol{x}||_1 = S_{s\lambda}(\boldsymbol{x}) \end{aligned} \tag{27}$$

Esto produce la función de umbral suave S en el gráfico de más arriba como la actualización $\boldsymbol{x}_{k+1}$.

$$\boxed{\textbf{Gradientes proximales para LASSO (descenso rápido)} \quad \boldsymbol{x}_{k+1} = S_{\lambda s}(\boldsymbol{x}_k + sA^{\mathrm{T}}(\boldsymbol{b} - A\boldsymbol{x}_k))}$$

Problemas de la Sección VI.4

1 Para una matriz de 1 por 1 en el Ejemplo 3, el determinante es tan solo $\det X = x_{11}$. Encuentre las derivadas primeras y segundas de $F(X) = -\log(\det X) = -\log x_{11}$ para $x_{11} > 0$. Dibuje el gráfico de $F = -\log x$ para ver que esta función F es convexa.

2 El determinante de una matriz de 2 por 2 es $\det(X) = ad - bc$. Sus derivadas primeras son $d, -c, -b, a$ en $\boldsymbol{\nabla F}$. Dividiendo por $\det X$, obtenemos los coeficienes de la matriz inversa X^{-1}. Esa división por el $\det X$ las convierte en las cuatro derivadas de $\log(\det X)$:

$$\begin{matrix}\textbf{Derivadas}\\ \textbf{de } \boldsymbol{F = \det X}\end{matrix} \quad \boldsymbol{\nabla F} = \begin{bmatrix} \boldsymbol{d} & \boldsymbol{-c} \\ \boldsymbol{-b} & \boldsymbol{a} \end{bmatrix} \quad \begin{matrix}\textbf{Inversa}\\ \textbf{de } \boldsymbol{X}\end{matrix} = \frac{1}{\boldsymbol{\det X}} \begin{bmatrix} \boldsymbol{d} & \boldsymbol{-b} \\ \boldsymbol{-c} & \boldsymbol{a} \end{bmatrix} = \frac{(\boldsymbol{\nabla F})^{\mathrm{T}}}{\boldsymbol{\det X}}$$

La simetría da $b = c$. Así $F = -\log(ad - b^2)$ es una función convexa de a, b, d. *Muestre que la matriz* 3 *por* 3 *de derivadas segundas de de esta función* F *es definida positiva.*

3 Muestre cómo las ecuaciones (7) y (8) llevan a la estimación básica (9) para la convergencia lineal del descenso más pronunciado. (Esto se extiende al método con retroceso para otras opciones de c.)

4 Muestre que este ejemlo no cuadrático tiene su mínimo en $x = 0$ e $y = +\infty$:

$$f(x,y) = \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{x^2} + \boldsymbol{e^{-y}} \qquad \boldsymbol{\nabla f} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{-e^{-y}} \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 \\ 0 & e^{-y} \end{bmatrix} \quad \boldsymbol{\kappa = \frac{1}{e^{-y}}}$$

5 Explique por qué la proyección en un conjunto convexo K es una *contracción* en la ecuación (24). ¿ Por qué nunca aumenta la distancia $||\boldsymbol{x} - \boldsymbol{y}||$ cuando $\boldsymbol{x}$ e $\boldsymbol{y}$ son proyectados sobre K ?

6 ¿ Cuál es la ecuación del descenso por gradiente $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - s_k \nabla f(\boldsymbol{x}_k)$ para el problema de cuadrados mínimos que minimiza $f(\boldsymbol{x}) = \frac{1}{2}||A\boldsymbol{x} - \boldsymbol{b}||^2$?

VI.5. Descenso por gradiente estocástico y ADAM

El descenso por gradiente es fundamental para el entrenamiento de una red neuronal compleja. Está basado en un paso de la forma $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - s_k \nabla L(\boldsymbol{x}_k)$. Ese paso debería llevarnos cuesta abajo hacia el punto $\boldsymbol{x}^*$ donde la función de pérdida $L(\boldsymbol{x})$ se minimiza en los datos de entrenamiento $\boldsymbol{v}$. Pero para redes grandes con muchas muestras en el conjunto de entrenamiento, este algoritmo (así como está) no sirve.

Es importante reconocer dos problemas diferentes con el descenso más pronunciado clásico:

1. Calcular ∇L en cada paso del descenso: son muy costosas las derivadas de la pérdida total L con respecto a los pesos $\boldsymbol{x}$ en la red neuronal. Esa pérdida total suma las pérdidas individuales $\ell(\boldsymbol{x}, \boldsymbol{v}_i)$ *para cada muestra* $\boldsymbol{v}_i$ *en el conjunto de entrenamiento*, potencialmente millones de pérdidas separadas se calculan y suman en cada cálculo de $\boldsymbol{L}$.
2. El número de pesos es aún mayor. Así $\nabla_{\boldsymbol{x}} \boldsymbol{L} = \boldsymbol{0}$ para muchas elecciones distintas $\boldsymbol{x}^*$ de los pesos. **Algunas de esas elecciones pueden dar resultados pobres en datos de prueba no vistos antes**. La función de aprendizaje $\boldsymbol{F}$ puede fallar al "generalizar". Pero el **descenso por gradiente estocástico (SGD,** ***stochastic gradient descent***) encuentra los pesos $\boldsymbol{x}^*$ que generalizan, es decir, que funcionan con vectores no vistos $\boldsymbol{v}$ de una población similar.

El descenso por gradiente estocástico usa solo un "minilote" de los datos de entrenamiento en cada paso. B muestras serán seleccionadas al azar. Reemplazar el lote entero de los datos de entrenamiento con un minilote cambia $L(\boldsymbol{x}) = \ell(\boldsymbol{x}, \boldsymbol{v_i})$ a una suma de solo B pérdidas. Esto resuelve ambas dificultades de inmediato. El éxito del aprendizaje profundo depende de estos dos hechos:

1. Calcular $\nabla \boldsymbol{\ell_i}$ con retropropagación en B muestras es mucho más rápido. A menudo $B = 1$.
2. El algoritmo estocástico produce pesos $\boldsymbol{x}^*$ que funcionan con datos no vistos.

El primer punto es claro. El cálculo en cada paso se reduce significativamente. El segundo punto es un milagro. Una buena generalización para datos nuevos es un regalo, los investigadores trabajan duro para intentar explicarlo.

Podemos describir el panorama general de la optimización de pesos en una gran red neuronal. Los pesos se determinan con los datos de entrenamiento. Esos datos a menudo consisten de miles de muestras. Conocemos las "*características*" o "*predictores*" de cada muestra, tal vez su altura y peso, o su forma y color, o la cantidad de sustantivos, verbos y comas (para una muestra de texto). Esas características están en un vector $\boldsymbol{v}$ para cada muestra. Usaremos un minilote de muestras.

Y para cada muestra del conjunto de entrenamiento, sabemos si es "un gato o un perro" o si el texto es "poesía o prosa". Buscamos una **función de aprendizaje** $\boldsymbol{F}$ que asigne pesos buenos. Así para $\boldsymbol{v}$ en una población similar, F da la clasificación correcta "gato" o "poesía".

Usamos esta función F para **datos de prueba** no identificados. Los predictores de estos datos de prueba son los nuevos elementos $\boldsymbol{v}$. El resultado de F será la clasificación correcta (?), siempre y cuando la función aprenda de los datos de entrenamiento de una manera que generaliza.

He aquí una reflexión destacable obtenida de la experiencia. *No queremos que ajuste a la perfección a los datos de entrenamiento*. A menudo a esto se lo consideraría **sobreajustar**. La función F se vuelve supersensible. Memorizó todo pero no aprendió nada. **La generalización con SGD** es la habilidad de clasificar correctamente datos no vistos $\boldsymbol{v}$, basados en los pesos $\boldsymbol{x}$ que se aprendieron usando los datos de entrenamiento.

Comparo sobreajustar con elegir un polinomio de grado 60 *que se ajusta con exactitud a* 61 *puntos dados*. Sus 61 coeficientes a_0 a a_{60} aprenderán perfectamente los datos. Pero ese polinomio de alto grado oscilará salvajemente entre los datos. Para datos de prueba en un punto cercano, el polinomio con ajuste perfecto da una *respuesta totalmente errónea*.

Entonces una estrategia fundamental para el entrenamiento de una red neuronal (que significa encontrar una función que aprende de los datos de entrenamiento y generaliza bien los datos de prueba) es una **interrupción temprana**. El aprendizaje automático necesita saber cuándo detenerse. Posiblemente esto también podría aplicarse al aprendizaje humano.

La función de pérdida y la función de aprendizaje

Ahora establecemos el problema de optimización que la red neuronal resolverá. Necesitamos definir la "pérdida" $\boldsymbol{L}(\boldsymbol{x})$ que nuestra función minimizará (aproximadamente). Esta es la suma de los errores en la clasificación de cada uno de los vectores $\boldsymbol{v}$ de los datos de entrenamiento. Y necesitamos describir la forma de la función de aprendizaje $\boldsymbol{F}$ que clasifica cada vector de datos $\boldsymbol{v}$.

En los comienzos del aprendizaje automático, la función F era *lineal*, una limitación seria. Ahora F es sin duda no lineal. La sola inclusión de una función no lineal particular en cada neurona en cada capa ha logrado una diferencia impresionante. Resulta que con miles de muestras, se puede entrenar la función F correctamente.

La potencia de procesamiento de la computadora es la que logra operaciones rápidas con los datos. En particular, dependemos de la velocidad de las GPU (*Graphical Processing Units*, unidades de procesamiento de gráficos, que originalmente se desarrollaron para los videojuegos). Estas hacen que el aprendizaje profundo sea posible.

Primero elegimos una función de pérdida a minimizar. Luego describimos el descenso por gradiente estocástico. El gradiente está determinado por la arquitectura de la red, los pasos hacia adelante o *"feedforward"* cuyos pesos optimizaremos. Nuestro objetivo para esta sección es encontrar *algoritmos de optimización que se apliquen a problemas muy grandes*. Luego el Capítulo VII describirá cómo la arquitectura y los algoritmos lograron que las funciones de aprendizaje tengan éxito.

He aquí tres funciones de pérdida, la de entropía cruzada es una de las favoritas para las redes neuronales. La Sección VII.4 describe las ventajas de la pérdida de entropía cruzada por sobre la pérdida cuadrática (nos referimos a los cuadrados mínimos).

1 **Pérdida cuadrática** $L(\boldsymbol{x}) = \dfrac{1}{N}\displaystyle\sum_{1}^{N} ||F(\boldsymbol{x}, \boldsymbol{v}_i) - \text{true}||^2$: suma sobre las muestras de entrenamiento $\boldsymbol{v}_i$

2 **Pérdida bisagra** $L(\boldsymbol{x}) = \dfrac{1}{N}\displaystyle\sum_{1}^{N} \textbf{max}\,(0, 1 - t\,F(\boldsymbol{x}))$ para la **clasificación** $t = 1$ o -1

3 **Pérdida de entropía cruzada** $L(\boldsymbol{x}) = -\dfrac{1}{N}\displaystyle\sum_{1}^{N} [y_i \log \widehat{y_i} + (1-y_i) \log (1-\widehat{y_i})]$ para $y_i = 0$ o 1

La pérdida de entropía cruzada o "pérdida logística " es la preferida para la *regresión logística* (con solo dos opciones). La etiqueta verdadera $y_i = 0$ o 1 podría ser -1 o 1 ($\widehat{y_i}$ es una etiqueta calculada).

Para un minilote de tamaño B, reemplazamos N por B. Y elegimos las B muestras al azar.

Esta sección mejoró muchísimo gracias a la clase en 18.065 de Suvrit Sra del 20 de abril 2018.

Descenso estocástico usando una muestra en cada paso

Para simplificar, supongamos de cada minilote contiene solo una muestra $\boldsymbol{v}_k$ (así $B = 1$). Esa muestra se elige al azar. La teoría del descenso estocástico usualmente supone que la muestra se repone luego de su uso. En principio, la muestra puede usarse nuevamente en el paso $k+1$. Pero la reposición es costosa comparada con comenzar con un ordenamiento aleatorio de las muestras. En la práctica, a menudo omitimos la reposición y utilizamos las muestras en un orden aleatorio.

Cada pase por los datos de entrenamiento es **una época (*epoch*)** del algoritmo descendente. El descenso por gradiente ordinario calcula un época por paso (modo lote). El descenso por gradiente estocástico necesita muchos pasos (para los minilotes). En Internet se sugiere elegir $B \leq 32$.

El descenso estocástico comenzó con un artículo seminal de Herbert Robbins y Sutton Monro en *Annals of Mathematical Statistics* **22** (1951) 400-407 : A Stochastic Approximation Method. Su objetivo era un método rápido que converja a $\boldsymbol{x}^*$ en probabilidad :

$$\text{Mostrar que} \qquad \text{Prob}\left(||\boldsymbol{x}_k - \boldsymbol{x}^*|| > \epsilon\right) \text{ se aproxima cero a medida que } k \to \infty.$$

El descenso estocástico es más sensible a los tamaños s_k de paso que el descenso por gradiente usual. Si elegimos aleatoriamente una muestra $\boldsymbol{v}_i$ en el paso k, entonces el k-ésimo paso descendente resulta familiar :

$$\boxed{\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - s_k \nabla_{\boldsymbol{x}} \boldsymbol{\ell}(\boldsymbol{x}_k, \boldsymbol{v}_i)} \qquad \boxed{\nabla_{\boldsymbol{x}} \boldsymbol{\ell} = \text{derivada del término de pérdida de la muestra } \boldsymbol{v}_i}$$

El trabajo en cada paso es mucho menor (B elementos en lugar de todos los elementos del conjunto de entrenamiento). Pero no necesariamente se converge más lentamente. Una característica típica del descenso por gradiente estocástico es la **"semiconvergencia"** : una convergencia rápida al inicio.

Los primeros pasos del SGD a menudo convergen más rápido a la solución $\boldsymbol{x}^*$ que el GD.

Esto resulta muy conveniente para el aprendizaje profundo. La Sección VI.4 muestra convergencia en zigzag si se usan todos los datos. Esto se mejoró agregando momento del paso anterior (lo cual también podemos hacer en el SGD). Otra mejora a menudo se obtiene usando **métodos adaptativos** como alguna variación de ADAM. Los métodos adaptativos miran más allá del momento, ya que ahora todas las direcciones de descenso previas son almacenadas y utilizadas. Estos se desarrollan más adelante en esta sección.

Aquí pausamos para observar la semiconvergencia : un inicio rápido utilizando descenso por gradiente estocástico. Inmediatamente admitimos que las posteriores iteraciones del SGD a menudo son erráticas. **La convergencia al inicio se transforma en oscilaciones grandes cerca de la solución**. La Figura VI.10 lo muestra. Una respuesta es detenerse temprano. Y así evitamos sobreajustar los datos.

En el siguiente ejemplo, la solución $\boldsymbol{x}^*$ está en un intervalo I específico. Si la aproximación actual $\boldsymbol{x}_k$ está afuera de I, la siguiente aproximación $\boldsymbol{x}_{k+1}$ está más cerca de I (o dentro de I). Esto da semiconvergencia: un buen inicio. *Pero eventualmente los $\boldsymbol{x}_k$ rebotan dentro de I.*

Convergencia rápida al inicio : cuadrados mínimos con $n = 1$

Aprendimos de Suvrit Sra que el ejemplo más sencillo es el mejor. El vector $\boldsymbol{x}$ tiene solo una componente x. La i-ésima pérdida es $\ell_i = \frac{1}{2}(a_i x - b_i)^2$ con $a_i > 0$. El gradiente de ℓ_i es su derivada $a_i(a_i x - b_i)$. Esta es cero y ℓ_i se minimiza en $\boldsymbol{x = b_i/a_i}$. La pérdida total sobre todas las muestras N es $L(x) = \frac{1}{2N} \sum (a_i x - b_i)^2$: cuadrados mínimos con N ecuaciones, 1 incógnita.

$$\text{La ecuación a resolver es } \boldsymbol{\nabla L} = \frac{1}{N}\sum_{1}^{N} a_i(a_i x - b_i) = 0. \quad \text{La solución es } \boldsymbol{x^* = \frac{\sum a_i b_i}{\sum a_i^2}} \quad (1)$$

Importante Si B/A es el mayor cociente b_i/a_i, entonces la verdadera solución $\boldsymbol{x^*}$ **está por debajo de** $\boldsymbol{B/A}$. Esto se sigue de cuatro desigualdades :

$$\text{cada } \frac{\boldsymbol{b_i}}{\boldsymbol{a_i}} \le \frac{\boldsymbol{B}}{\boldsymbol{A}} \qquad A\, a_i\, b_i \le B\, a_i^2 \qquad A\left(\sum a_i b_i\right) \le B\left(\sum a_i^2\right) \qquad \boldsymbol{x^*} = \frac{\sum a_i b_i}{\sum a_i^2} \le \frac{\boldsymbol{B}}{\boldsymbol{A}}. \quad (2)$$

Del mismo modo $\boldsymbol{x^*}$ está por encima del menor cociente $\boldsymbol{\beta/\alpha}$. Conclusión : si $\boldsymbol{x_k}$ está fuera del intervalo $\boldsymbol{I}$ de β/α a B/A, entonces el k-ésimo paso de descenso por gradiente se moverá *hacia ese intervalo* $\boldsymbol{I}$ que contiene a $\boldsymbol{x^*}$. He aquí lo que podemos esperar del descenso por gradiente estocástico :

> **Si $\boldsymbol{x_k}$ está afuera de $\boldsymbol{I}$, entonces $\boldsymbol{x_{k+1}}$ se mueve hacia el intervalo $\boldsymbol{\beta/\alpha \le x \le B/A}$.**
> **Si $\boldsymbol{x_k}$ está dentro de $\boldsymbol{I}$, entonces $\boldsymbol{x_{k+1}}$ también. Las iteraciones pueden rebotar dentro de $\boldsymbol{I}$.**

Una sucesión típica $\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ obtenida al minimizar $||A\boldsymbol{x} - \boldsymbol{b}||^2$ con el descenso por gradiente estocástico está graficada en la Figura VI.10. *Vemos el comienzo rápido y el final oscilante*. Este comportamiento nos sugiere pensar en frenar temprano o promediar (página 365) cuando comienzan las oscilaciones.

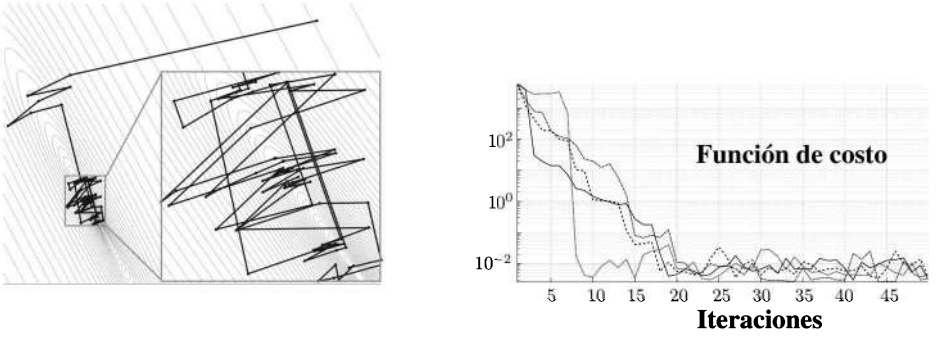


Figura VI.10: la figura izquierda muestra una trayectoria de descenso por gradiente estocástico con dos incógnitas. Las primeras iteraciones son exitosas pero las iteraciones posteriores oscilan (como se muestra en el recuadro). A la derecha, la función de costo cuadrático decrece rápido al inicio y luego fluctúa en lugar de converger. Los cuatro caminos comienzan en el mismo $\boldsymbol{x}_0$ con elecciones aleatorias de i en la ecuación (3). El número de condición de la matriz A de 40 por 2 es tan solo 8.6.

Kaczmarz aleatorizado es descenso por gradiente estocástico para $Ax = b$

Kaczmarz para $Ax = b$ con i aleatorio en el paso k $\quad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \dfrac{b_i - \boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{x}_k}{||\boldsymbol{a}_i||^2} \boldsymbol{a}_i$ (3)

Seleccionamos al alzar la fila i de A en el paso k. Ajustamos $\boldsymbol{x}_{k+1}$ para resolver la ecuación i-ésima en $A\boldsymbol{x} = \boldsymbol{b}$. (Multiplicamos la ecuación (3) por $\boldsymbol{a}_i^{\mathrm{T}}$ para verificar que $\boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{x}_{k+1} = b_i$. Esta es la ecuación i-ésima en $A\boldsymbol{x} = \boldsymbol{b}$.) Geométricamente, $\boldsymbol{x}_{k+1}$ es la proyección de $\boldsymbol{x}_k$ en uno de los hiperplanos $\boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{x} = b_i$ que se encuentran en $\boldsymbol{x}^* = A^{-1}\boldsymbol{b}$.

Este algoritmo resistió un análisis a fondo por muchos años. Las ecuaciones $\boldsymbol{a}_1^{\mathrm{T}} \boldsymbol{x} = b_1$, $\boldsymbol{a}_2^{\mathrm{T}} \boldsymbol{x} = b_2 \ldots$ se tomaban en orden cíclico con paso $s = 1$. Luego Strohmer y Vershynin mostraron la convergencia rápida de Kaczmarz aleatorio. Usaron el SGD con *muestreo por norma al cuadrado (muestreo de importancia)* como en la Sección II.4 : se elige la fila i de A con probabilidad p_i proporcional a $||\boldsymbol{a}_i^{\mathrm{T}}||^2$.

La página anterior describió las iteraciones de Kaczmarz para $A\boldsymbol{x} = \boldsymbol{b}$ cuando A era de N por 1. La sucesión $x_0, x_1, x_2, \ldots$ se movía hacia el intervalo I. La solución de cuadrados mínimos x^* estaba en ese intervalo. Para una matriz A de N por K, esperamos que los vectores $\boldsymbol{x}_i$ de K por 1 se muevan a una caja K-dimensional alrededor de $\boldsymbol{x}^*$. La Figura VI.10 muestra esto para $K = 2$.

La siguiente página presenta experimentos numéricos para el descenso por gradiente estocástico :

Gower y Richtarik desarrollaron una variante de Kaczmarz aleatorio, con no menos de seis interpretaciones equivalentes. He aquí referencias que conectan las muchas variantes del original trabajo de Kaczmarz en el Bulletin de l'Académie Polonaise de 1937.

1 T. Strohmer y R. Vershynin, A randomized Kaczmarz algorithm with exponential convergence, *Journal of Fourier Analysis and Applications* **15** (2009) 262-278.

2 A. Ma, D. Needell y A. Ramdas, Convergence properties of the randomized extended Gauss-Seidel and Kaczmarz methods, arXiv : 1503.08235$v3$ 1 de febrero de 2018.

3 D. Needell, N. Srebro y R. Ward, Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm, *Math. Progr.* **155** (2015) 549-573.

4 R. M. Gower y P. Richtarik, Randomized iterative methods for linear systems, *SIAM J. Matrix Analysis*. **36** (2015) 1660-1690; arXiv : 1506.03296v5 6 de enero de 2016.

5 L. Bottou et al, en *Advances in Neural Information Processing Systems*, NIPS **16** (2004) y NIPS **20** (2008), MIT Press.

6 S. Ma, R. Bassily y M. Belkin, *The power of interpolation* : *Understanding the effectiveness of SGD in modern over-parametrized learning*, arXiv : 1712.06559.

7 S. Reddi, S. Sra, B. Poczos y A. Smola, *Fast stochastic methods for nonsmooth nonconvex optimization*, arXiv: 1605.06900, 23 de mayo de 2016.

Kaczmarz aleatorio y proyecciones iteradas

Sea $A\boldsymbol{x}^* = \boldsymbol{b}$. Un paso típico de Kaczmarz aleatorio proyecta el error actual $\boldsymbol{x}_k - \boldsymbol{x}^*$ en el hiperplano $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{b}_i$. Aquí i se elige al azar en el paso k (a menudo con muestreo de importancia usando probabilidades proporcionales a $||a_i||^2$). Para visualizar esa matriz de proyección $\boldsymbol{a}_i\boldsymbol{a}_i^{\mathrm{T}}/\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{a}_i$, reemplazamos $b_i = \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}^*$ en el paso de actualización (3):

$$\boldsymbol{x}_{k+1} - \boldsymbol{x}^* = \boldsymbol{x}_k - \boldsymbol{x}^* + \frac{b_i - \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}_k}{||\boldsymbol{a}_i||^2}\,\boldsymbol{a}_i = (\boldsymbol{x}_k - \boldsymbol{x}^*) - \frac{\boldsymbol{a}_i\boldsymbol{a}_i^{\mathrm{T}}}{\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{a}_i}\,(\boldsymbol{x}_k - \boldsymbol{x}^*) \tag{4}$$

La proyección ortogonal nunca aumenta la magnitud. El error solo decrece. La norma del error $||\boldsymbol{x}_k - \boldsymbol{x}^*||$ decrease a paso firme, incluso si la función de costo $||A\boldsymbol{x}_k - \boldsymbol{b}||$ no lo hace. ¡ *Pero generalmente la convergencia es lenta* ! Strohmer-Vershynin estiman el error esperado:

$$\mathrm{E}\left[||\boldsymbol{x}_k - \boldsymbol{x}^*||^2\right] \le \left(1 - \frac{1}{c^2}\right)^k ||\boldsymbol{x}_0 - \boldsymbol{x}^*||^2,\ c = \text{número de condición de } A. \tag{5}$$

Esto es lento en comparación con el descenso por gradiente (allí se reemplaza c^2 por c, y luego por $\sqrt{c}$ con momento en VI.4). Pero (5) es independiente del tamaño de A: atractivo para problemas grandes.

La teoría de proyecciones alternantes fue iniciada por von Neumann (en espacios de Hilbert). Vea libros y artículos de Bauschke-Borwein, Escalante-Raydan, Diaconis, Xu,...

Nuestros experimentos convergen lentamente. La matriz A de 100 por 10 es aleatoria con $c \approx 400$. Las figuras muestran Kaczmarz aleatorio con 600.000 pasos. Medimos la convergencia con el ángulo θ_k entre $\boldsymbol{x}_k - \boldsymbol{x}^*$ y la fila $\boldsymbol{a}_i$ elegida en el paso k. La ecuación del error (4) es

$$||\boldsymbol{x}_{k+1} - \boldsymbol{x}^*||^2 = (1 - \cos^2\theta_k)\,||\boldsymbol{x}_k - \boldsymbol{x}^*||^2 \tag{6}$$

El gráfico confirma que esos números $1 - \cos^2\theta_k$ *son muy cercanos a* 1: **convergencia lenta**. Pero el segundo gráfico confirma que hay convergencia. La cota de Strohmer-Vershynin (5) es $\mathrm{E}\left[\cos^2\theta_k\right] \ge 1/c^2$. La matriz de nuestro ejemplo satisface $1/c^2 \approx 10^{-5}$ y a menudo $\cos^2\theta_k \approx 2 \cdot 10^{-5}$, confirmando así esa cota.

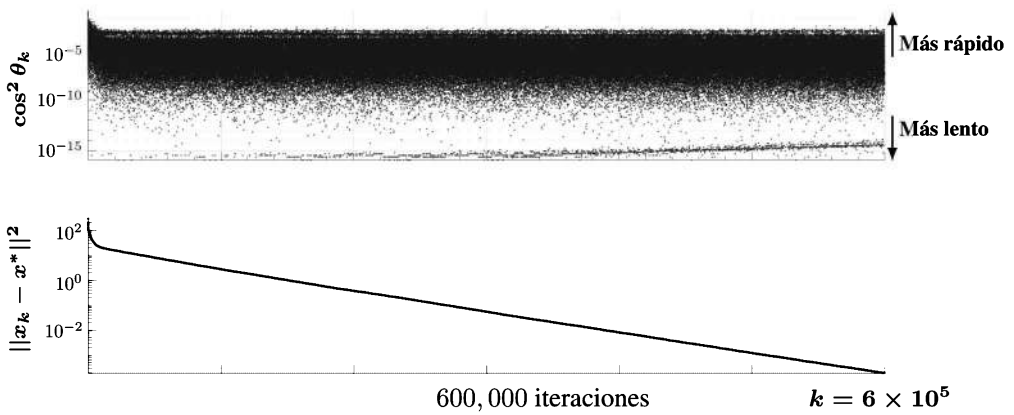


Figura VI.11: convergencia del error cuadrático para Kaczmarz aleatorio. La ecuación (6) con $1 - \cos^2\theta_k$ cercano a $1 - 10^{-5}$ produce la convergencia lenta en el gráfico inferior.

Convergencia en esperanza

Para un algoritmo estocástico como el SGD necesitamos una prueba de convergencia que tenga en cuenta la aleatoriedad, tanto en las hipótesis como en las conclusiones. Suvrit Sra nos dio esa prueba, que reproducimos a continuación. La función $f(\boldsymbol{x})$ es una suma $\frac{1}{n}\Sigma f_i(\boldsymbol{x})$ de n términos. El muestreo elige, en el paso k, uniformemente a $i(k)$ entre los números 1 a n (con reemplazo), y el tamaño del paso es $s = \text{constante}/\sqrt{T}$. Primero vienen las suposiciones de $f(\boldsymbol{x})$ y $\boldsymbol{\nabla f}(\boldsymbol{x})$, seguidas del requisito estándar (sin sesgo) de muestreo aleatorio.

1	**Suavidad Lipschitz de $\boldsymbol{\nabla f(x)}$**	$\|\|\boldsymbol{\nabla f}(\boldsymbol{x}) - \boldsymbol{\nabla f}(\boldsymbol{y})\| \leq L \, \|\|\boldsymbol{x} - \boldsymbol{y}\|\|$
2	**Gradientes acotados**	$\|\|\boldsymbol{\nabla f}_{i(k)}(\boldsymbol{x})\| \leq G$
3	**Gradiente estocásticos insesgados**	$\mathbf{E}\left[\boldsymbol{\nabla f}_{i(k)}(\boldsymbol{x}) - \boldsymbol{\nabla f}(\boldsymbol{x})\right] = 0$

De la hipótesis 1 resulta que

$$f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}_k) + (\boldsymbol{\nabla f}(\boldsymbol{x}_k), \boldsymbol{x}_{k+1} - \boldsymbol{x}_k) + \frac{1}{2} L s^2 ||\boldsymbol{\nabla f}_{i(k)}(\boldsymbol{x}_k)||^2$$
$$f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}_k) + (\boldsymbol{\nabla f}(\boldsymbol{x}_k), -s\,\boldsymbol{\nabla f}_{i(k)}(\boldsymbol{x}_k)) + \frac{1}{2} L s^2 ||\boldsymbol{\nabla f}_{i(k)}(\boldsymbol{x}_k)||^2$$

Ahora tomamos esperanzas en ambos miembros y usamos las Suposiciones 2-3 :

$$\mathbf{E}\left[f(\boldsymbol{x}_{k+1})\right] \leq \mathbf{E}\left[f(\boldsymbol{x}_k)\right] - s\,\mathbf{E}\left[\,||\boldsymbol{\nabla f}(\boldsymbol{x}_k)||^2\,\right] + \frac{1}{2} L s^2\, G^2$$
$$\Rightarrow \mathbf{E}\left[\,||\boldsymbol{\nabla f}(\boldsymbol{x}_k)||^2\,\right] \leq \frac{1}{s}\mathbf{E}\left[f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1})\right] + \frac{1}{2} L\, s\, G^2. \tag{7}$$

Elegimos el tamaño de paso $s = c/\sqrt{T}$ y luego sumamos (7) de $k = 1$ a T. La suma es telescópica :

$$\frac{1}{T}\sum_{k=1}^{T} \mathbf{E}\left[\,||\boldsymbol{\nabla f}(\boldsymbol{x}_k)||^2\,\right] \leq \frac{1}{\sqrt{T}}\left(\frac{f(\boldsymbol{x}_1) - f(\boldsymbol{x}^*)}{c} + \frac{Lc}{2}\,G^2\right) = \frac{C}{\sqrt{T}}. \tag{8}$$

Aquí $f(\boldsymbol{x}^*)$ es el mínimo global. El término más pequeño en (8) está debajo del promedio :

$$\min_{1 \leq k \leq T} \mathbf{E}[\,||\boldsymbol{\nabla f}(\boldsymbol{x_k})||^2\,] \leq \boldsymbol{C/\sqrt{T}}. \tag{9}$$

La conclusión del teorema de Sra es la convergencia en esperanza a una velocidad sublineal.

Promediando pesos dentro del SGD

La idea de **promediar los resultados** de varios pasos del descenso por gradiente estocástico parece prometedora. La velocidad de aprendizaje (tamaño del paso) puede ser constante o cíclica sobre cada grupo de resultados. Gordon Wilson y sus coautores nombraron a este método *promedio estocástico de pesos* (SWA, por su sigla en inglés). Ellos enfatizan que esto da resultados prometedores en el entrenamiento de redes neuronales profundas, con mejor generalización y casi sin costos extra. Parece natural y eficaz.

P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov y A. Gordon Wilson, *Averaging weights leads to wider optima and better generalization*, arXiv : 1803.05407.

Métodos adaptativos usando gradientes previos

Una estrategia popular para conseguir convergencias más rápidas que usando descenso por gradiente y descenso por gradiente estocástico, es utilizar métodos adaptativos. La idea es *usar gradientes de pasos previos*. Esa "memoria" orienta la elección de la dirección de búsqueda $\boldsymbol{D}$ y el tan importante tamaño de paso s. Buscamos el vector $\boldsymbol{x}^*$ que minimice una función de pérdida especificada $L(\boldsymbol{x})$. En el paso de $\boldsymbol{x}_k$ a $\boldsymbol{x}_{k+1}$, somos libres de elegir $\boldsymbol{D}_k$ y s_k :

$$\boldsymbol{D}_k = \boldsymbol{D}(\boldsymbol{\nabla L}_k, \boldsymbol{\nabla L}_{k-1}, \ldots, \boldsymbol{\nabla L}_0) \quad \text{y} \quad s_k = s(\boldsymbol{\nabla L}_k, \boldsymbol{\nabla L}_{k-1}, \ldots, \boldsymbol{\nabla L}_0). \tag{10}$$

En una iteración de SGD estándar, $\boldsymbol{D}_k$ solo depende del gradiente actual $\boldsymbol{\nabla L_k}$ (y s_k puede ser $s/\sqrt{k}$). Ese gradiente $\boldsymbol{\nabla L_k}(\boldsymbol{x}_k, B)$ es evaluado solo en un minilote aleatorio B de los datos de entrenamiento. Ahora, las redes profundas a menudo tienen la opción de usar algunos o todos los gradientes previos (que fueron calculados en minilotes aleatorios previos) :

$$\boxed{\textbf{Descenso por gradiente estocástico adaptativo} \qquad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k - s_k \boldsymbol{D}_k} \tag{11}$$

El éxito o el fracaso dependen de $\boldsymbol{D}_k$ y s_k. El primer método adaptativo (llamado ADAGRAD) elegía la dirección de búsqueda usual $\boldsymbol{D}_k = \boldsymbol{\nabla L}(\boldsymbol{x}_k)$ pero calculaba el tamaño del paso a partir de todos los gradientes previos [Duchi-Hazan-Singer] :

$$\textbf{Tamaño de paso ADAGRAD} \qquad s_k = \left(\frac{\alpha}{\sqrt{k}}\right) \left[\frac{1}{k} \operatorname{diag}\left(\sum_1^k ||\boldsymbol{\nabla L_i}||^2\right)\right]^{1/2} \tag{12}$$

$\alpha/\sqrt{k}$ es un tamaño de paso típico en la prueba de convergencia del descenso estocástico. A menudo se omite cuando, en la práctica, ralentiza la convergencia. El "factor de memoria" en (12) trae verdaderos beneficios en la velocidad de convergencia. Esos beneficios hacen que los métodos adaptativos sean un objetivo a profundizar para lograr un mayor desarrollo.

Los promedios móviles exponenciales en ADAM [Kingma-Ba] se volvieron los favoritos. A diferencia de (12), los gradientes previos recientes $\boldsymbol{\nabla L}$ tienen mayor peso que los gradientes previos más antiguos tanto en s_k como en la dirección del paso $\boldsymbol{D}_k$. Los pesos exponenciales en $\boldsymbol{D}$ y s se obtienen cuando $\delta < 1$ y $\beta < 1$:

$$\boldsymbol{D}_k = (1-\delta)\sum_{i=1}^{k} \delta^{k-i}\, \boldsymbol{\nabla} L(\boldsymbol{x}_i) \quad s_k = \left(\frac{\alpha}{\sqrt{k}}\right) \left[(1-\beta)\, \operatorname{diag} \sum_{i=1}^{k} \beta^{k-i} ||\boldsymbol{\nabla L}\,(\boldsymbol{x}_i)||^2\right]^{1/2} \tag{13}$$

Los valores típicos son $\delta = 0.9$ y $\beta = 0.999$. Los valores pequeños de δ y β efectivamente matarían la memoria móvil y se perderían las ventajas de los métodos adaptativos en la velocidad de convergencia. Esa velocidad es importante en el costo total del descenso por gradiente. La fórmula retrospectiva para la dirección $\boldsymbol{D}_k$ es como incluir momento en el método de la bola pesada de la Sección VI.4.

El cálculo de $\boldsymbol{D}_k$ y s_k en realidad será una combinación recursiva del viejo y del nuevo método :

$$\boxed{\boldsymbol{D}_k = \delta \boldsymbol{D}_{k-1} + (1-\delta)\boldsymbol{\nabla L}(\boldsymbol{x}_k) \qquad s_k^2 = \beta s_{k-1}^2 + (1-\beta)\,||\boldsymbol{\nabla L}\,(\boldsymbol{x}_k)||^2} \tag{14}$$

Para varios de los proyectos del curso este método adaptativo claramente produce una convergencia más rápida.

ADAM es muy popular en la práctica (esto se escribió en 2018). Pero varios autores han señalado sus defectos. Hardt, Recht y Singer construyeron ejemplos para mostrar que su límite $\boldsymbol{x}_\infty$ para los pesos en el aprendizaje profundo podría ser problemático: la convergencia puede fracasar o (peor) los pesos límites pueden *generalizarse pobremente* cuando se aplica sobre datos de prueba no vistos.

Las ecuaciones (13) y (14) siguen un reciente artículo de Reddi, Kale y Kumar. Esos autores demuestran la no convergencia de ADAM con ejemplos sencillos en los que *el tamaño del paso s_k aumenta con el tiempo* (un resultado indeseado). En su ejemplo, ADAM toma la dirección incorrecta dos veces y la dirección correcta en uno de cada tres pasos. El decaimiento exponencial reduce ese paso bueno y, en general, los tamaños de paso s_k no decrecen. Se necesita y se usa un β grande (cerca de 1), pero siempre hay problemas de optimización convexa en los que ADAM falla. *Aún así la idea es buena.*

Un enfoque es usar un minilote de tamaño B creciente. El artículo de NIPS de 2018 propone un nuevo algoritmo adaptativo, YOGI, que controla mejor la velocidad de aprendizaje (el tamaño del paso). En comparación con ADAM, un cambio clave es una actualización aditiva; los demás pasos no cambian. Por el momento, los experimentos muestran mejores resultados con YOGI.

Luego de la convergencia rápida a los pesos, que casi resuelven $\boldsymbol{\nabla L(x)} = \mathbf{0}$, todavía nos queda un problema crucial: **¿Por qué esos pesos generalizan bien con datos de prueba no vistos?**

Referencias

1. S. Ruder, *An overview of gradient descent optimization algorithms*, arXiv:1609.04747.
2. J. Duchi, E. Hazan y Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. of Machine Learning Research* **12** (2011) 2121–2159.
3. P. Kingma y J. Ba, ADAM: A method for stochastic optimization, ICLR, **2015**.
4. M. Hardt, B. Recht y Y. Singer, *Train faster, generalize better: Stability of stochastic gradient descent*, arXiv:1509.01240v2, 7 de febrero de 2017, Proc. ICML (2016).
5. A. Wilson, R. Roelofs, M. Stern, N. Srebro y B. Recht, *The marginal value of adaptive gradient methods in machine learning*, arXiv: 1705.08292, 23 de mayo de 2017.
6. S. Reddi, S. Kale y S. Kumar, On the convergence of ADAM and beyond, ICLR **2018**: *Proc. Intl. Conference on Learning Representations.*
7. S. Reddi, M. Zaheer, D. Sachan, S. Kale y S. Kumar, *Adaptive methods for nonconvex optimization*, NIPS (2018).
8. F. Curtis y K. Scheinberg, *Optimization Methods for Supervised Machine Learning*, arXiv: 1706.10207, 30 de junio de 2017.

Para cerrar este capítulo destaco que: el descenso por gradiente estocástico es ahora el método principal para encontrar pesos $\boldsymbol{x}$ que minimizan la pérdida $L(\boldsymbol{x})$ y resuelven $\boldsymbol{\nabla L(x^*)} = \mathbf{0}$. Esos pesos del SGD normalmente son exitosos con datos de prueba no vistos.

Generalización : ¿ Por qué es tan efectivo el aprendizaje profundo ?

Cerramos el Capítulo VI y lo conectamos al Capítulo VII con una breve discusión sobre una pregunta fundamental del aprendizaje profundo. El problema aquí es la **generalización**. Esto se refiere al comportamiento de una red neuronal con datos de prueba que no ha visto. Si construimos una función $F(\boldsymbol{x}, \boldsymbol{v})$ que sucesivamente clasifica los datos de entrenamiento conocidos $\boldsymbol{v}$, ¿ F continuará dando resultados correctos cuando $\boldsymbol{v}$ esté fuera del conjunto de entrenamiento ?

La respuesta debe encontrarse en el algoritmo de descenso por gradiente estocástico que elige los pesos. Esos pesos $\boldsymbol{x}$ minimizan una función de pérdida $L(\boldsymbol{x}, \boldsymbol{v})$ sobre los datos de entrenamiento. La pregunta es : **¿ por qué los pesos calculados en los datos de entrenamiento funcionan tan bien en los datos de prueba ?**

A menudo tenemos más parámetros libres en $\boldsymbol{x}$ que datos en $\boldsymbol{v}$. En ese caso podemos esperar que muchos conjuntos de pesos (muchos vectores $\boldsymbol{x}$) sean igualmente precisos en el conjunto de entrenamiento. Esos pesos pueden ser buenos o malos. Pueden generalizar bien o mal. Nuestro algoritmo elige un $\boldsymbol{x}$ en particular y aplica esos pesos a datos nuevos $\boldsymbol{v}_{\text{test}}$.

Un experimento inusual produjo resultados positivos inesperados. Se barajaron al azar las componentes de cada vector ingresado $\boldsymbol{v}$. Así los predictores individuales representados por $\boldsymbol{v}$ de pronto no tenían sentido. Sin embargo la red neuronal profunda aprendió esas muestras aleatorizadas. La función de aprendizaje $\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{v})$ todavía clasificaba correctamente los datos de entrenamiento. Por supuesto $\boldsymbol{F}$ no podría tener éxito con datos no vistos cuando las componentes de $\boldsymbol{v}$ están reordenadas.

Es una característica común de la optimización que las funciones suaves sean más fáciles de aproximar que las funciones irregulares. Pero aquí, para vectores ingresados completamente aleatorizados, el descenso por gradiente estocástico solo necesitó el triple de épocas (el triple de iteraciones) para aprender los datos de entrenamiento. Este etiquetado aleatorio de las muestras de entrenamiento (el experimento ahora es famoso) se describe en arXiv : 1611.03530.

El método de Kaczmarz en imágenes tomográficas

Una propiedad clave de Kaczmarz es su rápido éxito en las primeras iteraciones. Esto se conoce como *semiconvergencia* en tomografía (donde la resolución de $A\boldsymbol{x} = \boldsymbol{b}$ da lugar a una imagen de tomografía computarizada, y el método produce una solución regularizada cuando los datos tienen ruido). Una semiconvergencia rápida para datos con ruidos es una excelente propiedad para un método tan simple. Los primeros pasos aproximan el intervalo correcto de α/β a A/B (para una incógnita escalar). Pero dentro de ese intervalo, Kaczmarz salta impredeciblemente.

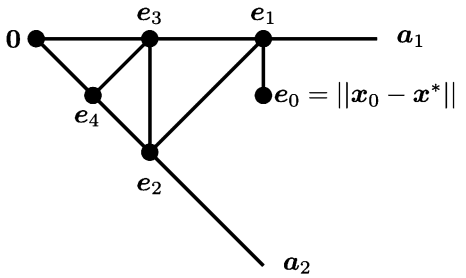
Nos adentramos aquí en el imponente tema de problemas inversos mal condicionados (vea los libros de P. C. Hansen). En este libro no podemos hacer más que abrir la puerta.

Problemas de la Sección VI.5

1 La matriz de rango uno $P = \boldsymbol{a}\boldsymbol{a}^{\mathrm{T}}/\boldsymbol{a}^{\mathrm{T}}\boldsymbol{a}$ es una proyección ortogonal sobre la recta con dirección $\boldsymbol{a}$. Verifique que $P^2 = P$ (proyección), que $P\boldsymbol{x}$ se encuentra sobre esa recta y que $\boldsymbol{x} - P\boldsymbol{x}$ es siempre perpendicular a $\boldsymbol{a}$ (*¿ por qué es* $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{a}^{\mathrm{T}}P\boldsymbol{x}$?)

2 Verifique la ecuación (4) que muestra que $\boldsymbol{x}_{k+1} - \boldsymbol{x}^*$ es exactamente $(I - P)(\boldsymbol{x}_k - \boldsymbol{x}^*)$.

3 Si A tiene solo dos filas $\boldsymbol{a}_1$ y $\boldsymbol{a}_2$, entonces Kaczmarz producirá las *proyecciones alternantes* de esta figura. Comenzando por cualquier vector de error $\boldsymbol{e}_0 = \boldsymbol{x}_0 - \boldsymbol{x}^*$, ¿ por qué $\boldsymbol{e}_k$ se aproxima a cero ? ¿ qué tan rápido lo hace ?



4 Supongamos que queremos minimizar $F(x, y) = y^2 + (y - x)^2$. El mínimo es $F = 0$ en $(x^*, y^*) = (0, 0)$. Encuentre el vector gradiente $\boldsymbol{\nabla F}$ en el punto de inicio $(x_0, y_0) = (1, 1)$. Para el descenso por gradiente usual (*no estocástico*) con paso $s = \frac{1}{2}$, ¿dónde está (x_1, y_1)?

5 Al minimizar $F(\boldsymbol{x}) = ||A\boldsymbol{x} - \boldsymbol{b}||^2$, el descenso por gradiente estocástico con tamaño de minilote $B = 1$ resolverá una ecuación $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} = b_i$ en cada paso. Explique el paso típico para un minilote de tamaño $B = 2$.

6 (Experimento) Para A y $\boldsymbol{b}$ (20 por 4 y 20 por 1) aleatorios, pruebe el descenso por gradiente estocástico con minilotes de tamaño $B = 1$ y $B = 2$. Compare las velocidades de convergencia: los cocientes $r_k = ||\boldsymbol{x}_{k+1} - \boldsymbol{x}^*|| \,/\, ||\boldsymbol{x}_k - \boldsymbol{x}^*||$.

7 (Experimento) Pruebe el promedio ponderado de la página 365 propuesto en arXiv : 1803.05407. Aplíquelo en la minimización de $||A\boldsymbol{x} - \boldsymbol{b}||^2$ con una A (20 por 10) y $\boldsymbol{b}$ (20 por 1) elegidos al azar, y un minilote $B = 1$.

¿ Los promedios en el descenso estocástico convergen más rápido que las iteraciones usuales $\boldsymbol{x}_k$?

Parte VII

Aprendizaje con datos

VII.1 La construcción de redes neuronales profundas

VII.2 Redes neuronales convolucionales

VII.3 Retropropagación y regla de la cadena

VII.4 Hiperparámetros : las decisiones cruciales

VII.5 El mundo del aprendizaje automático

Parte VII : Aprendizaje con datos

Esta parte de libro es una gran aventura para mí, y espero que también lo sea para el lector, ya que involucra toda la ciencia del pensamiento y la inteligencia. Podríamos llamarla aprendizaje automático (ML por *Machine Learning*) o inteligencia artificial (AI por *Artificial Intelligence*). La inteligencia humana la creó (pero todavía no entendemos por completo su alcance). A partir de una combinación de ideas y fallas, que intentaban al principio imitar las neuronas en el cerebro, surgió un enfoque exitoso para encontrar **patrones en datos**.

Lo más importante que hay que entender sobre aprendizaje profundo es que esos cálculos que ajustan datos, de tamaños casi sin precedentes, a menudo son muy indeterminados. Hay muchos puntos en los datos de entrenamiento, pero hay muchos más pesos por calcular en una red neuronal. El arte del aprendizaje profundo es encontrar, entre muchas soluciones posibles, una que **generalice a datos nuevos.**

Es importante destacar que el aprendizaje en una red neuronal profunda con muchos pesos lleva a un resultado exitoso : F es precisa en los datos de entrenamiento y en el conjunto de prueba consistente de datos no vistos. Esto es el buen resultado que se obtiene usando descenso por gradiente con minilote y momento y usando los hiperparámetros de la Sección VII.4 (incluyendo selección del tamaño del paso e interrupción temprana).

Este capítulo tiene un orden irregular. **Comenzamos con aprendizaje profundo**. Podemos encontrar descritos brevemente en VII.5 los modelos previos como las máquinas vectoriales de soporte y los métodos de núcleos. El orden aquí utilizado no es cronológico y el lector entenderá por qué. Las redes neuronales se han vuelto la arquitectura primaria para los problemas más interesantes (y más difíciles) del aprendizaje profundo. La arquitectura multicapa a menudo tiene éxito, ¡pero no siempre!. Este libro se ha estado preparando para el aprendizaje profundo y por lo tanto debemos darle prioridad.

Las Secciones VII.1-2 describen la función de aprendizaje $\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{v})$ para *redes completamente conectadas y redes convolucionales*. Los datos de entrenamiento tienen un conjunto de vectores $\boldsymbol{v}$ de predictores. Los pesos que permiten que $\boldsymbol{F}$ clasifique esos datos están en el vector $\boldsymbol{x}$. Para optimizar $\boldsymbol{F}$, el descenso por gradiente necesita sus derivadas $\boldsymbol{\partial F / \partial x}$. Los pesos $\boldsymbol{x}$ son las matrices $A_1, \ldots, A_L$ y los vectores de sesgo $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_L$ que transforman los datos muestrales $\boldsymbol{v} = \boldsymbol{v}_0$ al resultado $\boldsymbol{w} = \boldsymbol{v}_L$.

Las fórmulas para $\partial F / \partial A$ y $\partial F / \partial \boldsymbol{b}$ no son difíciles. Es útil ver estas fórmulas, aunque los códigos reales usan *diferenciación automática (AD) para la retropropagación* (Sección VII.3). Cada capa oculta con sus pesos optimizados aprende más sobre los datos y la población que los origina. Esto permite clasificar datos nuevos no vistos de la misma población.

Las funciones del aprendizaje profundo

Supongamos que uno de los dígitos $0, 1, \ldots, 9$ está dibujado en un cuadrado. ¿ Cómo reconoce una persona cuál dígito es? Esta es una pregunta de la neurociencia que no responderemos aquí. ¿ Cómo lo reconoce una computadora? Esta es una pregunta del aprendizaje automático. Probablemente la respuesta a ambas preguntas parta de la misma idea : *se aprende a partir de ejemplos.*

Comenzamos con M imágenes diferentes (el conjunto de entrenamiento). Una imagen puede ser un conjunto de p píxeles pequeños o un vector $\boldsymbol{v} = (v_1, \ldots, v_p)$. La componente v_i nos informa "la escala de grises" del i-ésimo píxel en la imagen : qué tan oscuro o claro es. Entonces, tenemos M imágenes, cada una con p predictores : M vectores $\boldsymbol{v}$ en un espacio p-dimensional. Para cada $\boldsymbol{v}$ en ese conjunto de entrenamiento, sabemos qué dígito representa.

De cierta forma, conocemos una función. Tenemos M elementos de entrada en $\mathbf{R}^p$ cada uno con un resultado entre 0 y 9. Pero no tenemos una "regla". No tenemos respuesta ante un nuevo elemento de entrada. El aprendizaje automático propone crear una regla que logre el éxito en la mayoría de las imágenes de entrenamiento. Pero "éxito"significa mucho más que eso : la regla debería dar el dígito correcto en un conjunto mucho más amplio de imágenes de prueba tomadas de la misma población. Este requisito se llama *generalización.*

¿ Qué forma tomará la regla ? Aquí nos encontramos con la pregunta fundamental. Nuestra primera respuesta podría ser : $F(\boldsymbol{v})$ podría ser una función lineal de $\mathbf{R}^p$ a $\mathbf{R}^{10}$ (una matriz de 10 por p). Las 10 salidas serían las probabilidades de los números 0 a 9. Tendríamos $10p$ elementos y M muestras de entrenamiento para acertar en su mayoría.

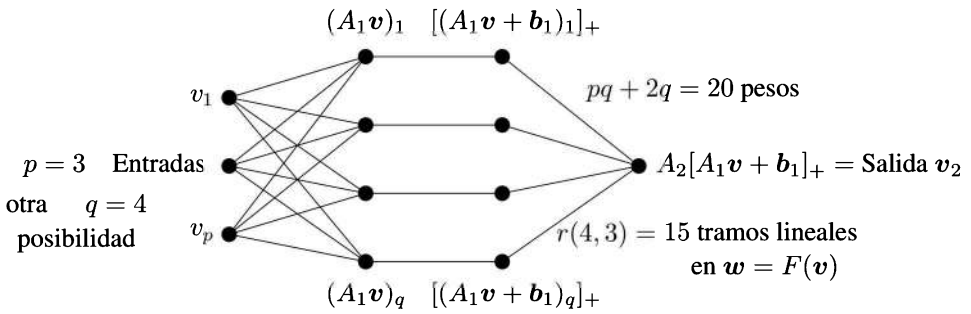
La dificultad es que la linealidad es demasiado limitada. Artísticamente dos ceros forman un 8. 1 y 0 pueden combinarse y formar un 9 o posiblemente un 6 manuscrito. Las imágenes no se suman. Para reconocer caras en lugar de números, necesitamos muchos píxeles, y la regla de entrada-salida no es para nada lineal.

La inteligencia artificial se estancó por una generación, esperando ideas nuevas. Nadie afirma que se haya descubierto la mejor clase de funciones del universo. Esa clase debe permitir muchos, muchos parámetros (llamados pesos) y se deben poder calcular todos esos pesos (en un tiempo razonable) a partir del conocimiento del conjunto de entrenamiento.

Las funciones continuas y lineales a trozos (CPL) son la opción que ha tenido un éxito que superó las expectativas: convirtió al aprendizaje superficial en aprendizaje profundo. **Lineal** por su simpleza, **continua** para modelar una regla desconocida pero razonable y **a trozos** para lograr la no linealidad esencial para las imágenes reales y los datos.

Esto nos deja la pregunta crucial sobre la computabilidad. ¿ Qué parámetros describen rápidamente una gran familia de funciones CPL ? Los elementos finitos lineales comienzan con una malla de triángulos, pero especificar muchos nodos individuales en $\mathbf{R}^p$ es costoso. Es mucho mejor si esos nodos son las *intersecciones* de una cantidad más pequeña de rectas (o hiperplanos). Tenga en cuenta que una grilla regular es demasiado simple.

He aquí una primera construcción de una función lineal a trozos del vector de datos $\boldsymbol{v}$. Elegimos una matriz A_1 y un vector $\boldsymbol{b}_1$. Luego hacemos cero todas las componentes negativas de $A_1\boldsymbol{v} + \boldsymbol{b}_1$ (paso no lineal). Luego multiplicamos por una matriz A_2 para producir 10 predicciones $\boldsymbol{w} = F(\boldsymbol{v}) = A_2(A_1\boldsymbol{v} + \boldsymbol{b}_1)_+$. Ese vector $(A_1\boldsymbol{v} + \boldsymbol{b}_1)_+$ forma una "capa oculta" entre la entrada $\boldsymbol{v}$ y la salida $\boldsymbol{w}$.



De hecho, la función no lineal llamada ReLU $(x) = x_+ = \text{máx}\,(x, 0)$ originalmente se suavizó a una curva logística como $1/(1+e^{-x})$. Era razonable pensar que las derivadas continuas ayudarían a optimizar los pesos $A_1, \boldsymbol{b}_1, A_2$. Eso resultó no ser cierto.

El gráfico de cada componente de $(A_1\boldsymbol{v}+\boldsymbol{b}_1)_+$ tiene dos semiplanos (uno es horizontal, de altura cero donde $A_1\boldsymbol{v}+\boldsymbol{b}_1$ es negativo). Si A_1 es de q por p, el espacio de entrada $\mathbf{R}^p$ es rebanado por los q hiperplanos en r partes, ¡ las cuales podemos contar ! Esto mide la "expresividad" de la función general $F(\boldsymbol{v})$. La fórmula combinatoria usa los coeficientes binomiales (vea la Sección VII.1) :

$$r(q,p) = \binom{q}{0} + \binom{q}{1} + \cdots + \binom{q}{p}$$

Este número nos da una impresión del gráfico de F. Pero nuestra función todavía no es lo suficientemente expresiva, y necesitamos una idea más.

He aquí el ingrediente indispensable en la función de aprendizaje F. La mejor manera de crear funciones complejas a partir de funciones simples es mediante la **composición**. Cada F_i es lineal (o afín) seguida por la ReLU no lineal : $F_i(\boldsymbol{v}) = (A_i\boldsymbol{v} + \boldsymbol{b}_i)_+$. Su composición es $F(\boldsymbol{v}) = F_L(F_{L-1}(\ldots F_2(F_1(\boldsymbol{v}))))$. Ahora tenemos $L-1$ capas ocultas antes de la capa de salida que contiene las predicciones finales. La red se vuelve más profunda a medida que L aumenta. Esa profundidad puede crecer rápidamente para redes convolucionales (con matrices banda A de Toeplitz).

El gran problema de optimización del aprendizaje profundo es el cálculo de los pesos A_i y $\boldsymbol{b}_i$ que harán que la respuesta $F(\boldsymbol{v})$ sea casi correcta (cercana al dígito $w(\boldsymbol{v})$ que la imagen $\boldsymbol{v}$ representa). Este problema de minimizar alguna medida de $F(\boldsymbol{v}) - w(\boldsymbol{v})$ se resuelve siguiendo un gradiente cuesta abajo. El gradiente de esta función compleja se calcula con *retropropagación*, el caballito de batalla del aprendizaje profundo que ejecuta la regla de la cadena.

Una competencia histórica en 2012 consistió en identificar las 1,2 millones de imágenes recolectadas en ImageNet. La red neuronal innovadora AlexNet tenía 60 millones de pesos. Su precisión (después de 5 días de descenso por gradiente estocástico) resultó ser la mitad del mejor error hasta ese momento. El aprendizaje profundo hacía su aparición.

Nuestro objetivo aquí fue identificar las funciones lineales a trozos como aproximantes poderosos. Esa familia también es conveniente: cerrada bajo la suma, la maximización y la composición. La magia es que la función de aprendizaje $F(A_i, \boldsymbol{b}_i, \boldsymbol{v})$ da resultados acertados con imágenes $\boldsymbol{v}$ que F jamás vio.

Este ensayo de dos páginas fue escrito para SIAM News (Diciembre de 2018).

Sesgo versus varianza : subajustar versus sobreajustar

Un **conjunto de entrenamiento** contiene N vectores $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N$ con m componentes cada uno (los m predictores de cada muestra). Para cada uno de esos N puntos en $\mathbf{R}^m$, tenemos un valor y_i. Suponemos que hay una función desconocida $f(\boldsymbol{x})$, tal que $y_i = f(\boldsymbol{x}_i) + \epsilon_i$, donde el ruido ϵ tiene media cero y varianza σ^2. Esa es la función $f(\boldsymbol{x})$ que nuestros algoritmos intentan aprender.

Nuestro algoritmo de aprendizaje en efecto encuentra una función $F(\boldsymbol{x})$ cercana a $f(\boldsymbol{x})$. Por ejemplo, la F de nuestro algoritmo de aprendizaje podría ser lineal (no muy útil) o lineal a trozos (mucho mejor), esto depende del algoritmo que usemos. Deseamos fervientemente que $F(\boldsymbol{x})$ esté cerca de la $f(\boldsymbol{x})$ correcta, **no solo en los datos de entrenamiento sino también para muestras obtenidas posteriormente**.

A menudo se repite esta advertencia, siempre vigente : **no sobreajuste los datos**. La opción de reproducir todas las observaciones conocidas está siempre disponible. Es más importante prevenir variaciones grandes en la función de aprendizaje (que se construye con los pesos). Esta función se aplicará a datos nuevos: implícita o explícitamente necesitamos **regularizar** esta función F.

Normalmente, regularizamos agregando un término de penalización como $\lambda||\boldsymbol{x}||$ a la función que estamos minimizando. Esto nos da una solución más suave y estable en el mínimo. Para los problemas de aprendizaje profundo esto no siempre es necesario. No entendemos por completo por qué, aún sin penalización, el descenso por pendiente máxima o el descenso por pendiente máxima estocástico encuentra un punto cerca del mínimo que generaliza bien para los datos de prueba no vistos. Tal vez el éxito viene de seguir esta regla : *interrumpir la minimización a tiempo antes de sobreajustar*.

> Si F tiene un desempeño pobre en las muestras de entrenamiento con un error grande (**sesgo**), eso es *subajustar*
>
> Si F tiene un buen desempeño en las muestras de entrenamiento pero no con las muestras de prueba, eso es *sobreajustar*.

Este es el **balance sesgo-varianza**. Un sesgo alto por subajustar, una varianza alta por sobreajustar. Supongamos que escalamos f y F de modo que $\mathrm{E}\,[F(\boldsymbol{x})] = 1$.

$$\textbf{Sesgo} = \mathrm{E}\,[f(\boldsymbol{x}) - F(\boldsymbol{x})] \qquad \textbf{Varianza} = \mathrm{E}\,[(F(\boldsymbol{x}))^2] - (\mathrm{E}\,[F(\boldsymbol{x})])^2$$

Nos vemos forzados a este balance debido a la identidad para (Sesgo)^2 + (Varianza) + (Ruido)^2 :

$$\mathrm{E}\,[(y - F(\boldsymbol{x}))^2] = (\mathrm{E}\,[f(\boldsymbol{x}) - F(\boldsymbol{x})])^2 + \mathrm{E}\,[(F(\boldsymbol{x}))^2] - (\mathrm{E}\,[F(\boldsymbol{x})])^2 + \mathrm{E}\,[(y - f(\boldsymbol{x}))^2]$$

Una vez más, el *sesgo* se obtiene de permitir menos libertad y del uso de menos parámetros (pesos). La *varianza* es mayor cuando permitimos demasiada libertad y demasiados parámetros para F. Así la función de aprendizaje F puede ser súper precisa en el conjunto de entrenamiento pero estar fuera de control en un conjunto de prueba no visto. **Sobreajustar produce una F que no generaliza**.

He aquí enlaces a seis sitios con códigos de aprendizaje automático :

Caffe : arXiv:1408.5093	Keras : http://keras.io/
MatConvNet : www.vlfeat.org/matconvnet	Theano : arXiv : 1605.02688
Torch : torch.ch	TensorFlow : www.tensorflow.org

VII.1. La construcción de redes neuronales profundas

Las redes neuronales profundas han evolucionado y se han convertido en una fuerza importante dentro del aprendizaje automático. Paso a paso la estructura de la red se ha vuelto más resiliente y poderosa, y se adapta con mayor facilidad a nuevas aplicaciones. Una manera de comenzar es describiendo las piezas esenciales de la estructura. Esas piezas se unen en una **función de aprendizaje** $\boldsymbol{F(x, v)}$ *con pesos* $\boldsymbol{x}$ *que capturan información de los datos de entrenamiento* $\boldsymbol{v}$, y la preparan para su uso en nuevos datos de prueba.

He aquí los pasos importantes para la creación de esa función F:

1	Operación clave	**Composición** $\boldsymbol{F(x, v) = F_3(F_2(F_1(x, v)))}$
2	Regla clave	**Regla de la cadena para las derivadas de** $\boldsymbol{F}$ **con respecto a** $\boldsymbol{x}$
3	Algoritmo clave	**Descenso por gradiente estocástico para hallar los mejores pesos** $\boldsymbol{x}$
4	Subrutina clave	**Retropropagación para ejecutar la regla de la cadena**
5	No linealidad clave	$\textbf{ReLU}(\boldsymbol{y}) = \textbf{máx}(\boldsymbol{y}, \boldsymbol{0})$ **= función rampa**

Nuestro primer paso es describir las piezas $F_1, F_2, F_3, \ldots$ para una capa de neuronas a la vez. Los pesos $\boldsymbol{x}$ que conectan las capas $\boldsymbol{v}$ se optimizan al crear $\boldsymbol{F}$. El vector $\boldsymbol{v} = \boldsymbol{v}_0$ se obtiene del conjunto de entrenamiento, y la función F_k produce el vector $\boldsymbol{v}_k$ en la capa k. Todo el éxito está en lograr una F poderosa a partir de esas piezas F_k de la ecuación (1).

F_k es una función lineal a trozos en v_{k-1}

La entrada de F_k es un vector $\boldsymbol{v}_{k-1}$ de longitud N_{k-1}. El resultado es un vector $\boldsymbol{v}_k$ de longitud N_k, listo para utilizarse como entrada para F_{k+1}. Esta función F_k tiene dos partes, primero una lineal y luego una no lineal:

1. La parte lineal de F_k da $A_k\boldsymbol{v}_{k-1} + \boldsymbol{b}_k$ (ese vector de sesgo $\boldsymbol{b}_k$ las hace "afines")
2. Una función fija no lineal como ReLU se aplica a *cada componente* de $A_k\boldsymbol{v}_{k-1} + \boldsymbol{b}_k$

$$\boxed{\boldsymbol{v}_k = F_k(\boldsymbol{v}_{k-1}) = \text{ReLU}\,(A_k\boldsymbol{v}_{k-1} + \boldsymbol{b}_k)} \tag{1}$$

Los datos de entrenamiento para cada muestra se encuentran en un vector de predictores $\boldsymbol{v}_0$. La matriz A_k es de N_k por N_{k-1}. El vector columna $\boldsymbol{b}_k$ tiene N_k componentes. **Estos** $\boldsymbol{A_k}$ **y** $\boldsymbol{b_k}$ **son pesos** construidos por el algoritmo de optimización. El paso central del aprendizaje profundo consiste en calcular los pesos óptimos $\boldsymbol{x} = (A_1, \boldsymbol{b}_1, \ldots, A_L, \boldsymbol{b_L})$ usando el *descenso por gradiente estocástico*. Se basa en la retropropagación para hallar las derivadas de F con respecto a $\boldsymbol{x}$, al resolver $\boldsymbol{\nabla F = 0}$.

La función de activación $\text{ReLU}(y) = \text{máx}(y, 0)$ da flexibilidad y adaptabilidad. Los pasos lineales tienen un poder limitado y, a fin de cuentas, fracasaron.

ReLU se aplica a cada "neurona" en cada capa interna. Hay N_k neuronas en la capa k, que contienen los N_k resultados de $A_k\boldsymbol{v}_{k-1} + \boldsymbol{b}_k$. Observe que ReLU es continua y lineal a trozos, como lo muestra su gráfico. (El gráfico es una rampa con pendientes 0 y 1. Su derivada es la función salto usual.) Cuando elegimos ReLU, la función compuesta $F = F_L \cdot\cdot (F_2(F_1(\boldsymbol{x}, \boldsymbol{v})))$ tiene una propiedad importante y atractiva:

La función de aprendizaje $\boldsymbol{F}$ **es continua y lineal a trozos en** $\boldsymbol{v}$**.**

Una capa interna ($L = 2$)

Supongamos que tenemos $m = 3$ predictores en una muestra del conjunto de entrenamiento. Esos predictores son las 3 componentes del vector de entrada $\boldsymbol{v} = \boldsymbol{v}_0$. Así la primera función F_1 en la cadena multiplica $\boldsymbol{v}_0$ por una matriz A_1 y le suma un vector de compensación (*offset*) $\boldsymbol{b}_1$ (vector de sesgo). Si A_1 es de 4 por 3 y el vector $\boldsymbol{b}_1$ es de 4 por 1, tenemos 4 componentes en $A_1\boldsymbol{v}_0 + \boldsymbol{b}_1$.

Ese paso encontró 4 combinaciones de los 3 predictores originales en $\boldsymbol{v} = \boldsymbol{v}_0$. Los 12 pesos en la matriz A_1 se optimizaron utilizando a muchos vectores predictores $\boldsymbol{v}_0$ del conjunto de entrenamiento, para elegir una matriz de 4 por 3 (y un vector de sesgo de 4 por 1) que encuentre 4 combinaciones esclarecedoras.

El paso final para obtener $\boldsymbol{v}_1$ es aplicar la "función de activación" no lineal a cada una de las 4 componentes de $A_1\boldsymbol{v}_0 + \boldsymbol{b}_1$. Históricamente, el gráfico de esa función no lineal era una curva suave con "*forma de S*". Estas elecciones particulares pueden verse en la Figura VII.1.

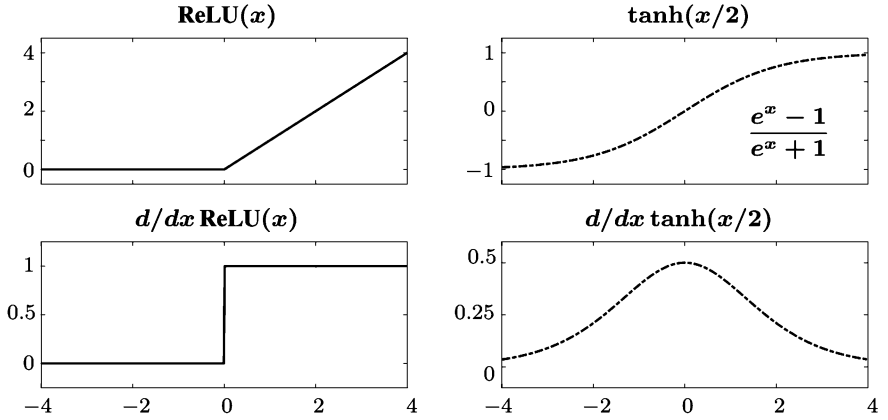


Figura VII.1: unidad lineal rectificada (**Re**ctified **L**inear **U**nit, ReLU) y opción sigmoidea.

Antes se pensaba que un cambio repentino de pendiente sería peligroso y posiblemente inestable. ¡ Pero experimentos numéricos a gran escala demostraron lo contrario ! Se obtuvo un mejor resultado con la **función rampa** $\text{ReLU}(y) = \text{máx}(y, 0)$. Trabajaremos con ReLU :

$$\boxed{\textbf{Reemplazar } \boldsymbol{A_1v_0 + b_1} \textbf{ en ReLU para encontrar } \boldsymbol{v_1} \quad (\boldsymbol{v_1})_k = \text{máx}((A_1\boldsymbol{v_0} + \boldsymbol{b}_1)_k, 0).} \tag{2}$$

Ahora tenemos las componentes de $\boldsymbol{v}_1$ en las cuatro "neuronas" de la capa 1. La capa de entrada contiene las tres componentes de esta muestra particular de datos de entrenamiento. Podemos tener miles o millones de muestras. El algoritmo de optimización encontró A_1 y $\boldsymbol{b}_1$, posiblemente mediante el descenso por gradiente estocástico con retropropagación para el cálculo de los gradientes de la pérdida total.

Supongamos que nuestra red neuronal es superficial en lugar de profunda. Solo tiene esta primera capa de 4 neuronas. Entonces el paso final multiplicará al vector $\boldsymbol{v}_1$ de 4 componentes por una matriz A_2 de 1 por 4 (un vector fila). Puede sumar un único número $\boldsymbol{b}_2$ para hallar el valor $\boldsymbol{v}_2 = A_2\boldsymbol{v}_1 + \boldsymbol{b_2}$. La función no lineal ReLU no se aplica al resultado.

$$\boxed{\begin{array}{l}\text{Calculamos } \boldsymbol{v}_2 = F(\boldsymbol{x}, \boldsymbol{v_0}) \text{ por cada vector predictor } \boldsymbol{v}_0 \text{ del conjunto de entrenamiento.} \\ \text{Los pasos son } \boldsymbol{v}_2 = A_2\boldsymbol{v}_1 + \boldsymbol{b}_2 = A_2\left(\text{ReLU}\left(A_1\boldsymbol{v}_0 + \boldsymbol{b}_1\right)\right) + \boldsymbol{b}_2 = F(\boldsymbol{x}, \boldsymbol{v_0}).\end{array}} \tag{3}$$

El objetivo al optimizar $\boldsymbol{x} = A_1, \boldsymbol{b}_1, A_2, \boldsymbol{b}_2$ es que los valores resultantes $\boldsymbol{v}_\ell = \boldsymbol{v}_2$ en la última capa $\ell = 2$ deberían capturar las características importantes de los datos de entrenamiento $\boldsymbol{v}_0$.

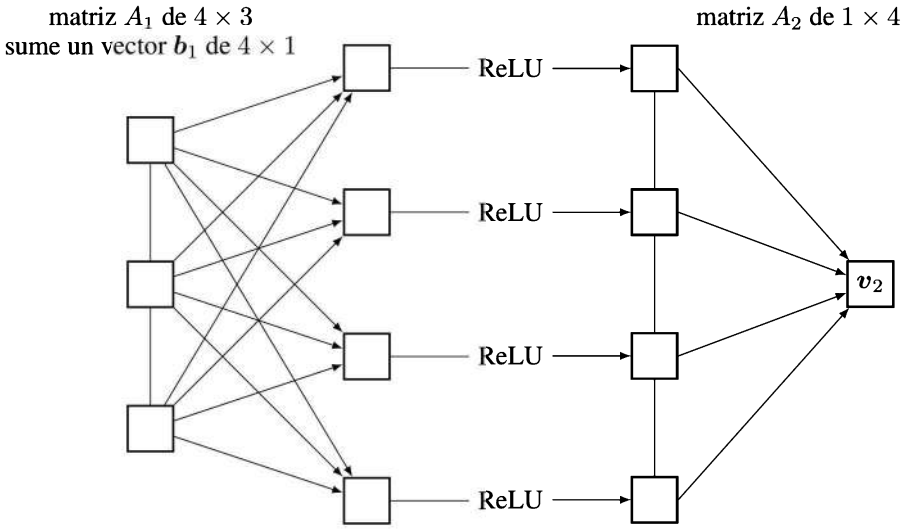


Figura VII.2: una red neuronal prealimentada (*feed-forward*) con 4 neuronas en **una capa interna**. La predicción $\boldsymbol{v}_2$ (más o menos) clasifica el elemento $\boldsymbol{v}_0$ (perro o gato). Entonces $\boldsymbol{v}_2$ es una función compuesta del vector de predictores $\boldsymbol{v}_0$ de 3 componentes. Esta red tiene 20 pesos en A_k y $\boldsymbol{b}_k$.

Para un **problema de clasificación** a cada muestra $\boldsymbol{v}_0$ de los datos de entrenamiento se le asigna $\mathbf{1}$ **o** $\mathbf{-1}$. Queremos que los resultados $\boldsymbol{v}_2$ tengan el signo correcto (la mayoría del tiempo). Para un **problema de regresión** usamos el valor numérico (**no solo el signo**) de $\boldsymbol{v}_2$. No elegimos suficientes pesos A_k y $\boldsymbol{b}_k$ para obtener todas las muestras correctas. ¡ Y no necesariamente queremos esto ! Eso probablemente sería **sobreajustar los datos de entrenamiento**. Podría dar resultados erráticos cuando se aplica F a datos de prueba nuevos y desconocidos.

Dependiendo de nuestra elección de función de pérdida $L(x, \boldsymbol{v}_2)$ a minimizar, este problema puede ser como cuadrados mínimos o minimización de entropía. Elegimos $\boldsymbol{x}$ = matrices de pesos A_k y vectores de sesgo $\boldsymbol{b}_k$ para minimizar L. Esas dos funciones de pérdida, pérdida cuadrática y pérdida de entropía cruzada, se comparan en la Sección VII.4.

Nuestra esperanza es que **la función F haya "aprendido" los datos**. Esto es el aprendizaje automático. No queremos elegir tantos pesos en $\boldsymbol{x}$ de tal manera que cada muestra ingresada sea correctamente clasificada. *Eso no es aprender*. Eso es ajustar (sobreajustar) los datos.

Queremos un balance donde la función F aprenda lo que es importante para reconocer *perro o gato, o distinguir entre auto que se acerca y un auto que dobla.*

El aprendizaje automático no busca capturar cada detalle de los números $0, 1, 2 \ldots, 9$. Tan solo busca capturar suficiente información como para decidir correctamente *cuál número es*.

Los pesos iniciales x_0 en descenso por gradiente

La arquitectura de una red neuronal decide la forma de la función de aprendizaje $F(\boldsymbol{x}, \boldsymbol{v})$. Los datos de entrenamiento van en $\boldsymbol{v}$. Luego *inicializamos* los pesos $\boldsymbol{x}$ en las matrices A y los vectores $\boldsymbol{b}$. A partir de esos pesos iniciales $\boldsymbol{x}_0$, buscando minimizar la pérdida total, el algoritmo de optimización (normalmente una forma de descenso por gradiente) calcula los pesos $\boldsymbol{x}_1$ y $\boldsymbol{x}_2$ y los que siguen.

La pregunta es : *¿ Con qué pesos $\boldsymbol{x}_0$ deberíamos comenzar* ? Elegir $\boldsymbol{x}_0 = \mathbf{0}$ sería un desastre. Una inicialización pobre es una causa importante de falla en el aprendizaje profundo. Una elección apropiada de la red y el $\boldsymbol{x}_0$ inicial deben cumplir con dos requisitos :

1. $\boldsymbol{x}_0$ tiene pesos aleatorios (e independientes) con varianza σ^2 cuidadosamente elegida.
2. Las capas ocultas en la red neuronal tienen suficientes neuronas (no son demasiado pobres).

Hanin y Rolnick muestran que la varianza inicial σ^2 controla la media de los pesos calculados. Las alturas de las capas controlan la varianza de los pesos. El punto clave es este : *muchas capas de profundidad pueden reducir la pérdida en el conjunto de entrenamiento. Pero si σ^2 es erróneo o se sacrifica la altura, entonces el descenso por gradiente puede perder el control de los pesos. Estos pueden explotar a infinito o implosionar a cero.*

Lo que controla la varianza σ^2 de $\boldsymbol{x}_0$ es el peligro de tener pesos exponencialmente grandes o exponencialmente pequeños. La buena elección es $\sigma^2 = 2/\text{fan-in}$. El fan-in (abanico de entrada) es el número máximo de neuronas de entrada (la Figura VII.2 tiene fan-in $= 4$ en los resultados). La inicialización "He uniform" en Keras utiliza esta elección de σ^2.

El peligro de las capas ocultas angostas es una varianza exponencialmente grande de $\boldsymbol{x}$ para redes profundas. Si la capa j tiene n_j neuronas, la cantidad a controlar es la suma de 1/(altura de capas n_j).

Adelantándonos, las redes convolucionales (ConvNets) y las redes residuales (ResNets) pueden ser muy profundas. Pesos que explotan o se desvanecen son un peligro constante. Ciertas ideas de la física (*como la teoría del campo medio*) se han vuelto herramientas poderosas para explicar y evitar estos peligros. Pennington y sus coautores propusieron una manera de mantenerse en el límite entre el crecimiento y el deterioro rápido, aun para 10.000 capas. Una clave es usar transformaciones ortogonales : al igual que para las multiplicaciones matriciales $Q_1 Q_2 Q_3$, la ortogonalidad no afecta el tamaño.

Para ConvNets, fan-in se transforma en la cantidad de predictores multiplicada por el tamaño del núcleo (y no el tamaño completo de A). Para ResNets, un σ^2 correcto normalmente quita ambos peligros. Las redes muy profundas pueden lograr un aprendizaje espectacular.

El punto clave : el aprendizaje profundo puede salir mal si no se empieza correctamente.

K. He, X.Zhang, S. Ren y J. Sun, *Delving deep into rectifiers*, arXiv : 1502.01852.

B. Hanin y D. Rolnick, *How to start training : The effect of initialization and architecture*, arXiv : 1803.01719, 19 de junio de 2018.

L. Xiao, Y. Bahri, J. Sohl-Dickstein, S. Schoenholz y J. Pennington, *Dynamical isometry and a mean field theory of CNNs : How to train* $10,000$ *layers*, arXiv : 1806.05393, 2018.

Segmentación y submuestreo

Esas palabras representan dos maneras de lograr el mismo objetivo : *reducir la dimensión*. Supongamos que comenzamos con una señal unidimensional de longitud 128. Queremos filtrar esa señal, multiplicar ese vector por una matriz de pesos A. También queremos reducir la longitud a 64. He aquí dos maneras de alcanzar ese objetivo.

En dos pasos Multiplicar el vector $\boldsymbol{v}$ de 128 componentes por A, y luego descartar del resultado las componentes de índice impar. Este es filtrado seguido de submuestreo. El resultado es $(\downarrow 2)\, A\boldsymbol{v}$.

En un paso Descartar las filas con índice impar de la matriz A. La nueva matriz A_2 se vuelve corta y ancha : 64 filas y 128 columnas. El "**paso**" del filtro ahora es 2. Ahora multiplicamos el vector de 128 componentes $\boldsymbol{v}$ por A_2. Así $A_2\boldsymbol{v}$ es lo mismo que $(\downarrow 2)\, A\boldsymbol{v}$. Un segmento de 3 mantendría una de cada tres componentes.

Sin duda el método de segmentación de un paso es más eficiente. Si el segmento es 4, se divide la dimensión por 4. En dos dimensiones (para imágenes) se reduce por 16.

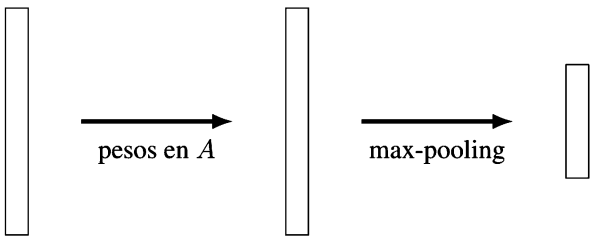
El método de dos pasos evidencia que se pierde entre la mitad y tres cuartos de la información. He aquí una manera de reducir la dimensión de 128 a 64 como antes, pero corriendo menos riesgo de destruir información importante : *Max-pooling*.

Max-pooling

Como antes, multiplicamos el vector $\boldsymbol{v}$ de 128 componentes por A. Luego, de cada dúo par-impar de resultados, como $(A\boldsymbol{v})_2$ y $(A\boldsymbol{v})_3$, *mantenemos el máximo*. Notamos esto de inmediato : el max-pooling es sencillo y rápido, **pero tomar el máximo no es una operación lineal**. Es un camino sensato a reducción de dimensiones, puro y simple.

Para una imagen (una señal de dos dimensiones) podemos usar max-pooling en cada cuadrado de 2 por 2 píxeles. Cada dimensión se reduce por 2, la dimensión de la imagen se reduce por 4. Esto acelera el entrenamiento, cuando la cantidad de neuronas en una capa oculta se divide por 4.

Normalmente a un paso de max-pooling se le da su propio lugar en la arquitectura general de la red neuronal. Por lo tanto una parte de la arquitectura podría verse así :



La reducción de dimensión tiene otra ventaja importante, además de reducir el cálculo. *El pooling también reduce la posibilidad de sobreajuste*. El pooling promedio mantiene el *promedio* de los números en cada pool : ahora la capa de pooling es lineal.

El gráfico de la función de aprendizaje $F(v)$

El gráfico de $F(\boldsymbol{v})$ es una superficie compuesta de muchísimas piezas planas, planos o hiperplanos que se encajan a lo largo de todos los pliegues donde ReLU produce un cambio de pendiente. Es como el origami, excepto que este gráfico tiene piezas planas que van hasta el infinito. Y el gráfico puede no estar en $\mathbf{R}^3$, el vector de predictores $\boldsymbol{v} = \boldsymbol{v}_0$ tiene $N_0 = m$ componentes.

Parte de la *matemática del aprendizaje profundo* es estimar la cantidad de piezas planas y visibilizar cómo se encajan en una superficie lineal a trozos. Veremos esa estimación luego de ver un ejemplo de red neuronal con una capa interna. Cada muestra del conjunto de entrenamiento tiene un vector de predictores $\boldsymbol{v}_0$ con m medidas como altura, peso y edad.

En el ejemplo, F tenía tres elementos de entrada en $\boldsymbol{v}_0$ y un resultado $\boldsymbol{v}_2$. Su gráfico será una superficie plana a trozos en un espacio de 4 dimensiones. En el punto $\boldsymbol{v}_0$ del espacio tridimensional la altura del gráfico es $\boldsymbol{v}_2 = F(\boldsymbol{v}_0)$. Las limitaciones de espacio en el libro (y las limitaciones graves de la imaginación del autor) nos impiden dibujar ese gráfico en $\mathbf{R}^4$. Sin embargo, podemos intentar contar las piezas planas, basándonos en 3 componentes de entrada, 4 neuronas y 1 resultado, salida o predicción.

Observación **1** Con tan solo $m = 2$ componentes de entrada (2 predictores para cada muestra de entrenamiento) el gráfico de F es una superficie en 3D. Intentaremos describirla.

Observación **2** Se pueden ver puntos del gráfico de F al ejecutar ejemplos del sitio web **playground.tensorflow.org**. Este es un sitio web muy didáctico.

Ese sitio web le ofrece cuatro opciones para el conjunto de entrenamiento de los puntos $\boldsymbol{v}_0$. Elija la cantidad de capas y neuronas. ¡ Elija la función de activación ReLU ! Así el programa cuenta las épocas a medida que el descenso por gradiente optimiza los pesos. (En promedio, una *época* ve todas las muestras una vez.) Si ha permitido suficientes capas y neuronas para clasificar correctamente las muestras de entrenamiento naranjas y azules, verá un polígono que las separa. **Ese polígono muestra dónde $\boldsymbol{F = 0}$.** Es una sección representativa del gráfico de $z = F(\boldsymbol{v})$ a altura $z = 0$.

Ese polígono que separa azul de naranja (o *más* de *menos* : esto es una clasificación) es análogo al hiperplano de separación en una máquina de vectores de soporte. Si estuvieramos limitados a separar con funciones lineales y a una recta entre una bola azul y un anillo naranja alrededor de ella, la separación sería imposible. Pero para la función de aprendizaje profundo F esto no es difícil...

Discutiremos experimentos del sitio **playground.tensorflow.org** en el conjunto de problemas.

Observación importante : completamente conectada versus convolucional

No queremos engañar al lector. Esas redes "completamente conectadas" a menudo no son las más efectivas. Si los pesos alrededor de un píxel en una imagen pueden repetirse alrededor de todos los píxeles (¿ por qué no ?), entonces una fila de A es todo lo que se necesita. La fila puede asignar cero pesos a los píxeles lejanos. Las **redes neuronales convolucionales locales (CNN, por su sigla en inglés)** constituyen el tema de la Sección VII.2. El sitio web **math.mit.edu/ENNUI** permite que el lector cree una CNN con pooling.

Notará que ese número de piezas planas aumenta exponencialmente junto con la cantidad de neuronas y capas. Esta es una perspectiva útil del poder del aprendizaje profundo. Necesitamos imperiosamente del conocimiento, porque el tamaño y la profundidad de la red neuronal dificultan la visualización detallada.

Contar piezas planas en un gráfico : una capa interna

Es fácil contar los elementos en las matrices A_k que contienen los pesos y los vectores de sesgo $\boldsymbol{b}_k$. Esos números determinan la función F. Pero es mucho más interesante contar la cantidad de tramos lineales en el gráfico de F. Este número mide la **expresividad** de la red neuronal. $F(\boldsymbol{x}, \boldsymbol{v})$ es una función más complicada que lo que logramos comprender (al menos por el momento). El sistema decide y actúa por su propia cuenta sin la autorización expresa de su "pensamiento". Pronto veremos las consecuencias en los coches sin conductor.

Supongamos que $\boldsymbol{v}_0$ tiene m componentes y $A_1\boldsymbol{v}_0 + \boldsymbol{b}_1$ tiene N componentes. Tenemos N funciones de $\boldsymbol{v}_0$. Cada una de esas funciones lineales es cero a lo largo de un hiperplano (dimensión $m-1$) en $\mathbf{R}^m$. Cuando aplicamos ReLU a esa función lineal se vuelve lineal a trozos, con un pliegue a lo largo de ese hiperplano. De un lado del pliegue el gráfico tiene una pendiente, y del otro lado, la función cambia de negativa a cero.

Luego la siguiente matriz A_2 combina esas N funciones lineales a trozos de v_0, así ahora tenemos pliegues a lo largo de N *hiperplanos diferentes en* $\mathbf{R}^m$. Esto describe cada componente lineal a trozos de la siguiente capa $A_2(\text{ReLU}(A_1\boldsymbol{v}_0 + \boldsymbol{b}_1))$ en el caso típico.

Podríamos pensar en N pliegues rectos en el plano (los pliegues de hecho se encuentran a lo largo de los N hiperplanos en el espacio m dimensional). El primer pliegue separa el plano en dos piezas. El siguiente pliegue de ReLU nos deja con cuatro piezas. El tercer pliegue es más difícil de visualizar, pero la Figura VII.3 muestra que hay siete piezas (*no ocho*).

En la teoría combinatoria, tenemos una **disposición de hiperplanos** y un teorema de Tom Zaslavsky que cuenta las piezas. La prueba se encuentra en el fantástico libro de Richard Stanley sobre *combinatoria enumerativa* (publicado en 2001). Pero ese teorema es más complicado que lo que necesitamos, porque permite que las rectas donde se pliegan los gráficos se encuentren de todas las maneras posibles. Nuestra tarea es más sencilla porque suponemos que las rectas donde se pliegan los gráficos se encuentran en una "posición general": $m+1$ pliegues no se cortan. Para este caso aplicamos el bello argumento de conteo de Raghu, Poole, Kleinberg, Gangul y Dickstein : *On the Expressive Power of Deep Neural Networks*, arXiv : 1606.05336v6 : vea también *The Number of Response Regions* de Pascanu, Montufar y Bengio en arXiv 1312.6098.

Teorema Para $\boldsymbol{v}$ en $\mathbf{R}^m$, supongamos que el gráfico de $F(\boldsymbol{v})$ tiene pliegues a lo largo de N hiperplanos $H_1, \dots, H_N$. Estos se obtienen de N ecuaciones lineales $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{v} + b_i = 0$, en otras palabras de ReLU en N neuronas. Entonces $r(N, m)$, la cantidad de tramos lineales de F y de regiones acotadas por los N hiperplanos es :

$$\boldsymbol{r(N, m)} = \sum_{i=0}^{m} \binom{\boldsymbol{N}}{\boldsymbol{i}} = \binom{\boldsymbol{N}}{\boldsymbol{0}} + \binom{\boldsymbol{N}}{\boldsymbol{1}} + \cdots + \binom{\boldsymbol{N}}{\boldsymbol{m}}. \tag{4}$$

Estos coeficientes binomiales son

$$\binom{\boldsymbol{N}}{\boldsymbol{i}} = \frac{N\,!}{i\,!(N-i)\,!} \quad \text{con } 0\,! = 1 \text{ y } \binom{\boldsymbol{N}}{\boldsymbol{0}} = 1 \text{ y } \binom{\boldsymbol{N}}{\boldsymbol{i}} = 0 \text{ para } i > N.$$

Ejemplo La función $F(x, y, z) = \text{ReLU}\,(x) + \text{ReLU}\,(y) + \text{ReLU}\,(z)$ tiene 3 pliegues a lo largo de los 3 planos $x = 0, y = 0, z = 0$. Esos planos dividen $\mathbf{R}^3$ en $r(3, 3) = 8$ piezas donde $F = x+y+z$, $x+z$, x y 0 (y 4 más). Agregar $\text{ReLU}\,(x+y+z-1)$ da un cuarto pliegue y así tnenemos $r(4, 3) = 15$ piezas de $\mathbf{R}^3$. No da 16 porque el nuevo plano de los pliegues $x + y + z = 1$ no alcanza a la octava pieza original dada por $x < 0, y < 0, z < 0$.

El famoso vídeo en YouTube de George Polya *Let Us Teach Guessing* corta un pastel con 5 planos. En él ayuda a encontrar $r(5,3) = 26$ porciones. La fórmula (4) permite pasteles m-dimensionales.

Un hiperplano en $\mathbf{R}^m$ produce $\binom{1}{0} + \binom{1}{1} = 2$ regiones. Y $N = 2$ hiperplanos producen $r(2,m) = 1 + 2 + 1 = 4$ regiones cuando $m > 1$. Si $m = 1$, tenemos dos pliegues en una recta, que solo separan la recta en $r(2,1) = 3$ porciones.

La cuenta r de piezas lineales sigue la fórmula recursiva

$$\boxed{r(N,m) = r(N-1,m) + r(N-1,m-1).} \tag{5}$$

Para entender esa recursión comencemos con $N-1$ hiperplanos en $\mathbf{R}^m$ y $r(N-1,m)$ regiones. *Agregamos otro hiperplano* H (dimensión $m-1$). Los $N-1$ hiperplanos ya establecidos cortan H en $r(N-1,m-1)$ regiones. Cada una de esas piezas de H divide una región existente en dos, sumando $r(N-1,m-1)$ regiones al $r(N-1,m)$ original; vea la Figura VII.3. Entonces la recursión es correcta y ahora aplicamos la ecuación (5) para calcular $r(N,m)$.

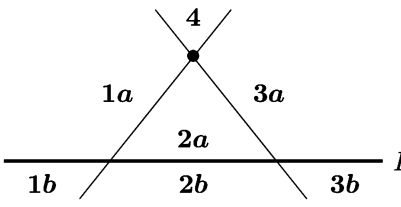
La cuenta comienza por $r(1,0) = r(0,1) = \mathbf{1}$. Así se demuestra (4) por inducción en $N+m$:

$$\begin{aligned} r(N-1,m) + r(N-1,m-1) &= \sum_0^m \binom{N-1}{i} + \sum_0^{m-1} \binom{N-1}{i} \\ &= \binom{N-1}{0} + \sum_0^{m-1} \left[\binom{N-1}{i} + \binom{N-1}{i+1} \right] \\ &= \binom{N}{0} + \sum_0^{m-1} \binom{N}{i+1} = \sum_0^m \binom{\boldsymbol{N}}{\boldsymbol{i}}. \end{aligned} \tag{6}$$

Los dos términos entre corchetes (segunda línea) se transforman en un término debido a una identidad útil:

$$\binom{N-1}{i} + \binom{N-1}{i+1} = \binom{N}{i+1} \quad \text{y la inducción está completa.}$$

Mike Giles hizo muy clara esta presentación y sugiere la Figura VII.3 para mostrar el efecto del último hiperplano H. Hay $r = 2^N$ piezas lineales de $F(\boldsymbol{v})$ para $N \le m$ y $r \approx N^m/m!$ piezas para $N >> m$, cuando la capa oculta tiene muchas neuronas.



Comenzar con 2 pliegues

$\leftarrow r(2,2) = 4$

Agregar un nuevo pliegue H

$\leftarrow r(2,1) = 3$

Total $r(3,2) = 7$

Figura VII.3: las $r(2,1) = 3$ piezas de H crean 3 regiones nuevas. Luego la cuenta se vuelve $r(3,2) = 4+3 = 7$ regiones planas en la superficie lineal a trozos continua $\boldsymbol{v}_2 = F(\boldsymbol{v}_0)$. Un cuarto pliegue cruza los 3 pliegues existentes y crea 4 regiones nuevas de modo que $r\,(4,2) = 11$.

Piezas planas de $F(v)$ con más capas ocultas

Contar las piezas lineales de $F(\boldsymbol{v})$ es mucho más difícil con 2 capas internas en la red. De nuevo $\boldsymbol{v}_0$ y $\boldsymbol{v}_1$ tienen m y N_1 componentes. Ahora $A_2\boldsymbol{v}_1 + \boldsymbol{b}_2$ tendrá N_2 componentes antes de aplicar ReLU. Cada una es como la función F para una capa antes descrita. La aplicación de ReLU creará nuevos pliegues en su gráfico. Esos pliegues se encontrarán a lo largo de las rectas donde una componente de $A_2\boldsymbol{v}_1 + \boldsymbol{b}_2$ es cero.

Recordemos que cada componente de $A_2\boldsymbol{v}_1 + \boldsymbol{b}_2$ no es lineal, sino que es lineal a trozos. Entonces cruza cero (si es que lo hace) a lo largo de una superficie lineal a trozos, no de un hiperplano. Las rectas en la Figura VII.3 para los pliegues en $\boldsymbol{v}_1$ cambiarán a *poligonales, rectas a trozos,* para los pliegues en $\boldsymbol{v}_2$. Así la cuenta se vuelve variable, dependiendo de como son $\boldsymbol{v}_0, A_1, \boldsymbol{b}_1, A_2$, y $\boldsymbol{b}_2$.

Aún podemos estimar la cantidad de tramos lineales. Tenemos N_2 líneas, rectas a trozos (o hiperplanos a trozos en $\mathbf{R}^m$) de N_2 ReLU en la segunda capa oculta. Si esas líneas fueran rectas, tendríamos un total de $N_1 + N_2$ pliegues en cada componente de $\boldsymbol{v}_3 = F(\boldsymbol{v}_0)$. *Entonces la fórmula* (4) *para contar las piezas tendría* $N_1 + N_2$ *en lugar de* N. Hanin y Rolnick confirman esta estimación (*arXiv*: 1906.00904). Entonces es la cantidad de neuronas, no de capas y profundidad, la que establece la cantidad de piezas lineales en $F(\boldsymbol{x}, \boldsymbol{v})$.

Composición $F_3(F_2(F_1(v_0)))$

La palabra "composición" representaría simplemente "multiplicación matricial" si todas nuestras funciones fueran lineales: $F_k(\boldsymbol{v}) = A_k\boldsymbol{v}$. Así $F(\boldsymbol{v}_0) = A_3A_2A_1\boldsymbol{v}_0$: una matriz. Para F_k no lineal, el significado es el mismo: calcular $\boldsymbol{v}_1 = F_1(\boldsymbol{v}_0)$, luego $\boldsymbol{v}_2 = F_2(\boldsymbol{v}_1)$, y finalmente $\boldsymbol{v}_3 = F_3(\boldsymbol{v}_2)$. Esta operación de **composición** $\boldsymbol{F_3(F_2(F_1(v_0)))}$ es mucho más poderosa para crear funciones que la suma.

Para una red neuronal, la composición produce funciones continuas lineales a trozos $F(\boldsymbol{v}_0)$. El Problema 13 en la lista de 23 problemas sin solución de Hilbert de 1900 hace una pregunta sobre *todas* las funciones continuas. Una famosa generalización de su pregunta era esta:

> ¿Son todas las funciones continuas $F(x, y, z)$ de tres variables iguales a una composición de funciones continuas $G_1, \ldots, G_N$ de dos variables? La respuesta es *sí*.

Hilbert parece haber esperado que la respuesta fuera *no*. Pero en 1957 Vladimir Arnold dio una respuesta positiva (a los 19 años). Antes, su maestro Andrey Kolmogorov había creado funciones de muchas variables a partir de funciones de tres variables.

Preguntas similares tienen respuestas negativas. Si $F(x, y, z)$ tiene derivadas continuas, puede ser imposible que todas las funciones de dos variables cuya composición da lugar a F tengan derivadas continuas (Vitushkin). Y para construir funciones continuas de dos variables $F(x, y)$ como composición de funciones continuas de una variable (el famoso problema 13°), debemos permitir *sumar*. Las funciones de dos variables xy y x^y usan funciones de una variable exp, log, y log log:

$$\boldsymbol{xy = \exp(\log x + \log y)} \quad \text{y} \quad \boldsymbol{x^y = \exp(\exp(\log y + \log\log x))}. \tag{7}$$

Hay mucho que aprender de Internet. Un capítulo de *Kolmogorov's Heritage in Mathematics* (Springer, 2007) vincula inequívocamente estas preguntas a las redes neuronales.

¿ Es afirmativa la respuesta a Hilbert para funciones continuas lineales a trozos en $\mathbf{R}^m$?

Las redes neuronales proporcionan aproximación universal

Una pregunta clave es la aproximación de una función $f(\boldsymbol{v})$ por las funciones $F(\boldsymbol{x}, \boldsymbol{v})$ de redes neuronales con pesos $\boldsymbol{x}$ cuidadosamente elegidos. He aquí tres preguntas específicas :

1 (Cualitativa) Para una función $f(\boldsymbol{v})$ continua cualquiera con $\boldsymbol{v}$ en un cubo en $\mathbf{R}^d$, ¿ puede una red con suficientes capas, neuronas y pesos $\boldsymbol{x}$ dar una aproximación uniforme a f dentro de una precisión deseada $\epsilon > 0$? Esta propiedad se llama **universalidad**.

$$\boxed{\textbf{Si } \boldsymbol{f(v)} \textbf{ es continua, existe } \boldsymbol{x} \textbf{ tal que } |F(\boldsymbol{x}, \boldsymbol{v}) - f(\boldsymbol{v})| < \boldsymbol{\epsilon} \textbf{ para todo } \boldsymbol{v}.} \tag{8}$$

2 (Cuantitativa) Si $f(\boldsymbol{v})$ pertenece a un espacio normado S de funciones suaves, ¿ qué tan rápido mejora el error de aproximación a medida que aumenta la cantidad de pesos en la red ?

$$\boxed{\textbf{Precisión de la aproximación a } \boldsymbol{f} \qquad \min_{\boldsymbol{x}} \ ||F(\boldsymbol{x}, \boldsymbol{v}) - f(\boldsymbol{v})|| \leq C||f||_S} \tag{9}$$

Los espacios funcionales S a menudo usan la norma $\boldsymbol{L^2}$, $\boldsymbol{L^1}$ o $\boldsymbol{L^\infty}$ de f y sus derivadas parciales hasta el orden r. C decrece a medida que aumenta la suavidad r. Para una aproximación continua, lineal a trozos sobre una grilla uniforme con ancho de malla h, a menudo $C = O(h^2)$.

3 ¿ Cuándo pueden las redes neuronales profundas dar una **aproximación más cercana** que los splines o las redes neuronales superficiales ?

La respuesta a la Pregunta **1** es *sí*. Wikipedia señala que una capa oculta (con suficientes neuronas) es suficiente para una aproximación con tolerancia ϵ. La prueba de George Cybenko de 1989 usa una función sigmoidea en lugar de ReLU y el teorema se sigue extendiendo o generalizando. Ding-Xuan Zhou demostró que podemos requerir que A_k sean matrices de convolución (la estructura se vuelve una CNN). Las matrices de convolución tienen muchos menos pesos que las matrices arbitrarias, y la universalidad permite muchas convoluciones.

La cota usual del error ligada a la Pregunta **2** es $Cn^{-r/d}$, cuando f tiene r variables y d derivadas. Mhaskar, Liao y Poggio introdujeron funciones compuestas construidas a partir de funciones de 2 variables, de la forma $f_3(f_1(v_1, v_2), f_2(v_1, v_2))$. La cota de error se reduce a $Cn^{-r/2}$.

Una diferencia de funciones compuestas es una composición de diferencias de 2 variables.

Están apareciendo respuestas nuevas a la Pregunta **3**. Yarotsky y Telgarsky encontraron funciones con aproximaciones super-rápidas. En *Nonlinear Approximation and (Deep) ReLU Networks* (2019), Daubechies, DeVore, Foucart, Hanin y Petrova identifican las clases de funciones generadas por las redes profundas. La composición $F_L(\ldots(F_2(F_1)))$ es la clave.

1 G. Cybenko, Approximation by superpositions of a sigmoidal function, *Mathematics of Control, Signals, and Systems* **7** (1989) 303-314.

2 K. Hornik, Approximation capabilities of multilayer feedforward networks, *Neural Networks* **4** (1991) 251-257.

3 H. Mhaskar, Q. Liao y T. Poggio, *Learning functions : When is deep better than shallow*, arXiv : 01603.00988v4, 29 de mayo de 2016.

4 D.-X. Zhou, *Universality of deep convolutional neural networks*, arXiv: 1805. 10769, 20 de julio de 2018.

5 D. Rolnick y M. Tegmark, *The power of deeper networks for expressing natural functions*, arXiv: 1705.05502, 27 de abril de 2018.

Problemas de la Sección VII.1

1 En el ejemplo $F = \text{ReLU}\,(x)+\text{ReLU}\,(y)+\text{ReLU}\,(z)$ que sigue a la fórmula (4) para $r(N, m)$, supongamos que el cuarto pliegue se obtiene de $\text{ReLU}\,(x + y + z)$. El plano del pliegue $x + y + z = 0$ ahora se corta con los 3 planos de pliegue originales $x = 0, y = 0, z = 0$ en un único punto $(0, 0, 0)$; este es *un caso excepcional*. Describa las 12 (no 15) piezas lineales de $G =$ suma de esas cuatro ReLUs.

2 Supongamos que tenemos $m = 2$ entradas y N neuronas en una capa oculta, así $F(x, y)$ es una combinación lineal de N ReLUs. Escriba la fórmula para $r(N, 2)$ para mostrar que el conteo de piezas lineales de F contiene el término principal $\frac{1}{2}N^2$.

3 Supongamos que tenemos $N = 18$ rectas en un plano. Si 9 son verticales y 9 son horizontales, ¿en cuántas piezas se divide el plano? Compare con $r(18, 2)$ donde las rectas se encuentran en una posiciones genéricas y ningún conjunto de tres rectas se cortan en un punto.

4 ¿Qué matriz de pesos A_1 y vector de sesgo $\boldsymbol{b}_1$ producirán $\text{ReLU}\,(x + 2y - 4)$, $\text{ReLU}\,(3x - y + 1)$ y $\text{ReLU}\,(2x + 5y - 6)$ como las $N = 3$ componentes de la primera capa oculta? (La capa de entrada tiene 2 componentes x e y.) Si la respuesta w es la suma de las tres ReLU, ¿cuántas piezas hay en $w(x, y)$?

5 Plegar una recta cuatro veces da $r\,(4, 1) = 5$ piezas. Plegar un plano cuatro veces da $r\,(4, 2) = 11$ piezas. Según la fórmula (4), ¿cuántos subconjuntos planos se obtienen de plegar $\mathbf{R}^3$ cuatro veces? Los subconjuntos planos de $\mathbf{R}^3$ se cortan en planos 2D (como el marco de una puerta).

6 El teorema binomial encuentra los coeficientes $\binom{N}{k}$ en $(\boldsymbol{a} + \boldsymbol{b})^N = \sum_0^N \binom{N}{k} a^k b^{N-k}$.

Para $a = b = 1$, ¿qué revela esta fórmula de esos coeficientes y de $r(N, m)$ para $m \geq N$?

7 En la Figura VII.3, un pliegue más producirá 11 piezas planas en el gráfico de $z = F(x, y)$. Verifique que la fórmula (4) da $r\,(4, 2) = 11$. ¿Cuántas piezas se obtienen luego de plegar cinco veces?

8 Explique con palabras o muestre con gráficos por qué cada una de estas afirmaciones sobre funciones continuas, lineales a trozos (funciones CPL) son verdaderas:

M	El máximo $M(x, y)$ de dos funciones CPL $F_1(x, y)$ y $F_2(x, y)$ es CPL.
S	La suma $S(x, y)$ de dos funciones CPL $F_1(x, y)$ y $F_2(x, y)$ es CPL.
C	Si las funciones de una variable $y = F_1(x)$ y $z = F_2(y)$ son CPL, también lo es la composición $C(x) = z = (F_2(F_1(x))$.

9 ¿Cuántos pesos y sesgos hay en una red con $m = N_0 = 4$ entradas en cada vector de predictores $\boldsymbol{v}_0$ y $N = 6$ neuronas en cada una de las 3 capas ocultas? ¿Cuántas funciones de activación (ReLU) hay en esta red antes del resultado final?

10 (Experimental) En una red neuronal con dos capas internas y un total de 10 neuronas, ¿debería poner mas neuronas en la capa **1** o en la capa **2**?

Los problemas 11–13 usan el ejemplo de la bola azul con el anillo naranja del sitio web playground.tensorflow.org con una capa oculta y activación con ReLU (no Tanh). Cuando el aprendizaje es exitoso, un polígono blanco separa el azul del naranja en la figura que aparece.

11 ¿El aprendizaje es exitoso para $N = 4$? ¿Cuál es el conteo $r(N, 2)$ de piezas planas en $F(\boldsymbol{v})$? El polígono blanco muestra dónde cambian de signo las piezas planas en el gráfico $F(\boldsymbol{v})$ a medida que pasan por el plano base $z = 0$. ¿Cuántos lados tiene el polígono?

12 Reduzca a $N = 3$ neuronas en una capa. ¿F sigue clasificando azul y naranja correctamente? ¿Cuántas piezas planas $r(3, 2)$ hay en el gráfico de $F(\boldsymbol{v})$ y cuantos lados tiene el polígono que los separa?

13 Reduzca aún más a $N = 2$ neuronas en una capa. ¿Sigue siendo exitoso el aprendizaje? ¿Cuál es la cantidad $r(2, 2)$ de piezas planas? ¿Cuántos pliegues hay en el gráfico de $F(\boldsymbol{v})$? ¿Cuántos lados tiene el separador blanco?

14 El Ejemplo 2 tiene dos cuadrantes azules y dos naranjas, intercalados. Con una capa, ¿$N = 3$ neuronas e incluso con $N = 2$ neuronas clasifican los datos de entrenamiento correctamente? ¿Cuántas piezas planas se necesitan para lograr el éxito? Describa el gráfico inusual de $F(\boldsymbol{v})$ para $N = 2$.

15 El Ejemplo 4 con los espirales azules y naranjas es mucho más difícil. Con una capa oculta, ¿puede la red aprender estos datos de entrenamiento? Describa los resultados a medida que N aumenta.

16 Intente ese ejemplo difícil con dos capas ocultas. Comience con $4+4$, $6+2$ y $2+6$ neuronas. ¿$2 + 6$ es mejor, peor o más inusual que $6 + 2$?

17 Si usamos dos capas ocultas ¿cuántas neuronas logran la separación completa de los espirales? ¿Tres capas ocultas pueden tener éxito con menos neuronas que dos capas?

Yo descubrí que $4+4+2$ y $4+4+4$ neuronas dan iteraciones muy inestables para ese gráfico espiralado. Había picos en la función de pérdida en los datos entrenamiento hasta que el algoritmo deja de intentar. playground.tensorflow.org (en nuestra contratapa) fue un regalo de Daniel Smilkov.

18 ¿Cuál es la menor cantidad de piezas que 20 pliegues rectos pueden producir en un plano?

19 ¿Cuántas piezas se producen a partir de 10 pliegues verticales y 10 horizontales?

20 ¿Cuál es la cantidad máxima de piezas que se pueden obtener a partir de 20 rectas de pliegues en un plano?

VII.2. Redes neuronales convolucionales

Esta sección trata sobre aquellas redes con una arquitectura distinta. Hasta ahora, cada capa estaba completamente conectada a la siguiente capa. Si una capa tenía n neuronas y la siguiente tenía m neuronas, entonces la matriz A que conectaba esas capas era de m por n. *A contenía mn pesos independientes*. Los pesos de todas las capas eran elegidos para que el resultado final coincidiera con los datos de entrenamiento. Las derivadas necesarias en esa optimización se calculaban con retropropagación. Ahora podemos tener solo 3 o 9 pesos independientes por capa.

Esa red completamente conectada es extremadamente ineficiente para el reconocimiento de imágenes. Primero, las matrices de pesos A serán gigantes. Si una imagen es de 200 por 300 píxeles, entonces su capa de entrada tiene $60,000$ componentes. La matriz de pesos A_1 para la primera capa oculta tiene $60,000$ columnas. El problema es : estamos buscando conexiones entre píxeles lejanos. Casi siempre, las conexiones importantes en una imagen son **locales**.

El texto y la música tienen una estructura local de una dimensión : una serie temporal

Las imágenes tienen una estructura local de dos dimensiones : 3 copias para rojo-verde-azul

El vídeo tiene una estructura local de tres dimensiones : imágenes en una serie temporal

Más aún, la búsqueda de estructura es esencialmente la misma en cualquier subregión de la imagen. Normalmente no hay razón por la cual procesar una parte del texto, la imagen o el vídeo de un modo distinto que las demás partes. Podemos usar los mismos pesos en todas las partes : *compartir los pesos*. La red neuronal de conexiones locales entre píxeles es **invariante por desplazamientos** : es la misma en todos lados.

El resultado es una gran reducción en la cantidad de pesos independientes. Supongamos que cada neurona está conectada a solo E neuronas de la siguiente capa, y esas conexiones son las mismas para todas las neuronas. Así la matriz A entre esas capas tiene solo E pesos independientes x. La optimización de esos pesos se acelera considerablemente, lo que nos da tiempo de crear varios canales distintos con sus propios E o E^2 pesos. Se pueden buscar aristas en diferentes direcciones (horizontal, vertical y diagonal).

En una dimensión, una matriz banda e invariante por desplazamientos es una **matriz de Toeplitz** o un **filtro**. Multiplicar por esa matriz A es una **convolución** $\boldsymbol{x} * \boldsymbol{v}$. La red de conexiones entre todas las capas es una **red neuronal convolucional** (**CNN** o **ConvNet**). En este caso $E = 3$.

$$A = \begin{bmatrix} x_1 & x_0 & x_{-1} & 0 & 0 & 0 \\ 0 & x_1 & x_0 & x_{-1} & 0 & 0 \\ 0 & 0 & x_1 & x_0 & x_{-1} & 0 \\ 0 & 0 & 0 & x_1 & x_0 & x_{-1} \end{bmatrix} \qquad \begin{array}{r} \boldsymbol{v} = (v_0, v_1, v_2, v_3, v_4, v_5) \\ \boldsymbol{y} = A\boldsymbol{v} = \quad (y_1, y_2, y_3, y_4) \\ N+2 \text{ entradas y } N \text{ respuestas} \end{array}$$

Es valioso ver a A como *una combinación de matrices de desplazamiento $\boldsymbol{L}, \boldsymbol{C}, \boldsymbol{R}$: izquierda (Left), centro (Center), derecha (Right).*

Cada desplazamiento tiene una diagonal de unos $\qquad \boldsymbol{A = x_1 L + x_0 C + x_{-1} R}$

Entonces las derivadas de $\boldsymbol{y = Av = x_1 Lv + x_0 Cv + x_{-1} Rv}$ son excepcionalmente simples :

$$\boxed{\frac{\partial \boldsymbol{y}}{\partial x_1} = L\boldsymbol{v} \qquad \frac{\partial \boldsymbol{y}}{\partial x_0} = C\boldsymbol{v} \qquad \frac{\partial \boldsymbol{y}}{\partial x_{-1}} = R\boldsymbol{v}} \tag{1}$$

Convoluciones en dos dimensiones

Cuando la entrada $\boldsymbol{v}$ es una imagen, la convolución con $\boldsymbol{x}$ se vuelve bidimensional. Los números x_{-1}, x_0, x_1 cambian a $E^2 = 3^2$ pesos independientes. Los elementos v_{ij} tienen dos índices y $\boldsymbol{v}$ representa $(N+2)^2$ píxeles. Los resultados tienen solo N^2 píxeles a menos que rellenemos con ceros en el borde. La convolución 2D $\boldsymbol{x} * \boldsymbol{v}$ es una *combinación lineal de* 9 *desplazamientos*.

Pesos $\begin{bmatrix} x_{11} & x_{01} & x_{-11} \\ x_{10} & x_{00} & x_{-10} \\ x_{1-1} & x_{0-1} & x_{-1-1} \end{bmatrix}$

Imagen de entrada v_{ij} i, j de $(0,0)$ a$(N+1, N+1)$
Imagen de salida y_{ij} i, j de $(1,1)$ a (N, N)
Desplazamientos **L, C, R, U, D** = **L**eft (izquierda), **C**enter (centro), **R**ight (derecha), **U**p (arriba), **D**own (abajo)

$$A = x_{11}LU + x_{01}CU + x_{-11}RU + x_{10}L + x_{00}C + x_{-10}R + x_{1-1}LD + x_{0-1}CD + x_{-1-1}RD$$

Esto expresa la matriz de convolución A como una combinación de 9 desplazamientos. Las derivadas de la respuesta $\boldsymbol{y} = A\boldsymbol{v}$ son excepcionalmente simples. Usamos las nueve derivadas para crear los gradientes $\boldsymbol{\nabla F}$ y $\boldsymbol{\nabla L}$ que se necesitan en el descenso por gradiente estocástico para mejorar los pesos $\boldsymbol{x}_k$. La siguiente iteración $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - s\boldsymbol{\nabla L}_k$ tiene pesos que se combinan mejor con las respuestas correctas de los datos de entrenamiento. p

Estas nueve derivadas de $\boldsymbol{y} = A\boldsymbol{v}$ se calculan dentro de la retropropagación :

$$\frac{\partial \boldsymbol{y}}{\partial x_{11}} = LU\boldsymbol{v} \quad \frac{\partial \boldsymbol{y}}{\partial x_{01}} = CU\boldsymbol{v} \quad \frac{\partial \boldsymbol{y}}{\partial x_{-11}} = RU\boldsymbol{v} \quad \dots \quad \frac{\partial \boldsymbol{y}}{\partial x_{-1-1}} = RD\boldsymbol{v} \tag{2}$$

Las CNN pueden fácilmente llegar a utilizar ***H* canales paralelos** (y ese número H puede variar a medida que nos adentramos en la red). La cantidad de pesos en $\boldsymbol{x}$ se reduce tanto a causa de los pesos compartidos y la localidad de los pesos, que no necesitamos ni esperamos que un conjunto de $E^2 = 9$ pesos haga todo el trabajo de una red convolucional.

Permítame resaltar el significado operacional de la convolución. En una dimensión, la definición algebraica formal $y_j = \sum x_i v_{j-i} = \sum x_{j-k} v_k$ usa una "inversión de índices" de los v o de los x. Esta es una fuente de confusión que no necesitamos. En su lugar, observamos los desplazamientos de izquierda a derecha L y R de toda la señal (en 1D) y también desplazamientos de arriba a abajo U y D en dos dimensiones. Cada desplazamiento es una matriz con una diagonal de unos. Esto nos ahorra la complicación de recordar subíndices invertidos.

Una convolución es una combinación de matrices de desplazamiento
(que producen un filtro o una matriz de Toeplitz)

Una convolución cíclica es una combinación de desplazamientos cíclicos
(que producen una matriz circulante)

Una convolución continua es una combinación continua (una integral) de desplazamientos

En aprendizaje profundo, los coeficientes en la combinación serán los "pesos" a aprender.

Redes convolucionales bidimensionales

Ahora llegamos al verdadero éxito de las CNN : **el reconocimiento de imágenes**. Las ConvNets y el aprendizaje profundo produjeron una pequeña revolución en la visión por computadora. Sus aplicaciones incluyen vehículos sin conductores, drones, imágenes médicas, seguridad, robótica, y no se detienen. Nuestro interés está en el álgebra y la geometría y en la intuición que hace que todo esto sea posible.

En dos dimensiones (para imágenes) la matriz A es de tipo **Toeplitz por bloques**. Cada bloque pequeño es de E por E. Esta es una estructura familiar en ingeniería computacional. La cantidad E^2 de pesos independientes a optimizar es mucho menor que la de una red completamente conectada.

Los mismos pesos se usan en todos los píxeles (*invariancia con repecto a desplazamientos*). La matriz produce una convolución 2D $\boldsymbol{x} * \boldsymbol{v}$. A menudo a A se la conoce como **filtro**.

Para entender una imagen, miremos dónde cambia. *Encontremos las aristas*. Nuestros ojos buscan cortes marcados y gradientes pronunciados. Nuestra computadora puede hacer lo mismo si creamos un filtro. Los productos punto entre una función suave y una ventana de filtro móvil resultará suave. Pero cuando una arista en una imagen se alinea con una pared diagonal, *vemos un pico*. Esos productos punto (imagen fija) $\cdot$ (imagen móvil) son exactamente la "**convolución**" de dos imágenes.

Lo difícil en dos o más dimensiones es que las aristas pueden tener muchas direcciones. Necesitamos filtros horizontales, verticales y diagonales para las imágenes de prueba. Y los filtros tienen muchos propósitos, que incluyen el suavizado, la detección de gradientes y la detección de aristas.

1 Suavizado Para una función f en 2D, el suavizado natural es la *convolución con una gaussiana* :

$$Gf(x,y) = \frac{1}{2\pi\sigma^2}\, e^{-(x^2+y^2)/2\sigma^2} * f = \frac{1}{\sqrt{2\pi}\,\sigma}\, e^{-x^2/2\sigma^2} * \frac{1}{\sqrt{2\pi}\,\sigma}\, e^{-y^2/2\sigma^2} * f(x,y)$$

Esto muestra a G como producto de funciones suaves en 1D. La gaussiana es positiva para todo x y su integral es uno, entonces *promedia* : Gf no puede tener un máximo mayor a f. El filtro quita el ruido (con un costo para las aristas filosas). Si la varianza σ^2 es pequeña, los detalles se vuelven más claros.

Para un vector 2D (una matriz f_{ij} en lugar de una función $f(x,y)$) debemos usar una gaussiana discreta. La perfección de la simetría radial se perderá porque la matriz G es cuadrada. He aquí una gaussiana discreta G de 5 por 5 $(E = 5)$:

$$G = \frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} \approx \frac{1}{289}\begin{bmatrix} 1 \\ 4 \\ 7 \\ 4 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \end{bmatrix} \tag{3}$$

Tampoco disponemos de un producto exacto de filtros en unidimensionales. Para acercarnos, usamos una matriz más grande $G = \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}$ con $\boldsymbol{x} = (.006, .061, .242, .383, .242, .061, .006)$ y descartamos los píxeles exteriores pequeños.

2 Detección de gradiente El procesamiento de imágenes (a diferencia del aprendizaje con una CNN) necesita filtros que detecten el gradiente: contienen pesos especialmente elegidos. Mencionamos algunos filtros simples para indicar cómo se pueden encontrar gradientes: las primeras derivadas de $\boldsymbol{f}$.

Una dimensión $\boldsymbol{E = 3}$ $\quad (x_1, x_0, x_{-1}) = \left(-\frac{\mathbf{1}}{\mathbf{2}}, \mathbf{0}, \frac{\mathbf{1}}{\mathbf{2}}\right)$ $\quad \left[\left(\frac{1}{2}, 0, -\frac{1}{2}\right) \text{ en forma de convolución}\right]$

En este caso las componentes de $A\boldsymbol{v}$ son diferencias centradas: $(A\boldsymbol{v})_i = \frac{1}{2}v_{i+1} - \frac{1}{2}v_{i-1}$. Cuando las componentes de $\boldsymbol{v}$ aumentan linealmente de izquierda a derecha, como en $\boldsymbol{v}_i = 3i$, el resultado del filtro es $\frac{1}{2}3(i+1) - \frac{1}{2}3(i-1) = 3 = \textit{gradiente}$.

La inversión a $\left(\frac{1}{2}, 0, -\frac{1}{2}\right)$ se obtiene de la definición de convolución como $\sum x_{i-k}v_k$.

Dos dimensiones Estos *operadores de Sobel de* 3×3 aproximan $\partial/\partial x$ y $\partial/\partial y$:

$$\boldsymbol{E = 3} \qquad \frac{\boldsymbol{\partial}}{\boldsymbol{\partial x}} \approx \frac{1}{2}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad \frac{\boldsymbol{\partial}}{\boldsymbol{\partial y}} \approx \frac{1}{2}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{4}$$

Para funciones, el vector gradiente $\boldsymbol{g} = \text{grad}\, f$ satisface $||\boldsymbol{g}||^2 = |\partial f/\partial x|^2 + |\partial f/\partial y|^2$.

Esos pesos fueron creados para el procesamiento de imágenes, para ubicar las características más importantes de una imagen típica: *sus bordes*. Estos serían candidatos para filtros de E por E dentro de una matriz convolucional A en 2D. Pero recordemos que en el aprendizaje profundo, pesos como $\frac{1}{2}$ y $-\frac{1}{2}$ no son elegidos por el usuario, sino creados a partir de los datos de entrenamiento.

Las Secciones IV.2 y IV.5 de este libro estudian convoluciones cíclicas y matrices de Toeplitz. La invarianza ante desplazamientos llevó a aplicar transformadas de Fourier discretas. Pero en una CNN, es probable que ReLU actúe sobre cada neurona. La red puede incluir rellenos con ceros y también capas de max pooling. No podemos esperar poder aplicar todo el poder del análisis de Fourier.

3 Detección de bordes Luego de estimar la dirección del gradiente buscamos los bordes, la característica más valiosa. El "*detector de bordes de Canny*" es un proceso altamente desarrollado. Ahora no queremos suavizado, ya que esto desenfocaría las aristas. Los filtros buenos se vuelven *laplacianos de gaussianas*:

$$E\, f(x,y) = \nabla^2[g(x,y) * f(x,y)] = [\nabla^2\, g(x,y)] * f(x,y). \tag{5}$$

El laplaciano $\nabla^2\, G$ de una gaussiana es $(x^2 + y^2 - 2\sigma^2)\, e^{-(x^2+y^2)/2\sigma^2}/\pi\sigma^4$.

El paso de un filtro convolucional

Importante Todos los filtros descritos hasta el momento tienen *paso* $\boldsymbol{S} = 1$. Para pasos más largos, la *ventana móvil da pasos más largos* a medida que se desplaza por la imagen. He aquí la matriz A para un filtro de dimensión 1 y 3 pesos con un paso de tamaño 2. Preste atención especialmente a que la dimensión de $\boldsymbol{y} = A\boldsymbol{v}$ se ve reducida por ese factor 2 (antes teníamos cuatro respuestas y ahora dos):

$$\textbf{Paso } \boldsymbol{S = 2} \qquad A = \begin{bmatrix} x_1 & x_0 & x_{-1} & 0 & 0 \\ 0 & 0 & x_1 & x_0 & x_{-1} \end{bmatrix} \tag{6}$$

Ahora los pesos no nulos como x_1 en A están separados por dos columnas (cuando el paso es S, están separados por S columnas). En 2D, un paso $S = 2$ reduce cada dirección por 2 y todo el resultado por 4.

Extendiendo la señal

Cuando A no es cuadrada, en lugar de perder neuronas en los bordes de la imagen, podemos *extender la capa de entrada*. "Inventamos" componentes más allá de los bordes de la imagen. Así la respuesta $\boldsymbol{y} = A\boldsymbol{v}$ se ajusta al bloque de la imagen : dimensiones iguales para la entrada y la respuesta.

El enfoque más sencillo y más popular es el **relleno con ceros** : elegimos que todas los componentes adicionales sean *ceros*. Las columnas adicionales a la izquierda y a la derecha de A multiplican esos ceros. En el medio, tenemos una matriz de Toeplitz cuadrada como en la Sección IV.5, que está determinada por un conjunto de pesos mucho menor que la cantidad de componentes en A.

Para señales periódicas, se reemplaza el rellenado con ceros por una *envolvente* (*wraparound*). La matriz de Toeplitz se vuelve circulante (Sección IV.2). La transformada de Fourier discreta nos dice sus autovalores. Los autovectores son siempre las columnas de la matriz de Fourier. La multiplicación $A\boldsymbol{v}$ es una *convolución cíclica* y entonces se aplica la regla de la convolución.

Una opción más precisa es ir más allá del borde mediante una *reflexión*. Si la última componente de la señal es v_N, y la matriz pide v_{N+1} y v_{N+2}, podemos reusar v_N y v_{N-1} (o v_{N-1} y v_{N-2}). Sea cual sea la longitud de $\boldsymbol{v}$ y el tamaño de A, todas las componentes de A se obtienen de los mismos E pesos x_{-1} a x_1 o x_{-2} a x_2 (y E^2 pesos en 2D).

Observación Otra idea. Podemos aceptar la dimensión original (128 para nuestro ejemplo) y usar la reducción a 64 como una manera de aplicar **dos filtros $\boldsymbol{C_1}$ y $\boldsymbol{C_2}$**. Cada filtro reduce la muestra de 128 a 64. La cantidad total de muestras se mantiene en 128. Si la independencia de los filtros es adecuada, no se pierde información y los 128 valores originales pueden recuperarse.

Este proceso es lineal. Se combinan dos matrices de 64 por 128 en una de 128 por 128 : una matriz cuadrada. Si como pretendemos, esa matriz es invertible, el banco de filtros *no tiene pérdidas*.

Esto es lo que usualmente hacen las CNN : agregar más canales de matrices de peso A para captar más predictores de la muestra de entrenamiento. La red neuronal tiene un banco de B filtros.

Bancos de filtros y ondículas

La idea de esos últimos párrafos produce un **banco de filtros**. Esto es tan solo un conjunto de B filtros (convoluciones) diferentes. En el procesamiento de señales un caso importante combina un filtro pasobajo C_1 con un filtro pasoalto C_2. La salida de $C_1\boldsymbol{v}$ es una señal suavizada (dominada por frecuencias bajas). La salida $C_2\boldsymbol{v}$ está dominada por frecuencias altas. Un corte perfecto por filtros ideales no puede lograrse con matrices finitas C_1 y C_2.

A partir de dos filtros tenemos un total de 256 componentes de salida. Así ambas salidas son submuestreadas. La salida tiene 128 componentes, separadas aproximadamente en *promedios y diferencias*, frecuencias bajas y frecuencias altas. La matriz es de 128 por 128.

Ondículas La idea de las ondículas es repetir los mismos pasos en las 64 componentes de la salida pasobajo $(\downarrow 2)\, C_1\boldsymbol{x}$. *Así* $(\downarrow 2)\, C_1\, (\downarrow 2)\, C_1\boldsymbol{x}$ *es un promedio de promedios*. Sus frecuencias están concentradas en el cuartil inferior ($|\omega| \leq \pi/4$) de todas las frecuencias. La salida de frecuencia intermedia $(\downarrow 2)\, C_2\, (\downarrow 2)\, C_1\boldsymbol{x}$ con 32 componentes no se subdividirá. Así $128 = 64+32+16+16$.

En el límite de subdivisión infinita es donde entran las *ondículas*. Esta separación de frecuencias bajas-altas es un tema importante en el procesamiento de señales. No es tan importante para el aprendizaje profundo, pero con múltiples canales en una CNN, la separación de frecuencias podría ser efectiva.

Contando la cantidad de entradas y salidas

Para un problema unidimensional, suponga que una capa tiene N neuronas. Aplicamos una matriz convolucional con E pesos distintos de cero. El paso es S, y rellenamos la señal de entrada con P ceros en cada extremo. ¿ Cuántos resultados (M números) produce este filtro ?

$$\boxed{\textbf{Fórmula de Karpathy} \qquad \boldsymbol{M = \frac{N - E + 2P}{S} + 1}} \tag{7}$$

En un problema 2D o 3D, esta fórmula de 1D se aplica en cada dirección.

Sea $E = 3$ y el paso $S = 1$. Si agregamos un cero ($P = 1$) en cada extremo entonces

$$M = N - 3 + 2 + 1 = N \qquad \text{(longitud de entrada = longitud de salida)}$$

Este caso $2P = E - 1$ con paso $S = 1$ es la arquitectura más común para las CNN.

Si no rellenamos la entrada con ceros, entonces $P = 0$ y $M = N - 2$ (como en la matriz A de 4 por 6 al inicio de esta sección). En 2 dimensiones esto se vuelve $M^2 = (N - 2)^2$. De esta forma perdemos neuronas pero evitamos rellenar con ceros.

Ahora supongamos que el paso es $S = 2$. Entonces $N - E$ debe ser un número par para que la fórmula (4) no resulte una fracción. He aquí dos ejemplos exitosos para el paso $S = 2$, con $N - E = 5 - 3$ y relleno $P = 0$ o $P = 1$ en ambos extremos de las cinco entradas :

Paso 2

$$\begin{bmatrix} x_{-1} & x_0 & x_1 & 0 & 0 \\ 0 & 0 & x_{-1} & x_0 & x_1 \end{bmatrix} \qquad \begin{bmatrix} x_{-1} & x_0 & x_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & x_{-1} & x_0 & x_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_{-1} & x_0 & x_1 \end{bmatrix}$$

Una vez más nuestras cuentas se aplican en cada dirección para una imagen en 2D o un tensor.

Una red convolucional profunda

Reconocer imágenes es una aplicación importante del aprendizaje profundo (y uno de sus mayores éxitos). El éxito se logró con la creación de AlexNet y el desarrollo de las redes convolucionales. Esta página describirá la red profunda de matrices convolucionales locales para el reconocimiento de imágenes. Nos basamos en el artículo premiado de Simonyan y Zisserman de ICLR 2015. Este artículo recomienda una arquitectura profunda de $L = 16$–19 capas con filtros pequeños de (3×3). La red tiene una extensión de H canales paralelos (H imágenes en cada capa).

Si la extensión B se mantuviera igual en todas las capas, y todos los filtros tuvieran pesos locales de E por E, una fórmula directa estimaría la cantidad W de pesos en la red :

$$\boxed{\boldsymbol{W \approx LBE^2} \qquad \textbf{L capas, B canales, E por E convoluciones locales}} \tag{8}$$

Observe que W no depende de la cantidad de neuronas en cada capa. Esto se debe a que A tiene E^2 pesos, sin importar su tamaño. El pooling cambiará ese tamaño sin cambiar E^2.

Pero la cantidad B de canales puede cambiar, *y es muy común cerrar una CNN con capas completamente conectadas*. Esto cambiará drásticamente la cantidad de pesos W.

Vale la pena discutir, las decisiones tomadas por Simonyan y Zisserman, junto con otras opciones. Sus elecciones llevaron a $W \approx 135,000,000$ pesos. Los cálculos se realizaron en cuatro GPU de NVIDIA, y entrenar una red llevó de 2 a 3 semanas. Es posible que el lector tenga menos potencia de cálculo (y problemas más pequeños). Entonces los hiperparámetros de la red L y B serán menores. Los principios importantes se mantendrán iguales.

Un punto clave aquí es la recomendación de reducir el tamaño E de las convoluciones locales. Se rechazaron filtros de 5 por 5 y de 7 por 7. De hecho una capa convolucional de 1 por 1 puede ser una manera de introducir un banco adicional de ReLU, como en las ResNets que veremos a continuación.

Los autores comparan tres capas de convolución, cada una con filtros de 3 por 3, con una de menos convoluciones locales de 7 por 7. Ellos comparan 27 pesos contra 49 pesos, y tres capas no lineales contra una. En ambos casos, la influencia de un solo dato se propaga, en la imagen o en las imágenes RGB ($B = 3$), a tres puntos vecinos, vertical y horizontalmente. La preferencia va a los filtros de 3 por 3 con las no linealidades adicionales que dan más neuronas por capa.

Predicciones softmax para redes multiclase

En el problema de reconocimiento de dígitos tenemos 10 resultados posibles. Para letras y otros símbolos 26 o más. Con múltiples clases en la respuesta, necesitamos una manera apropiada de decidir la última capa (la capa de las predicciones $\boldsymbol{w}$ en la red neuronal que comenzó con $\boldsymbol{v}$). "Softmax" reemplaza el caso de dos clases de la regresión logística. **Transformamos n números en probabilidades.**

Los resultados $w_1, \ldots, w_n$ se convierten en probabilidades $p_1, \ldots, p_n$ que suman 1 :

$$\textbf{Softmax} \qquad p_j = \frac{1}{S} e^{w_j} \quad \text{donde} \quad S = \sum_{k=1}^{n} e^{w_k} \tag{9}$$

Sin duda softmax asigna la mayor probabilidad p_j al mayor resultado w_j. Pero e^w es una función no lineal de w. Así la asignación softmax no es invariante por escala : si duplicamos todos los resultados w_j, softmax producirá diferentes probabilidades p_j. Para w pequeños, softmax de hecho le quita énfasis al $w_{\text{máx}}$.

En el ejemplo CNN de **teachyourmachine.com** para reconocer dígitos, verá cómo softmax produce las probabilidades mostrándolas en un gráfico de torta, una excelente ayuda visual.

CNN Necesitamos muchos pesos para ajustar a los datos, y estamos orgullosos de poder calcularlos (con la ayuda del descenso por gradiente). Pero no hay justificación para la cantidad inmensa de pesos, *si es que los pesos pueden reutilizarse*. Para señales largas en 1D y especialmente imágenes en 2D, es posible que no exista motivo para cambiar los pesos de píxel a píxel.

1. cs231n.github.io/convolutional–networks/ (karpathy@cs.stanford.edu)
2. K. Simonyan y A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, ICLR (2015), arXiv : 1409.1556v6, 10 de abril de 2015.
3. A. Krizhevsky, I. Sutskever y G. Hinton, *ImageNet classification with deep convolutional neural networks*, NIPS (2012) 1106–1114.
4. Y. LeCun y Y. Bengio, *Convolutional networks for images, speech, and time-series, Handbook of Brain Theory and Neural Networks*, MIT Press (1998).

Máquina de soporte vectorial en la última capa

Para las CNN en visión por computadora, la última capa a menudo tiene una forma especial. Si las capas anteriores usaron ReLU y max-pooling (ambos lineales a trozos), el último paso puede volverse un programa de diferencias de convexas y, eventualmente, una máquina de soporte vectorial (Support Vector Machine, SVM). Así la optimización de los pesos en una CNN lineal a trozos puede realizarse de a una capa a la vez.

L. Berrada, A. Zisserman y P. Kumar, *Trusting SVM for piecewise linear CNNs*, arXiv : 1611.02185, 6 de marzo de 2017.

El campeonato mundial de go

Un logro espectacular de una red convolucional profunda fue derrotar al campeón mundial (humano) de Go. Este es un juego difícil que se juega en un tablero de 19 por 19. Por turnos, dos jugadores colocan piedras intentando rodear las de su oponente. Cuando un grupo de un color no tiene disponible un espacio libre a su lado (izquierda, derecha, arriba o abajo), se quitan esas piedras del tablero. Wikipedia tiene un juego animado.

En 2016, AlphaGo derrotó por 4 partidas a 1 al jugador líder Lee Sedol. AlphaGo se había entrenado con miles de partidas humanas. Fue una victoria convincente pero no abrumadora. Luego se profundizó y mejoró la red neuronal. La nueva versión de AlphaGo Zero de Google aprendió a jugar sin intervención humana, simplemente jugaba contra sí misma y derrotó a su versión anterior AlphaGo 100 a 0.

El punto clave de esta versión nueva y mejorada es que **la máquina aprendió por sí misma**. Solamente se le enseñaron las reglas, y nada más. La primera versión había sido entrenada con partidas anteriores, buscando descubrir por qué los ganadores habían ganado y los perdedores habían perdido. El resultado de este nuevo enfoque fue paralelo al de la traducción automática. Para dominar un lenguaje, ciertos aspectos de su gramática parecen esenciales. ¿ Cómo aprender las excepciones ? El equipo de traducción de Google le indicaba al sistema lo que debía saber.

Mientras tanto otro equipo pequeño tomaba otro camino : que la máquina lo descubra por su propia cuenta. En ambos casos, tanto para jugar al Go como para traducir de un idioma a otro, el éxito se obtuvo con redes neuronales profundas, más partidas y sin asesoramiento.

Es la profundidad y la arquitectura de AlphaGo Zero lo que nos interesa en este momento. Los hiperparámetros se estudiarán en la Sección VII.4 : las elecciones decisivas. La historia paralela del traductor de Google debe esperar hasta VII.5 porque las redes neuronales recurrentes (**RNN**) son necesarias para capturar la estructura secuencial de un texto.

Es interesante que a menudo la máquina realiza jugadas de apertura que raramente o nunca son utilizadas por los humanos. El elemento de entrada en la red es una posición en un tablero y su historia. El vector de salida da la probabilidad de seleccionar cada jugada, y también un escalar que estima la probabilidad de ganar a partir de esa posición. Cada paso se comunica con un árbol de búsqueda de Monte Carlo, para producir *aprendizaje por refuerzo*.

Redes residuales (ResNets)

Las redes se están volviendo muy profundas con cada vez más capas ocultas. Principalmente, estas son redes convolucionales con una cantidad moderada de pesos independientes. Pero la profundidad trae peligros. La información puede atascarse y no llegar a una predicción. El problema de los "gradientes que se anulan" puede ser serio : demasiadas multiplicaciones en las propagaciones, con la consecuencia que los gradientes calculados son exponencialmente pequeños. Cuando está bien diseñada, la profundidad es muy útil, **pero debemos crear caminos para que el aprendizaje progrese**.

Lo que se destaca es que esos caminos rápidos pueden ser muy sencillos : "*saltean conexiones*" que van directamente a la capa siguiente, omitiendo el paso $\boldsymbol{v}_n = (A_n \boldsymbol{v}_{n-1} + \boldsymbol{b}_n)_+$ usual. Una propuesta eficaz de Veit, Wilber y Belongie es permitir o bien un *salto* o bien una convolución normal con un paso ReLU por vez. Si la red tiene L capas, tendremos 2^L rutas posibles, rápidas o normales de cada capa a la siguiente.

Un resultado es que capas enteras pueden quitarse sin generar un gran impacto. La n-ésima capa se alcanza con 2^{n-1} caminos posibles. Muchos caminos tienen una longitud inferior a n, sin contar los saltos. Al enviar información más lejos, los predictores que se aprenden temprano no se pierden antes de llegar a la predicción. Las redes residuales se han vuelto redes profundas muy exitosas.

K. He, X. Zhang, S. Reny J. Sun, *Deep residual learning for image recognition*, arXiv : 1512.03385, 10 de diciembre 2015. Aquí trabajan con redes neuronales extremadamente profundas al agregar atajos que saltean capas con pesos $A = I$. De otro modo, la profundidad puede dañar el desempeño.

K. He, X. Zhang, S. Ren y J. Sun, *Identity mappings in deep residual networks*, arXiv : 1603.05027, 25 de julio de 2016.

A. Veit, M. Wilber y S. Belongie, *Residual networks behave like ensembles of relatively shallow networks*, arXiv : 1605.06431, 27 de octubre de 2016.

Una CNN simple : aprender a identificar letras

Uno de los proyectos de la clase en MIT fue una red convolucional. El usuario comienza dibujando varias copias (no muchas) de A y B. En este conjunto de entrenamiento, la clasificación correcta es parte de lo que ingresa a la red el usuario. Luego viene el paso misterioso de *aprender estos datos*, creando una función lineal a trozos continua $F(\boldsymbol{v})$ que da una alta probabilidad de obtener la predicción correcta (la letra indicada).

Para aprender a leer dígitos, aparecen 10 probabilidades en un gráfico de torta. Rápidamente descubrimos que un conjunto de entrenamiento demasiado pequeño lleva a errores frecuentes. Si los ejemplos tienen números o letras centradas y las imágenes de prueba no están centradas, el usuario entiende por qué aparecen esos errores.

Un objetivo de **teachyourmachine.com** es la educación del aprendizaje automático en todos los niveles (incluso en las escuelas). Está disponible para todos los lectores.

Estas referencias finales aplican ideas muy originales del procesamiento de señales a las CNN :

R. Balestriero y R. Baraniuk, *Mad Max : Affine spline insights into deep learning*, arXiv : 1805.06576.

S. Mallat, *Understanding deep convolutional networks*, Phil. Trans. Roy. Soc. **374** (2016); arXiv : 1601.04920.

C.-C. J. Kuo, *The CNN as a guided multilayer RECOS transform*, IEEE Signal Proc. Mag. **34** (2017) 81-89; arXiv : 1701.08481.

Problemas de la Sección VII.2

1 ¿ Cuál es el máximo error en la aproximación gaussiana de la ecuación (5) ?

2 ¿ Qué matriz (que corresponda a las matrices de Sobel en la ecuación (4)) usaría para encontrar gradientes en la dirección diagonal a $45\,^\circ$?

3 (Recomendado) Para el reconocimiento de imágenes, recuerde que la muestra de entrada $\boldsymbol{v}$ es una matriz (digamos que de 3 por 3). Rellénela con ceros en todos los lados para que sea de 5 por 5. Ahora aplique una convolución como en el texto (antes de la ecuación (2)) para producir un resultado $A\boldsymbol{v}$ de 3 por 3. ¿ Cuáles son los elementos $1,1$ y $2,2$ de $A\boldsymbol{v}$?

4 He aquí dos aproximaciones matriciales L al laplaciano $\partial^2 u/\partial x^2 + \partial^2 u/\partial y^2 = \nabla^2 u$:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{y} \quad \begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix}.$$

¿ Cuáles son $\boldsymbol{LV}$ y $\boldsymbol{LD}$ para una arista correspondiente a un escalón vertical o diagonal?

$$V = \begin{bmatrix} 2 & 2 & 2 & 6 & 6 & 6 \\ 2 & 2 & 2 & 6 & 6 & 6 \\ 2 & 2 & 2 & 6 & 6 & 6 \\ 2 & 2 & 2 & 6 & 6 & 6 \end{bmatrix} \qquad D = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

5 *¿ Podría una red convolucional aprender cálculo* ? Comience con las derivadas de los polinomios de cuarto grado $p(x)$. Las entradas podrían ser los gráficos de $p = a_0 + a_1 x + \cdots + a_4 x^4$ para $0 \le x \le 1$ y un conjunto de entrenamiento de coeficientes a. Los resultados correctos serían los coeficientes $0, a_1, 2a_2, 3a_3, 4a_4$ de dp/dx. ¿ Podría diseñar y crear una CNN para que aprenda cálculo diferencial ?

6 ¿Sería más fácil o más difícil aprender cálculo integral ? Con las mismas entradas, los seis resultados serían $0, a_0, \frac{1}{2}a_1, \frac{1}{3}a_2, \frac{1}{4}a_3, \frac{1}{5}a_4$.

7 ¿ Qué tan difícil es la suma de polinomios usando dos gráficos como entrada ? Los resultados del conjunto de entrenamiento serían las sumas correctas $a_0 + b_0, \ldots, a_4 + b_4$ de los coeficientes. ¿ La multiplicación de polinomios es difícil con 9 resultados $a_0 b_0, a_0 b_1 + a_1 b_0, \ldots, a_4 b_4$?

En los problemas 5-7 los datos son los gráficos. Cleve Moler informó sobre experimentos en :

https://blogs.mathworks.com/cleve/2018/08/06/teaching–calculus–to–a–deep–learner

Además **.. 2018/10/22/teaching–a–newcomer–about–teaching–calculus–to–a–deep–learner**

Está emergiendo una teoría del aprendizaje profundo para la física oculta : por ejemplo, vea arXiv : 1808.04327.

VII.3. Retropropagación y la regla de la cadena

El aprendizaje profundo es fundamentalmente un problema gigante de optimización de la función de aprendizaje F. Elegimos "pesos" numéricos que minimicen la función de pérdida L (que depende de esos pesos). $L(\boldsymbol{x})$ es la suma de todas las pérdidas $\boldsymbol{\ell}(\boldsymbol{w} - \textbf{true}) = \ell(F(\boldsymbol{x}, \boldsymbol{v}) - \text{true})$ entre las predicciones $\boldsymbol{w} = F(\boldsymbol{x}, \boldsymbol{v})$ y las verdaderas clasificaciones de los elementos $\boldsymbol{v}$. El cálculo nos da las ecuaciones a resolver para encontrar los pesos que minimizan L:

Las derivadas parciales de L con respecto a los pesos x deberían ser cero.

El descenso por gradiente en todas sus variaciones necesita *calcular derivadas* (componentes del gradiente de F) en los valores actuales de los pesos. Las derivadas $\partial F/\partial \boldsymbol{x}$ llevan a $\partial L/\partial \boldsymbol{x}$. A partir de esta información podemos encontrar nuevos pesos que dan una pérdida menor. Así recalculamos las derivadas de F y L en los nuevos valores de los pesos y repetimos.

La retropropagación es un método rápido para calcular derivadas, con la regla de la cadena:

Regla de la cadena
$$\frac{dF}{dx} = \frac{d}{dx}(F_3(F_2(F_1(x)))) = \left(\frac{dF_3}{dF_2}(F_2(F_1(x)))\right)\left(\frac{dF_2}{dF_1}(F_1(x))\right)\left(\frac{dF_1}{dx}(x)\right)$$

Esa afirmación de la regla de la cadena se aplica a una función $F(x)$ de una variable. Pero cuando tenemos muchos nodos (neuronas) en cada capa de una red neuronal, tenemos funciones de varias variables. La regla de la cadena aún se aplica, excepto que ahora tenemos una matriz (y a veces un tensor de orden 3) en cada paso. Esto lleva a dos preguntas:

1. ¿Cual es la regla de la cadena multivariable?
2. ¿Cuál orden de multiplicación (hacia adelante o hacia atrás sobre la cadena) es más rápido?

Permítame responder la Pregunta 2 de inmediato, con un caso sencillo pero importante. Supongamos que la regla de la cadena tiene tres factores $M_1M_2\boldsymbol{w}$, dos matrices y un vector. *¿Multiplicamos primero las matrices M_1M_2 o multiplicamos primero $M_2\boldsymbol{w}$?*

Para matrices de N por N, M_1M_2 requiere de N^3 multiplicaciones separadas. $M_2\boldsymbol{w}$ tiene N^2.

$(M_1M_2)\boldsymbol{w}$ necesita $N^3 + N^2$ multiplicaciones $\qquad$ $M_1(M_2\boldsymbol{w})$ necesita solo $N^2 + N^2$

Esta es una diferencia importante. Y si tenemos una cadena de L matrices a partir de L capas en la red neuronal, la diferencia es esencialmente un factor N:

Hacia adelante $(((M_1M_2)M_3)\dots M_L)\boldsymbol{w}$ necesita $(L-1)N^3 + N^2$

Modo reverso (hacia atrás) es mejor. **Hacia atrás** $M_1(M_2(\dots(M_L\boldsymbol{w})))$ necesita LN^2

La retropropagación se ha descubierto muchas veces. Otro de sus nombres es **diferenciación automática (automatic differentiation, AD)**. Los pasos pueden acomodarse de dos modos básicos: el **modo hacia adelante** y el **modo hacia atrás**.La elección correcta del modo puede hacer una gran diferencia en el costo (un factor de miles). Esa elección depende de si tenemos muchas funciones que dependen de unas pocas variables o unas pocas funciones que dependen de muchas variables.

El aprendizaje profundo básicamente tiene una función de pérdida que depende de muchos pesos.

Para minimizar una función $F(\boldsymbol{x})$, necesitamos derivadas. Queremos la dirección del descenso con pendiente máxima en cada paso. Para el paso k que sale de $\boldsymbol{x}_k$, esa dirección está dada por el vector gradiente $\nabla_{\boldsymbol{x}} \boldsymbol{F} = (\partial F/\partial x_1, \ldots, \partial F/\partial x_N)$ en el punto $\boldsymbol{x}_k$.

La regla de la cadena multivariada

Supongamos que el vector $\boldsymbol{v}$ con n componentes v_i es una función del vector $\boldsymbol{u}$ con m componentes u_j. La derivada de cada v_i con respecto a cada u_j va en la matriz de derivadas $\partial \boldsymbol{v}/\partial \boldsymbol{u}$ (a menudo llamada matriz jacobiana J). Su tamaño es de n por m (y no de m por n). De manera similar, la matriz de derivadas $\partial \boldsymbol{w}/\partial \boldsymbol{v}$ de la función vectorial $\boldsymbol{w} = (w_1, \ldots, w_p)$ con respecto a las componentes de $\boldsymbol{v} = (v_1, \ldots, v_n)$ es una matriz de p por n :

$$\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}} = \begin{bmatrix} \frac{\partial w_1}{\partial v_1} & \cdots & \frac{\partial w_1}{\partial v_n} \\ & \cdots & \\ \frac{\partial w_p}{\partial v_1} & \cdots & \frac{\partial w_p}{\partial v_n} \end{bmatrix} \qquad \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{u}} = \begin{bmatrix} \frac{\partial v_1}{\partial u_1} & \cdots & \frac{\partial v_1}{\partial u_m} \\ & \cdots & \\ \frac{\partial v_n}{\partial u_1} & \cdots & \frac{\partial v_n}{\partial u_m} \end{bmatrix} \tag{1}$$

Cada w_i depende de los v y cada v_j depende de los u. Por lo tanto cada función $w_1 \ldots, w_p$ depende de $u_1, \ldots, u_m$. La regla de la cadena busca encontrar las derivadas $\partial w_i/\partial u_k$. Y la regla es exactamente un producto punto : (fila i de $\partial \boldsymbol{w}/\partial \boldsymbol{v}$) $\cdot$ (columna k de $\partial \boldsymbol{v}/\partial \boldsymbol{u}$) en (1).

$$\frac{\partial w_i}{\partial u_k} = \frac{\partial w_i}{\partial v_1}\frac{\partial v_1}{\partial u_k} + \cdots + \frac{\partial w_i}{\partial v_n}\frac{\partial v_n}{\partial u_k} = \left(\frac{\partial w_i}{\partial v_1}, \ldots, \frac{\partial w_i}{\partial v_n}\right) \cdot \left(\frac{\partial v_1}{\partial u_k}, \ldots, \frac{\partial v_n}{\partial u_k}\right)$$

Regla de la cadena multivariada : multiplicar matrices en (1) $\dfrac{\partial \boldsymbol{w}}{\partial \boldsymbol{u}} = \left(\dfrac{\partial \boldsymbol{w}}{\partial \boldsymbol{v}}\right)\left(\dfrac{\partial \boldsymbol{v}}{\partial \boldsymbol{u}}\right)$ (2)

Observación : Estas dos páginas se basan en el maravilloso proyecto 18.065 de Alex Craig. Su proyecto (con código para retropropagación) se encuentra en **math.mit.edu/learningfromdata**.

Los pesos en el aprendizaje profundo de hecho son matrices A en lugar de vectores. Cada peso tiene dos índices, no uno. Las derivadas $\partial w_i/\partial A_{jk}$ tendrán tres índices, no dos. Así tenemos estos tensores de orden tres en la regla de la cadena. Por suerte estos tensores son muy sencillos (están calculados en la Sección I.12 y los calculamos de nuevo a continuación). Tienen muchas componentes nulas y la regla de la cadena no es abrumadora :

$$\frac{\partial w_i}{\partial A_{jk}} = \sum_{t=1}^{n} \frac{\partial w_i}{\partial v_t}\frac{\partial v_t}{\partial A_{jk}}$$

La retropropagación es una mejora maravillosamente eficaz en el cálculo por separado de cada $\partial F/\partial x_i$. Al principio parece difícil de creer que reorganizar los cálculos pueda hacer una diferencia tan grande. Al final (podrían decir los incrédulos) hay que calcular las derivadas para cada paso y multiplicar por la regla de la cadena. Pero el reordenamiento funciona: las N derivadas se calculan en mucho menos tiempo que N multiplicado por el costo de una derivada $\partial F/\partial x_1$.

La retropropagación es una diferenciación automática hacia atrás (*AD hacia atrás*).

Producto de matrices ABC : ¿ qué orden ?

La decisión entre el orden hacia adelante y hacia atrás también aparece en la multiplicación de matrices. Si se nos pide multiplicar A por B por C, la ley asociativa ofrece dos opciones para el orden de la multiplicación :

¿ Primero AB o primero BC ? **¿Calcular $(AB)\,C$ o $A(BC)$?**

El resultado es el mismo pero la cantidad de multiplicaciones individuales puede ser muy diferente. Supongamos que la matriz A es de $\boldsymbol{m}$ por $\boldsymbol{n}$, y B de $\boldsymbol{n}$ por $\boldsymbol{p}$, y C de $\boldsymbol{p}$ por $\boldsymbol{q}$.

Primera forma
$$AB = (m \times n)\,(n \times p) \text{ tiene } \boldsymbol{mnp} \text{ multiplicaciones}$$
$$(AB)C = (m \times p)\,(p \times q) \text{ tiene } \boldsymbol{mpq} \text{ multiplicaciones}$$

Segunda forma
$$BC = (n \times p)\,(p \times q) \text{ tiene } \boldsymbol{npq} \text{ multiplicaciones}$$
$$A(BC) = (m \times n)\,(n \times q) \text{ tiene } \boldsymbol{mnq} \text{ multiplicaciones}$$

Entonces la comparación es entre $mp(n+q)$ y $nq(m+p)$. Dividimos ambos números por $mnpq$:

La primera forma es más rápida cuando $\dfrac{1}{q} + \dfrac{1}{n}$ es más pequeño que $\dfrac{1}{m} + \dfrac{1}{p}$.

He aquí un caso extremo (extremadamente importante). *Supongamos que C es un vector columna* : de p por 1. O sea $q = 1$. ¿ Deberíamos multiplicar BC para obtener otro vector columna (n por 1) y luego $A(BC)$ para encontrar el resultado (m por 1) ? ¿ O deberíamos multiplicar AB primero ?

La pregunta casi se responde sola. El $A(BC)$ correcto produce un vector en cada paso. La multiplicación matriz-vector BC tiene np pasos. La siguiente multiplicación matriz-vector $A(BC)$ tiene mn pasos. Comparemos esos $np + mn$ pasos con el costo de comenzar con la opción matriz-matriz AB (mnp pasos). Nadie en sus cabales haría eso.

Pero si A es un vector fila, $(AB)C$ es mejor. Cada vez, una fila por una matriz.

Cálculo central del aprendizaje profundo : **Entrenar la red = optimizar los pesos.** El resultado $F(\boldsymbol{v})$ de una red profunda es una cadena que comienza con $\boldsymbol{v}$:

$F(\boldsymbol{v}) = A_L \boldsymbol{v}_{L-1} = A_L(\mathrm{R}\, A_{L-1}(\ldots(\mathrm{R}\, A_2(\mathbf{R}\, A_1 \boldsymbol{v}))))$ hace un recorrido hacia adelante por la red.

Las derivadas de F con respecto a las matrices A (y los vectores de sesgo $\boldsymbol{b}$) son los más fáciles para la *última matriz* A_L en $A_L \boldsymbol{v}_{L-1}$. La derivada de $A\boldsymbol{v}$ con respecto a A contiene los v :

$\dfrac{\partial F_i}{\partial A_{jk}} = \delta_{ij}\, v_k$. Luego viene la derivada de $A_L\,\mathrm{ReLU}\,(A_{L-1} v_{L-1})$ con respecto a A_{L-1}.

Explicaremos cómo ese mismo modo reverso también aparece cuando se comparan los *métodos directos con los métodos duales en optimización* (elegir pesos buenos).

Las derivadas $\partial F/\partial x$ de la función de aprendizaje $F(x, v)$

Los pesos x son todas las matrices $A_1, \ldots, A_L$ y todos los vectores de sesgo $b_1, \ldots, b_L$. Las entradas $v = v_0$ son el conjunto de entrenamiento. Las predicciones $w = F(x, v_0)$ aparecen en la capa L.

Así $w = v_L$ es el último paso en la red neuronal, luego de $v_1, \ldots, v_{L-1}$ en las capas ocultas.

Cada capa nueva v_n se obtiene de la capa anterior con R $(b_n + A_n v_{n-1})$. Aquí R es la función de activación no lineal (usualmente ReLU) aplicada a cada componente.

Así el aprendizaje profundo nos lleva de $v = v_0$ a $w = v_L$. Luego sustituimos w en la función pérdida para medir el error de esa muestra v: puede ser un error de clasificación : 0 en lugar de 1, o 1 en lugar de 0. Puede ser un error de regresión vía mínimos cuadrados $||g - w||^2$, con w en lugar del resultado deseado g. A menudo es una "entropía cruzada". La pérdida total $L(x)$ es la suma de las pérdidas para todos los vectores de entrada v.

La optimización gigante del aprendizaje profundo busca encontrar los pesos x que minimizan L. Para el descenso por gradiente completo, la pérdida es $L(x)$. Para el descenso por gradiente estocástico la pérdida en cada iteración es $\ell(x)$, utilizando un solo dato o un minilote de datos. **En todos los casos necesitamos las derivadas $\partial w/\partial x$ de las predicciones w** (las componentes de la última capa) con respecto a los pesos x (las A y los b que nos llevan de capa en capa).

Este es un motivo por el cual el aprendizaje profundo es tan costoso y requiere de tanto tiempo. Para las redes convolucionales, las derivadas se encontraron rápido y con facilidad en la Sección VII.2.

Cálculo de $\partial F/\partial x$: fórmulas explícitas

Comencemos con el último vector de sesgo b_L y la matriz de pesos A_L que producen la predicción final $v_L = w$. No hay no linealidad en esta capa y descartamos el índice de la capa L :

$$v_L = b_L + A_L v_{L-1} \quad \text{o simplemente} \quad w = b + Av. \tag{3}$$

Nuestro objetivo es encontrar las derivadas $\partial w_i/\partial b_j$ y $\partial w_i/\partial A_{jk}$ para todas las componentes de $b + Av$. Cuando j es diferente de i, *el iésimo resultado w_i no se ve afectado por b_j ni A_{jk}*. Multiplicando A por v, la fila j de A produce w_j y no w_i. Introducimos el símbolo δ que es 1 o 0 :

$$\delta_{ij} = 1 \text{ si } i = j \qquad \delta_{ij} = 0 \text{ si } i \neq j \qquad \text{La matriz identidad } I \text{ tiene elementos } \delta_{ij}.$$

Las columnas de I son **vectores con una sola componente no nula**. Las derivadas son 1 o 0 o v_k (Sección I.12) :

Capa completamente conectada

Pesos independientes A_{jk}

$$\boxed{\frac{\partial w_i}{\partial b_j} = \delta_{ij} \quad \text{y} \quad \frac{\partial w_i}{\partial A_{jk}} = \delta_{ij} v_k} \tag{4}$$

Ejemplo Hay seis b y a en $\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} + \begin{bmatrix} a_{11}v_1 + a_{12}v_2 \\ a_{21}v_1 + a_{22}v_2 \end{bmatrix}$.

Derivadas de w_1 $\dfrac{\partial w_1}{\partial b_1} = 1, \dfrac{\partial w_1}{\partial b_2} = 0, \dfrac{\partial w_1}{\partial a_{11}} = v_1, \dfrac{\partial w_1}{\partial a_{12}} = v_2, \dfrac{\partial w_1}{\partial a_{21}} = \dfrac{\partial w_1}{\partial a_{22}} = 0.$

Derivadas para las capas ocultas

Ahora supongamos que hay una capa oculta, es decir $L = 2$. La predicción es $\boldsymbol{w} = \boldsymbol{v}_L = \boldsymbol{v}_2$, la capa oculta contiene $\boldsymbol{v}_1$, y la entrada es $\boldsymbol{v}_0 = \boldsymbol{v}$. La R no lineal probablemente sea ReLU.

$$\boldsymbol{v}_1 = \mathrm{R}\,(\boldsymbol{b}_1 + A_1\boldsymbol{v}_0) \quad \text{y} \quad \boldsymbol{w} = \boldsymbol{b}_2 + A_2\boldsymbol{v}_1 = \boldsymbol{b}_2 + A_2\mathrm{R}\,(\boldsymbol{b}_1 + A_1\boldsymbol{v}_0).$$

La ecuación (4) sigue dando las derivadas de $\boldsymbol{w}$ con respecto a los últimos pesos $\boldsymbol{b}_2$ y A_2. La función R está ausente en la predicción y $\boldsymbol{v}$ es $\boldsymbol{v}_1$. Pero las derivadas de $\boldsymbol{w}$ con respecto a $\boldsymbol{b}_1$ y A_1 involucran la función no lineal R actuando sobre $\boldsymbol{b}_1 + A_1\boldsymbol{v}_0$.

Las derivadas parciales en $\partial \boldsymbol{w}/\partial A_1$ necesitan la regla de la cadena $\partial f/\partial x = (\partial f/\partial g)(\partial g/\partial x)$:

$$\textbf{Regla de la cadena} \quad \frac{\partial \boldsymbol{w}}{\partial A_1} = \frac{\partial[A_2\mathrm{R}\,(\boldsymbol{b}_1 + A_1\boldsymbol{v}_0)]}{\partial A_1} = A_2\,\mathrm{R}\,'(\boldsymbol{b}_1 + A_1\boldsymbol{v}_0)\,\frac{\partial(\boldsymbol{b}_1 + A_1\boldsymbol{v}_0)}{\partial A_1}. \quad (5)$$

Esta regla de la cadena tiene tres factores. Comenzando por $\boldsymbol{v}_0$ en la capa $L - 2 = 0$, los pesos $\boldsymbol{b}_1$ y A_1 nos llevan hacia la capa $L - 1 = 1$. Las derivadas de ese paso son exactamente como en la ecuación (4). Pero el resultado de ese paso parcial no es $\boldsymbol{v}_{L-1}$. Para encontrar esa capa oculta primero tenemos que aplicar R. Así la regla de la cadena incluye su derivada $\mathbf{R}'$. Luego el paso final (a $\boldsymbol{w}$) multiplica por la última matriz de pesos A_2.

En los problemas al final de la sección se extienden estas fórmulas a L capas. Estas podrían ser útiles, pero con el pooling y la normalización de lotes, la diferenciación automática parece derrotar a la programación compleja.

Muy importante Observe cómo las fórmulas como (5) van **hacia atrás, de $\boldsymbol{w}$ a $\boldsymbol{v}$**. La retropropagación automática hace lo mismo. El "**Modo reverso**" comienza con la predicción.

Detalles de las derivadas $\partial \boldsymbol{w}/\partial \boldsymbol{A}_1$

Tenemos la responsabilidad de mirar con un poco más de detalle a la ecuación (5). Su parte no lineal $\mathbf{R}'$ se obtiene de la derivada de la función de activación no lineal. La elección usual es la función rampa ReLU $(x) = (x)_+ =$ caso límite de una función sigmoidea con forma de S.

$$d\mathrm{R}/dx = \begin{cases} 0 & x < 0 \\ 1 & x > 0 \end{cases} \qquad \mathbf{R}'(\boldsymbol{b}_1 + A_1\boldsymbol{v}_0) = \begin{cases} 0 & \text{cuando } (\boldsymbol{b}_1 + A_1\boldsymbol{v}_0)_i < 0 \\ 1 & \text{cuando } (\boldsymbol{b}_1 + A_1\boldsymbol{v}_0)_i > 0 \end{cases}$$

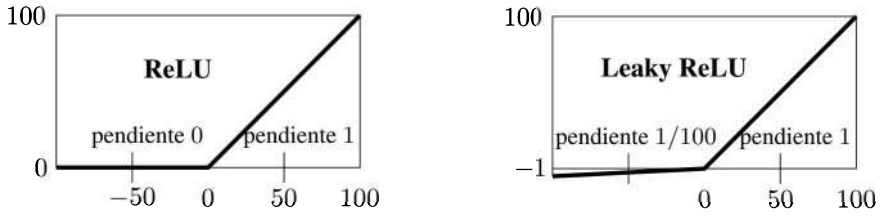


Figura VII.4: Gráficos de ReLU y Leaky ReLU (dos opciones para la activación no lineal).

Regresemos a la fórmula (5), escribamos A y $\boldsymbol{b}$ para la matriz A_{L-1} y el vector $\boldsymbol{b}_{L-1}$ que producen la última capa oculta. Así ReLU, A_L y $\boldsymbol{b}_L$ producen la salida final $\boldsymbol{w} = \boldsymbol{v}_L$. Nuestro interés está en $\partial w/\partial A$, cómo depende $\boldsymbol{w}$ de la penúltima matriz de pesos.

$$\boxed{\boldsymbol{w} = A_L(R(A\boldsymbol{v} + \boldsymbol{b})) + \boldsymbol{b}_L \quad \text{y} \quad \frac{\partial \boldsymbol{w}}{\partial A} = A_L R'(A\boldsymbol{v} + \boldsymbol{b})\frac{\partial(A\boldsymbol{v} + \boldsymbol{b})}{\partial A}} \tag{6}$$

Consideramos R una matriz diagonal de funciones ReLU que actúa componente por componente en $A\boldsymbol{v} + \boldsymbol{b}$. Entonces $J = R'(A\boldsymbol{v} + \boldsymbol{b})$ es una matriz diagonal con un 1 por cada componente positiva y un 0 por cada componente negativa. (No pregunte por los ceros.)

$$\boldsymbol{w} = A_L\, R\,(A\boldsymbol{v} + \boldsymbol{b}) \quad \text{y} \quad \frac{\partial \boldsymbol{w}}{\partial A} = A_L\, J\, \frac{\partial(A\boldsymbol{v} + \boldsymbol{b})}{\partial A} \tag{7}$$

Conocemos cada componente (v_k o cero) a partir del tercer factor de las derivadas en (4).

Cuando la función sigmoidea R_a reemplaza a la función ReLU, la matriz diagonal $J = R_a'(A\boldsymbol{v}+\boldsymbol{b})$ ya no contiene ningún 1 ni 0. Ahora evaluamos la derivada dR_a/dx en cada componente de $A\boldsymbol{v} + \boldsymbol{b}$.

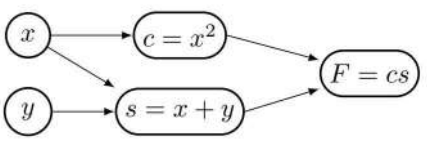
En la práctica, la retropropagación encuentra las derivadas con respecto a las A y las $\boldsymbol{b}$, y crea esas derivadas automáticamente (y efectivamente).

Grafos computacionales

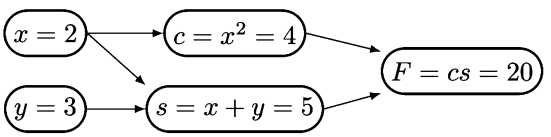
Supongamos que $F(x, y)$ es una función de dos variables x e y. Esos elementos son los primeros dos nodos en el grafo computacional. Un paso típico en el cálculo, **una arista en el grafo**, es una de las operaciones aritméticas (suma, resta, multiplicación...). El resultado final es la función $F(x, y)$. Nuestro ejemplo será $\boldsymbol{F = x^2(x + y)}$.

Los grafos computacionales fueron inspirados por la brillante exposición de Christopher Olah (**colah.github.io**) y (**https://distill.pub**). Catherine y Desmond Higham prestan especial atención a la retropropagación, con códigos (*SIAM Review* y arXiv : 1801.05894).

He aquí el grafo que calcula F con nodos intermedios $c = x^2$ y $s = x + y$:



Cuando tenemos los elementos x e y, por ejemplo $\boldsymbol{x = 2}$ e $\boldsymbol{y = 3}$, las aristas llevan a $\boldsymbol{c = 4}$, $\boldsymbol{s = 5}$ y $\boldsymbol{F = 20}$. Esto concuerda con el álgebra que usualmente compactamos en una línea : $F = x^2(x + y) = 2^2(2 + 3) = 4(5) = 20$.



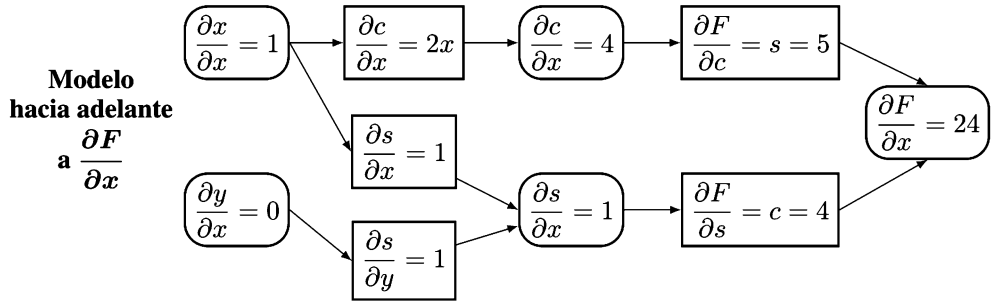
Ahora calculamos la derivada de cada paso: cada arista del grafo. Comencemos con la derivada con respecto a x. Primero elegimos el *modo hacia adelante*, empezando por el elemento x y nos movemos hacia la función $x^2(x+y)$. Entonces los primeros pasos usan la regla de la potencia para $c = x^2$ y la regla de la suma para $s = x+y$. En el último paso se aplica la regla del producto a $F = c$ veces s.

$$\frac{\partial c}{\partial x} = 2x \qquad \frac{\partial s}{\partial x} = 1 \qquad \frac{\partial F}{\partial c} = s \qquad \frac{\partial F}{\partial s} = c$$

¡ Al movernos por el grafo se produce la regla de la cadena !

$$\begin{aligned}\frac{\boldsymbol{\partial F}}{\boldsymbol{\partial x}} &= \frac{\partial F}{\partial c}\,\frac{\partial c}{\partial x} + \frac{\partial F}{\partial s}\,\frac{\partial s}{\partial x} \\ &= (s)(2x) + (c)(1) = (5)(4) + (4)(1) = \mathbf{24}\end{aligned}$$

El resultado final es calcular la derivada de F con respecto a *una entrada* x. Podemos ver esas derivadas con respecto a x en el grafo computacional.



Habrá un grafo similar para las derivadas con respecto a y: el modo hacia adelante que lleva a $\partial F/\partial y$. He aquí la regla de la cadena y los números que aparecerían en ese grafo para $x = 2$, $y = 3$, $c = x^2 = 2^2$, $s = x + y = 2 + 3$ y $F = cs$ son :

$$\begin{aligned}\frac{\boldsymbol{\partial F}}{\boldsymbol{\partial y}} &= \frac{\partial F}{\partial c}\,\frac{\partial c}{\partial y} + \frac{\partial F}{\partial s}\,\frac{\partial s}{\partial y} \\ &= (s)(0) + (c)(1) = (5)(0) + (4)(1) = \mathbf{4}\end{aligned}$$

El grafo computacional para $\partial F/\partial y$ no lo hicimos pero esto es lo importante : *el modo hacia adelante requiere un nuevo grafo para cada entrada x_i, para calcular la derivada parcial $\partial F/\partial x_i$.*

Grafo en modo reverso para un resultado

El modo reverso comienza con el resultado o salida F. **Calcula las derivadas con respecto a ambos elementos de entrada**. Los cálculos van **hacia atrás** a lo largo del grafo.

Eso significa que no se sigue la línea vacía que comienza con $\partial y/\partial x = 0$ en el grafo hacia adelante para las derivadas con respecto a x. Y no se sigue la línea vacía $\partial x/\partial y = 0$ en el grafo hacia adelante (no dibujado) para las derivadas con respecto a y. Un problema más grande y realista con N elementos tendrá N grafos hacia adelante, cada uno con $N - 1$ lineas vacías (porque las N entradas son independientes). La derivada de x_i con respecto a cada otro elemento x_j es $\partial x_i/\partial x_j = 0$.

En lugar de N grafos hacia adelante a partir de N entradas, tendremos **un grafo hacia atrás a partir de un resultado**. He aquí ese grafo computacional en modo reverso. Encuentra la derivada de F con respecto a cada nodo. Comienza con $\partial F/\partial F = 1$ y luego va en la dirección inversa.

Un grafo computacional ejecuta la regla de la cadena para encontrar derivadas. El modo reverso encuentra todas las derivadas $\partial F/\partial x_i$ siguiendo todas las cadenas **hacia atrás, desde el resultado hacia la entrada.** Todas esas cadenas aparecen como caminos **en un grafo**, no como reglas de la cadena separadas para obtener un número exponencial de caminos posibles. Este es el éxito del modo reverso.

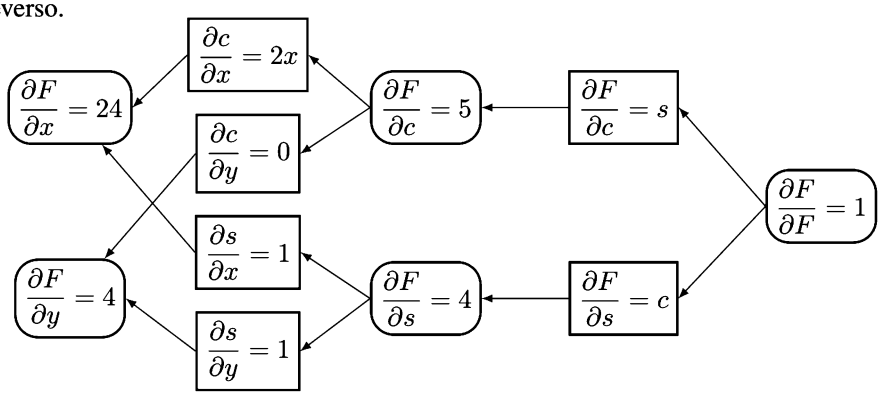


Figura VII.5: Cálculo con el modo reverso del gradiente $\left(\frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}\right)$ en $x = 2, y = 3$.

Métodos duales

La misma pregunta sobre cuál es el mejor orden para multiplicar matrices ABC aparece en una gran clase de problemas de optimización. Resolvemos un sistema cuadrado de N ecuaciones lineales $E\boldsymbol{v} = \boldsymbol{b}$. El vector $\boldsymbol{b}$ depende de las **variables de diseño** $\boldsymbol{p} = (p_1, \ldots, p_M)$. Luego el vector solución $\boldsymbol{v} = E^{-1}\boldsymbol{b}$ depende de $\boldsymbol{p}$. **La matriz $\partial \boldsymbol{v}/\partial \boldsymbol{p}$ que contiene las derivadas $\partial \boldsymbol{v}_i/\partial p_j$ será de $\boldsymbol{N}$ por $\boldsymbol{M}$.**

Reitero: minimizamos $F(\boldsymbol{v})$. El vector $\boldsymbol{v}$ depende de las variables de diseño $\boldsymbol{p}$. Entonces necesitamos una regla de la cadena que multiplique las derivadas $\partial F/\partial \boldsymbol{v}_i$ por las derivadas $\partial \boldsymbol{v}_i/\partial p_j$. Veamos cómo esto se vuelve un producto de **tres matrices**; el orden de la multiplicación es decisivo. **Tres conjuntos de derivadas** controlan cómo F depende de las variables de entrada p_i:

$$A = \boldsymbol{\partial F/\partial v_i} \quad \text{Las derivadas de } F \text{ con respecto a } v_1, \ldots, v_N \tag{8a}$$

$$B = \boldsymbol{\partial v_i/\partial b_k} \quad \text{La derivada de cada } v_i \text{ con respecto a cada } b_k \tag{8b}$$

$$C = \boldsymbol{\partial b_k/\partial p_j} \quad \text{La derivada de cada } b_k \text{ con respecto a cada } p_j \tag{8c}$$

Para hallar $\partial v_i/\partial p_j$ tomamos las derivadas de la ecuación $E\boldsymbol{v} = \boldsymbol{b}$ con respecto a p_j:

$$E\frac{\partial \boldsymbol{v}}{\partial p_j} = \frac{\partial \boldsymbol{b}}{\partial p_j}, \quad j = 1, \ldots, M \quad \text{entonces} \quad \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{p}} = E^{-1}\frac{\partial \boldsymbol{b}}{\partial \boldsymbol{p}}. \tag{9}$$

Parece que tenemos M sistemas lineales de tamaño N. Esos serán costosos de resolver una y otra vez, mientras buscamos $\boldsymbol{p}$ que minimice $F(\boldsymbol{v})$. La matriz $\partial \boldsymbol{v}/\partial \boldsymbol{p}$ contiene las derivadas de $v_1, \ldots, v_N$ con respecto a las variables de diseño $p_1, \ldots, p_M$.

Supongamos por ahora que la función costo $F(\boldsymbol{v}) = \boldsymbol{c}^{\mathbf{T}}\boldsymbol{v}$ es lineal (entonces $\partial F/\partial \boldsymbol{v} = \boldsymbol{c}^{\mathrm{T}}$). Por lo tanto, lo que la optimización de verdad necesita es el gradiente de $F(\boldsymbol{v})$ con respecto a los p. El primer conjunto de derivadas $\partial F/\partial \boldsymbol{v}$ es justamente el vector $\boldsymbol{c}^{\mathrm{T}}$:

$$\boxed{\frac{\partial F}{\partial \boldsymbol{p}} = \frac{\partial F}{\partial \boldsymbol{v}}\,\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{p}} = \boldsymbol{c}^{\mathbf{T}}\,E^{-1}\,\frac{\partial \boldsymbol{b}}{\partial \boldsymbol{p}} \quad \text{tiene tres factores que deben multiplicarse.}} \tag{10}$$

Esta es la pregunta clave, termina con un producto de un **vector fila** $\boldsymbol{c}^{\mathrm{T}}$ multiplicado por **una matriz** E^{-1} **de** $\boldsymbol{N}$ **por** $\boldsymbol{N}$ por una matriz $\partial \boldsymbol{b}/\partial \boldsymbol{p}$ de $\boldsymbol{N}$ **por** $\boldsymbol{M}$. ¿ Cómo deberíamos calcular ese producto ?

Una vez más la pregunta casi se responde por sí misma. *No* queremos multiplicar dos matrices así que *no* calculamos $\partial \boldsymbol{v}/\partial \boldsymbol{p}$ después de todo. En su lugar, el mejor primer paso es $\boldsymbol{c}^{\mathbf{T}}E^{-1}$. Esto produce un vector fila $\boldsymbol{\lambda}^{\mathbf{T}}$. En otras palabras, resolvemos la ecuación dual $E^{\mathrm{T}}\boldsymbol{\lambda} = \boldsymbol{c}$:

$$\boxed{\textbf{Ecuación dual} \qquad \boldsymbol{E}^{\mathbf{T}}\boldsymbol{\lambda} = \boldsymbol{c} \quad \text{da} \quad \boldsymbol{\lambda}^{\mathrm{T}}E = \boldsymbol{c}^{\mathrm{T}} \quad \text{y} \quad \boldsymbol{\lambda}^{\mathrm{T}} = \boldsymbol{c}^{\mathrm{T}}E^{-1}.} \tag{11}$$

Sustituyendo $\boldsymbol{\lambda}^{\mathbf{T}}$ por $\boldsymbol{c}^{\mathbf{T}}E^{-1}$ en la ecuación (10), el paso final multiplica ese vector fila por las derivadas del vector $\boldsymbol{b}$ (su gradiente) :

$$\boxed{\textbf{Gradiente del costo } \boldsymbol{F} \qquad \frac{\partial F}{\partial \boldsymbol{p}} = \boldsymbol{\lambda}^{\mathbf{T}}\,\frac{\partial \boldsymbol{b}}{\partial \boldsymbol{p}} \qquad (1 \text{ por } N \text{ multiplicado por } N \text{ por } M).} \tag{12}$$

El orden óptimo es $(AB)C$ porque el primer factor A es en realidad el vector fila $\boldsymbol{\lambda}^{\mathbf{T}}$.

Este ejemplo de un método dual comienza con $E\boldsymbol{x} = \boldsymbol{b}$. El lado derecho $\boldsymbol{b}$ depende de los parámetros de diseño $\boldsymbol{p}$. Luego la solución $\boldsymbol{x} = E^{-1}\boldsymbol{b}$ depende de $\boldsymbol{p}$. Entonces la función costo $F(\boldsymbol{x}) = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ depende de $\boldsymbol{p}$.

La ecuación dual $A^{\mathrm{T}}\boldsymbol{\lambda} = \boldsymbol{c}$ determina el vector $\boldsymbol{\lambda}$ que combina eficientemente los dos últimos pasos. "Dual" tiene un significado paralelo a "traspuesto" y también podemos aplicarlo a las ecuaciones diferenciales. Las variables de diseño $p_1, \ldots, p_M$ pueden aparecer en la matriz E, o en un problema de autovalores o en una ecuación diferencial.

Nuestro punto aquí es enfatizar y reforzar la idea clave de la retropropagación :

El modo reverso puede ordenar el cálculo de las derivadas de una manera más rápida.

Duales y sensibilidad en las capas profundas

Nos acercamos al problema del aprendizaje profundo, ¿ cuáles son las derivadas $\partial w/\partial x_j$ de las salidas $\boldsymbol{w} = (w_1, \ldots w_M)$ de la capa L con respecto a los parámetros $\boldsymbol{x} = (x_1, \ldots, x_N)$? Ese resultado $\boldsymbol{w} = \boldsymbol{v}_L$ se ve luego de L pasos a partir de la entrada $\boldsymbol{v}_0$. Escribimos el paso n de la siguiente forma

$$\boldsymbol{v}_n = F_n(\boldsymbol{v}_{n-1}, \boldsymbol{x}_n) \text{ donde } F_n \text{ depende de los pesos (parámetros) } \boldsymbol{x}_n. \tag{13}$$

(13) es una **relación de recurrencia**. Y los mismos P parámetros $\boldsymbol{x}$ podrían usarse en cada paso. El aprendizaje profundo tiene nuevos parámetros para cada capa nueva, lo que le da la "habilidad de aprender" a la que una relación de recurrencia ordinaria jamás podría aspirar. De hecho, una

recurrencia típica (13) es tan solo una diferencia finita, análoga a las de una ecuación diferencial $d\boldsymbol{v}/dt = f(\boldsymbol{v}, \boldsymbol{x}, t)$.

La analogía no es mala. En este caso también podemos apuntar hacia un resultado deseado $\boldsymbol{v}(T)$, y elegimos aquellos parámetros $\boldsymbol{x}$ que nos acerquen. El problema es encontrar la matriz de derivadas $J = \partial \boldsymbol{v}_N / \partial \boldsymbol{x}_M$. Debemos aplicar la regla de la cadena a la ecuación (13), retrocediendo de N a 0. He aquí un paso de la cadena:

$$v_N = F_N(v_{N-1}, x_N) = F_N(F_{N-1}(v_{N-2}, x_{N-1}), x_N). \tag{14}$$

Tomamos sus derivadas con respecto a x_{N-1}, para ver la regla en las últimas dos capas:

$$\frac{\partial v_N}{\partial x_{N-1}} = \frac{\partial F_N}{\partial v_{N-1}} \frac{\partial v_{N-1}}{\partial x_{N-1}} \qquad \frac{\partial v_N}{\partial x_{N-2}} = \frac{\partial F_N}{\partial v_{N-1}} \frac{\partial v_{N-1}}{\partial x_{N-2}} = \frac{\partial F_N}{\partial v_{N-1}} \frac{\partial v_{N-1}}{\partial v_{N-2}} \frac{\partial v_{N-2}}{\partial x_{N-2}}$$

Esa última expresión es un producto triple ABC. El cálculo requiere de una decisión: *¿ comenzar por AB o comenzar por BC ?* Ambos, el método dual para optimización y el modo reverso de la retropropagación, aconsejarían lo mismo: *comience por AB.*

Las últimas dos páginas fueron desarrolladas a partir de los apuntes de Steven Johnson: *Adjoint methods and sensitivity analysis for recurrence relations*, **http://math.mit.edu/∼stevenj/18.336/recurrence2.pdf**. También en línea: *Notes on adjoint methods for* 18.335.

Para el aprendizaje profundo, la relación de recurrencia está entre las capas de la red.

Problemas de la Sección VII.3

1 Si $\boldsymbol{x}$ e $\boldsymbol{y}$ son vectores columna en $\mathbf{R}^n$, ¿ es más rápido multiplicar $\boldsymbol{x}(\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x})$ o $(\boldsymbol{x}\boldsymbol{y}^{\mathrm{T}})\boldsymbol{x}$?

2 Si A es una matriz de m por n con $m > n$, ¿ es más rápido multiplicar $A(A^{\mathrm{T}}A)$ o $(AA^{\mathrm{T}})A$?

3 (a) Si $A\boldsymbol{x} = \boldsymbol{b}$ y A es fija, ¿ cuáles son las derivadas $\partial x_i / \partial b_j$?
(b) Y si $\boldsymbol{b}$ es fijo, ¿ Cuáles son las derivadas $\partial x_i / \partial A_{jk}$?

4 Para $\boldsymbol{x}$ e $\boldsymbol{y}$ en $\mathbf{R}^n$, ¿ qué son $\partial(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y})/\partial x_i$ y $\partial(\boldsymbol{x}\boldsymbol{y}^{\mathrm{T}})/\partial x_i$?

5 Dibuje un grafo computacional para calcular la función $f(x, y) = x^3(x - y)$. Use el grafo para calcular $f(2, 3)$.

6 Dibuje un grafo en modo reverso para calcular las derivadas $\partial f/\partial x$ y $\partial f/\partial y$ para $f = x^3(x - y)$. Use el grafo para encontrar esas derivadas en $x = 2$ e $y = 3$.

7 Supongamos que A es una matriz de Toeplitz en una red neuronal convolucional (**CNN**). El número a_k está sobre la diagonal k. Si $\boldsymbol{w} = A\boldsymbol{v}$, ¿ cuál es la derivada $\partial w_i / \partial a_k$?

8 En una capa de max-pooling, sea $w_i = \text{máx}\,(v_{2i-1}, v_{2i})$. Encuentre todos los $\partial w_i / \partial v_j$.

9 Para entender la regla de la cadena, comience a partir de esta identidad y tome $\Delta x \to 0$:

$$\frac{f(g(x + \Delta x)) - f(g(x))}{\Delta x} = \frac{f(g(x + \Delta x)) - f(g(x))}{g(x + \Delta x) - g(x)} \quad \frac{g(x + \Delta x) - g(x)}{\Delta x}$$

Consigna: encuentre la derivada en $x = 0$ de $\operatorname{sen}(\cos(\operatorname{sen} x))$.

VII.4. Hiperparámetros : las decisiones cruciales

Después de elegir la función de pérdida y de decidir la arquitectura de la red, todavía hay decisiones críticas por hacer. Debemos elegir los *hiperparámetros*. Estos rigen el comportamiento del algoritmo : el cálculo de los pesos. Esos pesos representan lo que la computadora aprendió del conjunto de entrenamiento : cómo predecir el resultado a partir de los predictores. En el aprendizaje automático, las decisiones incluyen esos hiperparámetros, la función de pérdida, el dropout y la regularización.

El objetivo es encontrar patrones que diferencien 5 de 7 y 2 a partir de la observación de píxeles. Los hiperparámetros determinan con qué velocidad y precisión se descubren esos patrones. El **tamaño del paso** s_k en el descenso por gradiente es lo primero y principal. Ese número que aparece en la iteración $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - s_k \boldsymbol{\nabla L}(\boldsymbol{x}_k)$ o una de sus variantes : *aceleradas* (por momento) o *adaptativas* (ADAM) o *estocásticas* con un minilote aleatorio de los datos de entrenamiento en cada paso k.

Las palabras **tasa de aprendizaje** se usan a menudo en lugar de tamaño del paso. Dependiendo del autor, los dos pueden ser idénticos o diferir por un factor normalizante. Además : η_k *a menudo reemplaza a* s_k. Primero buscamos el tamaño de paso óptimo cuando solo tenemos una incógnita. Luego buscamos un enfoque general. Eventualmente queremos la opción más rápida.

1. **Elegir $\boldsymbol{s_k = 1/L''(x_k)}$.** Newton usa la derivada segunda de L. Esa elección toma en cuenta el término cuadrático de la serie de Taylor de $L(x)$ alrededor del punto x_k. Como resultado, el método de Newton es de *segundo orden* : el error en x_{k+1} es proporcional al *cuadrado* del error en x_k. Cerca del x^* minimizante, la convergencia es rápida.

   En más dimensiones, la derivada segunda se vuelve la matriz hessiana $H(\boldsymbol{x}) = \boldsymbol{\nabla^2 L}(\boldsymbol{x}_k)$. Su tamaño es la cantidad de pesos (componentes de $\boldsymbol{x}$). Para encontrar $\boldsymbol{x}_{k+1}$, Newton resuelve un sistema grande de ecuaciones $H(\boldsymbol{x}_k)\,(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) = -\boldsymbol{\nabla L}(\boldsymbol{x}_k)$. El descenso por gradiente reemplaza H por un único número $1/s_k$.

2. **Decidir $\boldsymbol{s_k}$ a partir de una búsqueda lineal.** El gradiente $\boldsymbol{\nabla L}(\boldsymbol{x}_k)$ determina la dirección de la recta. El punto actual $\boldsymbol{x}_k$ es el comienzo de la recta. Al evaluar $L(\boldsymbol{x})$ en puntos sobre la recta, encontramos un punto casi minimizante, que se vuelve $\boldsymbol{x}_{k+1}$.

   La búsqueda lineal es una idea práctica. Un algoritmo es el que utiliza *retroceso* o *vuelta atrás (backtracking)*, como se describió en la Sección VI.4. Esto reduce el tamaño del paso s por un factor constante hasta que el decrecimiento en L sea consistente con la pendiente del gradiente (también dentro de un factor elegido). Optimizar una búsqueda lineal es un problema unidimensional cuidadosamente estudiado.

Pero ningún método es perfecto. Ahora vemos el efecto de una elección pobre de tamaño de paso s.

Demasiado pequeño o demasiado grande

Necesitamos identificar las dificultades que aparecen al hacer una mala elección de la tasa de aprendizaje :

$\boldsymbol{s_k}$ **demasiado pequeño**	Al descenso por gradiente le lleva mucho tiempo minimizar $L(\boldsymbol{x})$ Muchos pasos $\boldsymbol{x}_{k+1} - \boldsymbol{x}_k = -s_k \boldsymbol{\nabla L}(\boldsymbol{x}_k)$ con una mejora pequeña
$\boldsymbol{s_k}$ **demasiado grande**	Sobrepasamos la mejor elección de $\boldsymbol{x}_{k+1}$ en el descenso por gradiente El descenso por gradiente da saltos alrededor del $\boldsymbol{x}^*$ que minimiza

Supongamos que la búsqueda lineal encuentra los primeros pasos s_0 y s_1, y estos funcionan bien. Tal vez queremos mantener esa tasa de aprendizaje para las primeras iteraciones. **Normalmente reducimos** s a medida que continúa la minimización de $L(\boldsymbol{x})$.

Pasos grandes al inicio	Acercarnos rápidamente a un entorno del vector $\boldsymbol{x}^*$ de pesos óptimos
Pasos pequeños al final	Buscar convergencia sin sobrepasarse

Un cronograma de aprendizaje $\boldsymbol{s_k = s_0/\sqrt{k}}$ o $\boldsymbol{s_k = s_0/k}$ reduce los pasos sistemáticamente.

Luego de alcanzar un vector de pesos $\boldsymbol{x}$ que esté cerca de minimizar la función de pérdida $L(\boldsymbol{x}, \boldsymbol{v})$, tal vez queramos usar nuevos $\boldsymbol{v}$ de un **conjunto de validación**. El propósito de la **validación cruzada** es confirmar que los pesos calculados $\boldsymbol{x}$ son capaces de producir resultados precisos a partir de datos nuevos.

Validación cruzada

La validación cruzada busca estimar la validez de nuestro modelo y la fuerza de nuestra función de aprendizaje. ¿ El modelo es demasiado débil o simple para dar predicciones y clasificaciones precisas ? ¿ Estamos sobreajustando los datos de entrenamiento y por lo tanto arriesgándonos con nuevos datos de prueba ?

Observación Otro método estadístico (para otro propósito) también reutiliza los datos. Estamos hablando del *bootstrap* propuesto por Brad Efron. Se utiliza (y se necesita) cuando el tamaño de la muestra es pequeño o su distribución es desconocida. Buscamos máximo conocimiento al volver a esa muestra (pequeña) y *reusar esos datos* para obtener más información. Normalmente los conjuntos pequeños de datos no se dan en el contexto de las aplicaciones del aprendizaje profundo.

Un primer paso de la validación cruzada es dividir los datos disponibles en K subconjuntos. Si $K = 2$, estos serían esencialmente el conjunto de entrenamiento y el conjunto de prueba, pero usualmente buscamos obtener más información a partir de conjuntos pequeños antes de comenzar a trabajar con un conjunto de prueba grande. La *validación cruzada con K iteraciones* usa cada uno de los K subconjuntos por separado a modo de conjunto de prueba. En cada prueba los otros $K - 1$ subconjuntos forman el conjunto de entrenamiento. Retrabajamos los mismos datos (tamaño moderado) para aprender más que lo que nos puede enseñar una optimización.

La validación cruzada puede crear una tasa de aprendizaje adaptativa : cambia a medida que se desciende.

Hay muchas variantes, como la "validación cruzada doble". En un problema de mínimos cuadrados de m por n estandarizado $A\boldsymbol{x} = \boldsymbol{b}$, Wikipedia da el valor esperado $(m-n-1)/(m+n-1)$ para el error cuadrado medio. Errores más grandes normalmente indican un sobreajuste. La prueba correspondiente en el aprendizaje profundo nos advierte que consideremos detenernos antes.

Esta sección sobre hiperparámetros fue influenciada y mejorada por un capítulo largo en un libro fantástico de Bengio. Ese libro es *Neural Networks : Tricks of the Trade* (2da edición), editado por G. Montavon, G. Orr y K.-R. Müller. Fue publicado por Springer (2012) y contiene importantes contribuciones de referentes en el campo.

Normalización por lotes en cada capa

A medida que se avanza con el entrenamiento, la media y la varianza pueden cambiar en cada una de las capas de la red. Este cambio en la distribución de los predictores es un "desplazamiento covariado". A menudo tenemos que ajustar el tamaño del paso y otros hiperparámetros debido a este desplazamiento en los estadísticos de las capas. *Normalizar las entradas en cada capa* es un buen plan.

La normalización hace que el entrenamiento sea más seguro y más rápido. De esta manera, la necesidad del dropout suele desaparecer. Menos iteraciones pueden dar pesos más precisos. Y el costo puede ser moderado. *A menudo entrenamos solo dos parámetros adicionales en cada capa.*

El problema es mayor cuando la función no lineal es sigmoidea en lugar de ReLU. La sigmoidea "satura" pues se aproxima a un límite como 1 (mientras que ReLU siempre aumenta a medida que $x \to \infty$). La sigmoidea no lineal se vuelve virtualmente lineal y hasta constante cuando x es grande. El entrenamiento se retrasa porque apenas se usa la no linealidad.

Solo nos queda decidir el momento en el que los predictores serán normalizados. Ioffe y Szegedy evitan calcular matrices de covarianza (demasiado costoso). Su transformación normalizante actúa sobre cada predictor $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_B$ en un minilote de tamaño B :

media	$\boldsymbol{\mu} = (\boldsymbol{v}_1 + \cdots + \boldsymbol{v}_B)/B$
varianza	$\sigma^2 = \left(\|\|\boldsymbol{v}_1 - \boldsymbol{\mu}\|\|^2 + \cdots + \|\|\boldsymbol{v}_B - \boldsymbol{\mu}\|\|^2\right)/B$
normalizar	$\boldsymbol{V}_i = (\boldsymbol{v}_i - \boldsymbol{\mu})/\sqrt{\sigma^2 + \epsilon}$ para $\epsilon > 0$ pequeño
escalar/desplazar	$\boldsymbol{y}_i = \boldsymbol{\gamma}\boldsymbol{V}_i + \boldsymbol{\beta}$ ($\boldsymbol{\gamma}$ y $\boldsymbol{\beta}$ son parámetros entrenables)

El punto clave es **normalizar los elementos $\boldsymbol{y}_i$ para cada capa nueva**. Lo que resultó para el lote original de vectores (en la capa cero) también le sirve a los predictores en cada capa oculta.

S. Ioffe y C. Szegedy, *Batch normalization*, arXiv : 1502.03167v3, 2 de marzo de 2015.

Dropout

El dropout consiste en seleccionar aleatoriamente neuronas de la red y quitarlas. Esas son componentes de la capa entrante $\boldsymbol{v}_0$ o de las capas ocultas $\boldsymbol{v}_n$ antes de la capa resultante $\boldsymbol{v}_L$. Todos los pesos en las A y los $\boldsymbol{b}$ conectados a esas neuronas omitidas desaparecen de la red (Figura VII.6). Típicamente a las neuronas de capas ocultas se les da probabilidad $p = 0.5$ de supervivencia, mientras que las de entrada pueden tener $p = 0.8$ o más. *El objetivo principal del dropout aleatorio es evitar el sobreajuste.* Es un método para promediar relativamente económico en comparación con combinar predicciones de muchas redes.

Cinco líderes en el desarrollo de algoritmos de aprendizaje profundo propusieron el dropout : N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever y R. Salakhutdinov. Su artículo "*Dropout*" puede leerse en : *Journal of Machine Learning Research* **15** (2014) 1929-1958. Para asociaciones recientes de dropout a la física y la incertidumbre vea arXiv : 1506.02142 y 1809.08327.

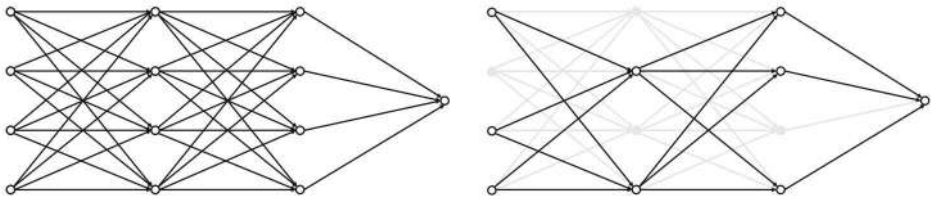


Figura VII.6: neuronas cruzadas han sido eliminadas en la red simplificada.

El dropout ofrece una manera de trabajar con muchas arquitecturas neuronales diferentes a la vez. En el entrenamiento, $\boldsymbol{v}_0$ (el vector predictor de una muestra ingresada) lleva a una nueva red simplificada. Comenzando con N neuronas tenemos 2^N redes simplificadas posibles.

Al momento de probar la red, usamos la red completa (sin dropout) con pesos reescalados de los pesos de entrenamiento. Los pesos salientes de una neurona no descartada se multiplican por p al reescalar. Este promedio aproximado, al momento de probar la red, llevó a los cinco autores a menores errores de generalización de una manera más simple que usando otros métodos de regularización.

Una inspiración para el dropout fue la reproducción genética, donde la mitad de los genes de cada padre se pierden y hay una pequeña mutación aleatoria. Ese dropout para un niño es más imperdonable y permanente que el dropout en el aprendizaje profundo, que promedia muchas redes simplificadas. (Es cierto, vemos un promedio en el caso de los hermanos, pero los autores especulan que con el tiempo nuestros genes deben volverse más fuertes para sobrevivir.)

El modelo del dropout usa una variable cero-uno r (una variable Bernoulli). Así $r = 1$ con probabilidad p y $r = 0$ con probabilidad $1 - p$. El paso hacia adelante usual para la capa n es $\boldsymbol{y}_n = A_n \boldsymbol{v}_{n-1} + \boldsymbol{b}_n$, seguido del $\boldsymbol{v}_n = \mathrm{R}\boldsymbol{y}_n$ no lineal. *Ahora una r aleatoria multiplica cada componente de $\boldsymbol{v}_{n-1}$* para descartar esa neurona cuando $r = 0$. Componente por componente, $\boldsymbol{v}_{n-1}$ se multiplica por 0 o 1 para dar lugar a $\boldsymbol{v}^*_{n-1}$. Así $\boldsymbol{y}_n = A_n \boldsymbol{v}^*_{n-1} + \boldsymbol{b}_n$.

Al calcular gradientes, usamos retropropagación para cada muestra de entrenamiento que pertenece al minilote. Luego promediamos esos gradientes. El descenso por gradiente estocástico todavía puede incluir la aceleración (con momento agregado), el descenso adaptativo y el decaimiento de los pesos. Los autores recomiendan fuertemente regularizar los pesos, por ejemplo, con un requisito de la norma máxima $||\boldsymbol{a}|| \leq c$ en las columnas de todas las matrices de pesos A.

Explorando el espacio de los hiperparámetros

A menudo optimizamos los hiperparámetros usando experimentos o la experiencia. Para determinar la tasa de aprendizaje, podemos intentar con tres posibilidades y medir la caída en la función de pérdida para cada una. Una sucesión geométrica como $.1, .01, .001$ tendría más sentido que una sucesión aritmética $.05, .03, .01$. Y si la opción más grande o más pequeña da los mejores resultados, entonces continuamos el experimento con el siguiente número en la serie. En el ejemplo del tamaño del paso, consideraríamos el costo del cálculo además del error de validación.

LeCun enfatiza que para una búsqueda multiparamétrica, el *muestreo aleatorio* es la manera de cubrir rápidamente muchas posibilidades. La búsqueda por cuadrícula es demasiado lenta en dimensiones altas.

Funciones de pérdida

La función de pérdida mide la diferencia, en cada muestra, entre el resultado correcto y el resultado calculado. El resultado correcto (a menudo una clasificación $y = 0, 1$ o $y = 1, 2, \ldots, n$) es parte de los datos de entrenamiento. El resultado calculado en la capa final es $\boldsymbol{w} = F(\boldsymbol{x}, \boldsymbol{v})$ de la función de aprendizaje con pesos $\boldsymbol{x}$ y predictor $\boldsymbol{v}$.

En la Sección VI.5 se definieron tres funciones de pérdida familiares. Luego este capítulo se enfocó en la estructura de la red neuronal y la función F. Aquí volvemos a comparar la pérdida cuadrática con la pérdida de entropía cruzada.

1. Costo cuadrático (Pérdida cuadrática): $\boldsymbol{\ell(y, w) = \frac{1}{2}||y - w||^2}$.

Esta es la función de pérdida de mínimos cuadrados, que resulta siempre una opción posible. Pero no es la opción preferida para el aprendizaje profundo. Una razón es la forma parabólica del gráfico de $\ell(y, w)$, a medida que la pérdida se aproxima a cero en $w = y$. La derivada también se aproxima a cero.

Una derivada cero en el mínimo es normal para una función de pérdida suave, pero a menudo lleva a resultados indeseados : *los pesos A y $\boldsymbol{b}$ cambian muy lentamente cerca del punto óptimo*. El aprendizaje se ralentiza y se necesitan muchas iteraciones.

2. Pérdida de entropía cruzada : $$\boldsymbol{\ell(y, w)} = -\frac{1}{n}\sum_{1}^{n}[y_i \log z_i + (1 - y_i)\log(1 - z_i)] \quad (1)$$

Aquí permitimos y esperamos que los N resultados w_i del entrenamiento de la red neuronal hayan sido normalizados a $z(w)$, con $0 < z_i < 1$. A menudo, esos z_i son probabilidades. Así $1 - z_i$ también está entre 0 y 1. Entonces ambos logaritmos en (1) son negativos, y el signo menos nos asegura que la pérdida general es positiva : $\ell > 0$.

Asimismo, los logaritmos dan un enfoque diferente y deseable para $z = 0$ o 1. Para este cálculo nos referimos al libro en versión digital de Nielsen *Neural Networks and Deep Learning*, que se enfoca en las funciones de activación sigmoideas en lugar de ReLU. El precio de esas funciones suaves es que *saturan* (pierden su no linealidad) cerca de los puntos extremos.

La entropía cruzada tiene propiedades buenas, pero ¿ de dónde salen los logaritmos ? El primer punto es la fórmula para la entropía (una medida de información) de Shannon. Si el mensaje i tiene probabilidad p_i, deberíamos permitir $-\log p_i$ bits para ese mensaje. Así la cantidad de bits (promedio) esperada por mensaje es la mejor posible :

$$\boxed{\textbf{Entropía} = -\sum_{1}^{m} p_i \log p_i. \text{ Para } m = 2 \text{ esto es } -p\log p - (1-p)\log(1-p).} \quad (2)$$

Se usa la entropía cruzada cuando no conocemos p_i y usamos $\widehat{p}_i$ en su lugar :

$$\boxed{\textbf{Entropía cruzada} = -\sum_{1}^{m} p_i \log \widehat{p}_i. \text{ Para } m = 2 \text{ esto es } -p\log \widehat{p} - (1-p)\log(1-\widehat{p}).} \quad (3)$$

(3) siempre es mayor que (2). El verdadero p_i es desconocido y los $\widehat{p}_i$ son más costosos. La diferencia es una función muy útil pero no simétrica llamada **divergencia de Kullback-Leibler (KL)**.

Regularización : ℓ^2 o ℓ^1 (o ninguna)

La regularización es una decisión voluntaria pero recomendada. Agrega un *término de penalización* a la función de pérdida $L(\boldsymbol{x})$ a minimizar : una penalización $\boldsymbol{\ell^2}$ en la regresión ridge y $\boldsymbol{\ell^1}$ en LASSO.

RR Minimiza $||\boldsymbol{b} - A\boldsymbol{x}||_2^2 + \lambda_2 ||\boldsymbol{x}||_2^2$ **LASSO** Minimiza $||\boldsymbol{b} - A\boldsymbol{x}||_2^2 + \lambda_1 \sum |x_i|$

La penalización controla el tamaño de $\boldsymbol{x}$. *La regularización produce un decaimiento de los pesos.*

El coeficiente λ_2 o λ_1 es un hiperparámetro. Su valor se puede calcular por validación cruzada. El propósito de la penalización es evitar el sobreajuste (a veces expresado como *ajustar el ruido*). La validación cruzada para un λ dado encuentra el $\boldsymbol{x}$ minimizante en un conjunto de entrenamiento. Luego se usan esos pesos en un conjunto de prueba. Si se ve un sobreajuste, aumenta λ.

Un pequeño valor de λ tiende a aumentar la varianza del error : sobreajuste. λ grande aumenta el sesgo : subajuste porque el término de ajuste $||\boldsymbol{b} - A\boldsymbol{x}||^2$ es menos importante.

¡ **Un punto de vista diferente** ! Experimentos recientes en MNIST no aclaran si una regularización explícita es siempre necesaria. El mejor rendimiento en el conjunto de prueba a menudo se ve con $\lambda = 0$ (así $\boldsymbol{x^*}$ es la solución de norma mínima $A^+\boldsymbol{b}$). El análisis de Liang y Rakhlin identifica matrices para las que este buen resultado puede esperarse, siempre y cuando los datos lleven a un decaimiento rápido del espectro de la matriz de covarianza muestral y de la matriz núcleo.

En muchos casos estas son las matrices de la Sección III.3 : *efectivamente de rango bajo*. Cada vez se escuchan más ideas similares, donde el aprendizaje profundo con muchos pesos adicionales y buenos hiperparámetros encontarán soluciones que generalizan, sin penalización.

T. Liang y A. Rakhlin, *Just interpolate : Kernel "ridgeless" regression can generalize*, disponible en arXiv : 1808.00387, 1 de agosto de 2018.

La estructura de AlphaGo Zero

Es interesante ver la secuencia de operaciones en AlphaGo Zero, mientras aprende a jugar al Go :

1. Una convolución de 256 filtros con núcleo de tamaño 3×3 y paso $1 : E = 3, S = 1$
2. Normalización por lotes
3. ReLU
4. Una convolución de 256 filtros de núcleo de tamaño 3×3 y paso 1
5. Normalización de lotes
6. Una conexión salteada como en ResNets que suma el predictor al bloque
7. ReLU
8. Una capa lineal completamente conectada a una capa oculta de tamaño 256
9. ReLU

El entrenamiento fue con descenso por gradiente estocástico en un conjunto de datos fijo, que contenía los últimos 2 millones de partidas jugadas consigo mismo de una versión anterior de AlphaGo Zero.

La CNN incluye una capa completamente conectada que produce un vector salida de tamaño $19^2 + 1$. Esto toma en consideración todas las posiciones en el tablero de 19×19, y la posibilidad de pasar un turno en el Go.

VII.5. El mundo del aprendizaje automático

Las redes completamente conectadas y las redes convolucionales son partes de un mundo más grande. Los datos de entrenamiento llevan a una función de aprendizaje $\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{v})$. Esa función produce una aproximación cercana al resultado correcto $\boldsymbol{w}$ para cada entrada $\boldsymbol{v}$ ($\boldsymbol{v}$ es el vector de predictores de esa muestra). Pero el aprendizaje automático desarrolló un abanico de otros enfoques, algunos utilizados desde hace mucho tiempo, para el problema de aprender a partir de datos.

Este libro no puede hacer justicia a todas esas ideas. Parece importante describir las redes neuronales recurrentes (y las máquinas de vectores de soporte). También se incluyen palabras clave para definir el alcance del aprendizaje profundo. (¡Necesitamos un *glosario* urgentemente! Esto sería una gran contribución a este campo.) Al final podrá encontrar una lista de libros sobre diferentes temas dentro del aprendizaje automático.

Redes neuronales recurrentes (RNN)

Estas redes son apropiadas para datos que vienen en un orden definido. Esto incluye series de tiempo y lenguaje natural : *habla, textos o manuscritos*. En la red de conexiones desde la entrada $\boldsymbol{v}$ a la salida $\boldsymbol{w}$, *el tiempo anterior* $t-1$ *aporta el nuevo predictor*. Esta entrada recurrente la determina la función $h(t-1)$.

La Figura VII.7 muestra un bosquejo del nuevo paso en la arquitectura de la red.

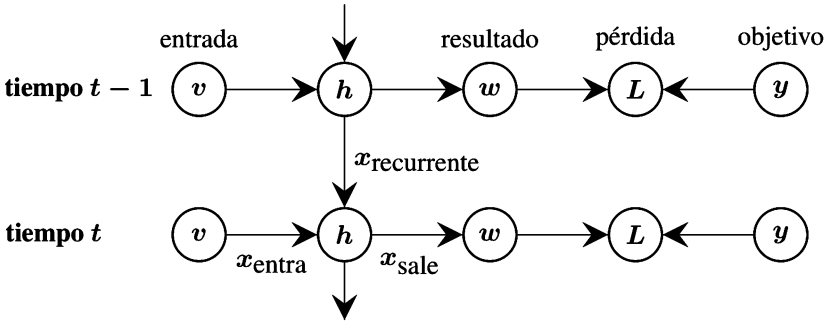


Figura VII.7: el grafo computacional para una red recurrente encuentra resultados que minimizan la pérdida $\boldsymbol{w}$ en cada t. Los elementos de entrada a $\boldsymbol{h}(t)$ son los nuevos datos $\boldsymbol{v}(t)$ y los datos recurrentes $\boldsymbol{h}(t-1)$ de la instancia anterior. Los pesos que multiplican los datos son $\boldsymbol{x}_{\text{entrada}}$, $\boldsymbol{x}_{\text{recurrente}}$ y $\boldsymbol{x}_{\text{salida}}$, elegidos para minimizar la pérdida $L(\boldsymbol{y}-\boldsymbol{w})$. Esta arquitectura de la red es *universal* : calculará cualquier fórmula que se pueda calcular con una máquina de Turing.

Palabras clave e ideas

1 Aprendizaje vía núcleos (siguiente página)

2 Máquinas de vectores de soporte (siguiente página)

3 Redes generativas adversarias

4 Análisis de componentes independientes

5 Modelos gráficos

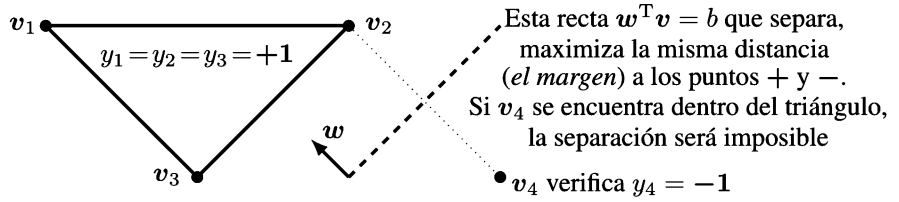
6 Estadísticas bayesianas

7 Bosques aleatorios

8 Aprendizaje por refuerzo

Máquinas de vectores de soporte

Comencemos con n puntos $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ en un espacio m dimensional. Cada $\boldsymbol{v}_i$ tiene una *clasificación* $y_i = 1$ *o* $y_i = -1$. El objetivo propuesto por Vapnik es encontrar un plano $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{v} = b$ en m dimensiones que *separe los puntos positivos de los negativos*, en caso de ser posible. Ese vector $\boldsymbol{w}$ será perpendicular al plano. El número b nos dice la distancia $|b|/||\boldsymbol{w}||$ desde la recta o el plano o el hiperplano en $\mathbf{R}^m$ al punto $(0, \ldots, 0)$.



Problema Encuentre $\boldsymbol{w}$ y b tal que $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{v}_i - b$ tenga el signo correcto y_i para todo $i = 1, \ldots, n$.

Si $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3$ son puntos positivos ($y = +1$) en un plano, entonces $\boldsymbol{v}_4$ *debe estar por fuera del triángulo de* $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3$. La imagen muestra la recta de separación máxima (*margen máximo*).

Margen máximo Minimizar $||\boldsymbol{w}||$ bajo las condiciones $y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{v}_i - b) \geq 1$.

Este es un "margen duro". Esa desigualdad requiere que $\boldsymbol{v}_i$ esté en el lado correcto del separador. Si los puntos no pueden separarse, entonces ningún $\boldsymbol{w}$ o b tendrá éxito. Para un "margen blando" podemos elegir los mejores $\boldsymbol{w}$ y b disponibles, resolviendo la siguiente minimización:

$$\textbf{Margen blando} \quad \text{Minimizar} \left[\frac{1}{n}\sum_{1}^{n} \text{máx}\,(0, 1 - y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{v}_i - \boldsymbol{b}))\right] + \lambda\,||\boldsymbol{w}||^2. \qquad (1)$$

Esa pérdida bisagra (el término máximo) es cero cuando $\boldsymbol{v}_i$ está del lado correcto del separador. Si la separación es imposible, la penalización $\lambda||\boldsymbol{w}||^2$ equilibra pérdidas bisagra con tamaños de márgenes.

Si introducimos una variable h_i para la pérdida bisagra, minimizamos una función cuadrática de $\boldsymbol{w}$ con desigualdades lineales que conectan $\boldsymbol{w}, b, y_i$ y h_i. Esto es programación cuadrática en dimensiones altas, que se conoce bien en teoría pero que es desafiante desde el punto de vista práctico.

El truco del núcleo

Las SVM son separaciones lineales. Un plano separa los puntos + de los puntos −. El truco del núcleo permite un separador no lineal cuando los vectores predictores $\boldsymbol{v}$ se transforman en $N(\boldsymbol{v})$. Así el producto punto de los vectores transformados dan la **función núcleo** $\boldsymbol{K}(\boldsymbol{v}_i, \boldsymbol{v}_j) = N(\boldsymbol{v}_i)^{\mathrm{T}}N(\boldsymbol{v}_j)$.

La clave es trabajar solamente con K sin usar la función N. *De hecho, nunca vemos o necesitamos* N. En el caso lineal, esto corresponde a elegir una K definida positiva sin ver la matriz A en $K = A^{\mathrm{T}}A$. El núcleo RBF $\exp(-||\boldsymbol{v}_i - \boldsymbol{v}_j||^2/2\sigma^2)$ está en III.3.

M. Belkin, S. Ma y S. Mandal, *To understand deep learning we need to understand kernel learning*, arXiv:1802.01396. "Non-smooth Laplacian kernels defeat smooth Gaussians".

T. Hofmann, B. Schölkopf y A. J. Smola, *Kernel methods in machine learning*, Annals of Statistics **36** (2008) 1171-1220 (con referencias exhaustivas).

El Traductor de Google

Un artículo excepcional sobre aprendizaje profundo y el desarrollo del Traductor de Google se publicó en la revista del New York Times el domingo 14 de diciembre de 2016. Cuenta cómo Google saltó de repente de una traducción convencional a una red neuronal recurrente. El autor Gideon Lewis-Kraus describe ese evento como tres historias en una : el trabajo del equipo de desarrollo, el grupo dentro de Google que vio que era posible y la comunidad internacional de científicos que gradualmente cambió nuestra comprensión sobre cómo aprender :

https://www.nytimes.com/2016/12/14/magazine/the-great-ai-awakening.html

El desarrollo llevó menos de un año. El equipo de Google Brain y sus competidores concibieron la idea en cinco años. La verdadera historia del aprendizaje automático es un orden de magnitud mayor en tiempo y espacio. El punto clave sobre esta historia reciente es el terremoto que provocó al ser utilizado para las traducciones automáticas :

> En lugar de programar cada palabra y regla gramatical con sus excepciones en ambos lenguajes, *se deja que la computadora descubra las reglas*. Solamente hay que darle suficientes traducciones correctas.
>
> Si estuviéramos reconociendo imágenes, los datos serían muchos ejemplos etiquetados correctamente (el conjunto de entrenamiento). La máquina crea la función $F(\boldsymbol{x}, \boldsymbol{v})$.

Esto se acerca a la manera en que aprenden los niños y cómo aprendemos nosotros. Si queremos enseñar a jugar a las damas o al ajedrez, la mejor manera es conseguir un tablero y jugar. Jugar el juego.

Llevar los pasos de esta visión a las redes neuronales y al aprendizaje profundo no fue una tarea fácil. Marvin Minsky sin duda fue uno de los líderes, pero su libro con Seymour Papert trata parcialmente sobre lo que los "perceptrones" no podían hacer. Con solo una capa, la función XOR (A o B pero no ambas) no estaba disponible. Faltaba la profundidad necesaria.

La obra de toda la vida de Geoffrey Hinton hizo una diferencia enorme en este tema. Para la traducción por computadora, resulta que él estaba en Google en el momento indicado. Para el reconocimiento de imágenes, ganó junto con sus estudiantes el desafío de reconocimiento visual en 2012 (con AlexNet). Esto cambió el diseño de las redes neuronales. Igual de impresionante fue un artículo de 1986 en *Nature*, donde Rumelhart, Hinton y Williams previeron que la retropropagación sería crucial para la optimización de los pesos : *aprender representaciones a partir de la retropropagación de errores*.

Estas ideas condujeron a importantes trabajos en todo el mundo. El "artículo sobre los gatos" de 2011-2012 describe el entrenamiento de un detector de caras sin imágenes etiquetadas. El autor principal de ese trabajo fue Quoc Le : *Building high-level features using large scale unsupervised learning* : **arxiv.org/abs/1112.6209**. Se tomaron muestras de YouTube para formar un gran conjunto de datos de imágenes de 200 por 200. La red tenía mil millones de pesos para entrenar, un volumen un millón de veces menor a la cantidad de neuronas en nuestra corteza visual. Le recomiendo leer este artículo para percibir la llegada del aprendizaje profundo.

Un pequeño equipo estaba silenciosamente superando al equipo que usaba reglas. Eventualmente se publicó en **arxiv.org/abs/1609.08144** el artículo con 31 autores. Y Google tuvo que cambiar a una red neuronal que no estuviera basada en reglas.

Libros sobre aprendizaje automático

1 Y. S. Abu-Mostafa *et al, Learning from Data*, AMLBook (2012).

2 C. C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*, Springer (2018).

3 E. Alpaydim, *Introduction to Machine Learning*, MIT Press (2016).

4 E. Alpaydim, *Machine Learning : The New AI*, MIT Press (2016).

5 C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer (2006).

6 F. Chollet, *Deep Learning with Python*, Manning (2017).

7 F. Chollet, *Deep Learning with R*, Manning (2018).

8 B. Efron y T. Hastie, *Computer Age Statistical Inference*, Cambridge (2016). https://web.stanford.edu/~hastie/CASI_files/PDF/casi.pdf

9 A. Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow*, O'Reilly (2017).

10 I. Goodfellow, Y. Bengio y A. Courville, *Deep Learning*, MIT Press (2016).

11 T. Hastie, R. Tibshirani y J. Friedman, *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*, Springer (2011).

12 M. Mahoney, J. Duchi y A. Gilbert, editores, *The Mathematics of Data*, American Mathematical Society (2018).

13 M. Minsky y S. Papert, *Perceptrons*, MIT Press (1969).

14 A. Moitra, *Algorithmic Aspects of Machine Learning*, Cambridge (2014).

15 G. Montavon, G. Orr y K. R. Müller, editores, *Neural Networks : Tricks of the Trade*, 2da edición, Springer (2012).

16 M. Nielsen, *Neural Networks and Deep Learning*, (title).com (2017).

17 B. Recht y S. Wright, *Optimization for Machine Learning*, próximamente.

18 A. Rosebrock, *Deep Learning for Computer Vision with Python*, pyimagsearch (2018).

19 S. Shalev-Schwartz y S. Ben-David, *Understanding Machine Learning : From Theory to Algorithms*, Cambridge (2014).

20 S. Sra, S. Nowozin y S. Wright, editores. *Optimization for Machine Learning*, MIT Press (2012).

21 G. Strang, *Linear Algebra and Learning from Data*, Wellesley-Cambridge Press (2019).

22 V. N. Vapnik, *Statistical Learning Theory*, Wiley (1998).

Autovalores y valores singulares : rango uno

Una matriz de rango uno tiene la sencilla forma $A = \boldsymbol{x}\boldsymbol{y}^{\mathrm{T}}$. Sus vectores singulares $\boldsymbol{u}_1, \boldsymbol{v}_1$ y su único valor singular no nulo σ_1 son muy fáciles de encontrar :

$$\boldsymbol{u}_1 = \frac{\boldsymbol{x}}{||\boldsymbol{x}||} \qquad \boldsymbol{v}_1 = \frac{\boldsymbol{y}}{||\boldsymbol{y}||} \qquad \sigma_1 = ||\boldsymbol{x}||\,||\boldsymbol{y}||.$$

Inmediatamente notamos que $A = \boldsymbol{x}\boldsymbol{y}^{\mathrm{T}} = \boldsymbol{u}_1\sigma_1\boldsymbol{v}_1^{\mathrm{T}}$. Un solo número no nulo en la matriz Σ de $m \times n$.
Todas las otras columnas de las matrices ortogonales U y V son perpendiculares a $\boldsymbol{u}_1$ y $\boldsymbol{v}_1$. La descomposición $A = U\Sigma V^{\mathrm{T}}$ se reduce al primer término $A = \boldsymbol{u}_1\sigma_1\boldsymbol{v}_1^{\mathrm{T}}$ porque el rango es igual a 1.

Los autovalores y los autovectores no son tan fáciles. Por supuesto la matriz A debe ser cuadrada. Para simplificar nuestras vidas continuamos con una matriz $A = \boldsymbol{x}\boldsymbol{y}^{\mathrm{T}}$ de 2 por 2. ¡ Sin dudas $\boldsymbol{x}$ es un autovector !

$$A\boldsymbol{x} = \boldsymbol{x}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x} = \lambda_1\boldsymbol{x} \;\text{ donde }\; \lambda_1 \;\text{ es el número }\; \boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}. \tag{2}$$

El otro autovalor es $\lambda_2 = 0$ porque A es singular (rango $= 1$). El autovector $\boldsymbol{x}_2 = \boldsymbol{y}^{\perp}$ debe ser perpendicular a $\boldsymbol{y}$, para que $A\boldsymbol{x}_2 = \boldsymbol{x}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}^{\perp} = \boldsymbol{0}$. Si $\boldsymbol{y} = (a, b)$ entonces $\boldsymbol{y}^{\perp}$ es su rotación de 90° $(b, -a)$.

La matriz traspuesta $A^{\mathrm{T}} = \boldsymbol{y}\boldsymbol{x}^{\mathrm{T}}$ tiene los mismos autovalores $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}$ y 0. Sus autovectores son los "autovectores a izquierda" de A. Serán $\boldsymbol{y}$ y $\boldsymbol{x}^{\perp}$ (porque $\boldsymbol{x}\boldsymbol{y}^{\mathrm{T}}$ tiene autovectores $\boldsymbol{x}$ e $\boldsymbol{y}^{\perp}$). La única pregunta es el escalamiento que decide las longitudes de los autovectores.

El requisito es (autovector a izquierda)$^{\mathrm{T}}$(autovector a derecha) $= 1$. Así los autovectores a izquierda son las filas de X^{-1} cuando los autovectores a derecha son las columnas de X : ¡ *perfecto* ! En nuestro caso, esos productos punto de los autovectores ahora se encuentran en $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}$ y también en $(\boldsymbol{x}^{\perp})^{\mathrm{T}}\boldsymbol{y}^{\perp}$. Dividamos ambos autovectores a izquierda $\boldsymbol{y}$ y $\boldsymbol{x}^{\perp}$ por el número $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}$, para producir $X^{-1}X = XX^{-1} = I$:

$$\boxed{\boldsymbol{x}\boldsymbol{y}^{\mathrm{T}} = \boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1} = \begin{bmatrix} \boldsymbol{x} & \boldsymbol{y}^{\perp} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}^{\mathrm{T}}\boldsymbol{x} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{x}^{\perp} \end{bmatrix} \Big/ \boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}} \tag{3}$$

Finalmente existe una última posibilidad crucial, que $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{0}$. Ahora los autovalores de $A = \boldsymbol{x}\boldsymbol{y}^{\mathrm{T}}$ son **cero y cero**. A solo tiene una recta de autovectores porque $\boldsymbol{y}^{\perp}$ está en la misma dirección que $\boldsymbol{x}$. La diagonalización (3) ya no es válida porque la matriz de autovectores X es *singular*. No podemos dividir por su determinante $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{0}$.

Esto muestra cómo los autovectores pueden entrar en una espiral de la muerte (o un abrazo fatal $\boldsymbol{x} = \boldsymbol{y}^{\perp}$). Por supuesto, los pares de vectores singulares $\boldsymbol{x}, \boldsymbol{x}^{\perp}$ e $\boldsymbol{y}, \boldsymbol{y}^{\perp}$ se mantienen ortogonales.

Pregunta En la ecuación (2), verifique que $\begin{bmatrix} \boldsymbol{y}^{\mathrm{T}} \\ \boldsymbol{x}^{\perp\mathrm{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} & \boldsymbol{y}^{\perp} \end{bmatrix} = (\boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

Pregunta ¿ Cuándo tiene $A = \boldsymbol{x}\boldsymbol{y}^{\mathrm{T}}$ autovectores ortogonales ?

Códigos y algoritmos de álgebra lineal numérica

LAPACK	es la primera opción para códigos de álgebra lineal para matrices densas.
ScaLAPACK	alcanza máximo rendimiento en problemas muy grandes.
COIN/OR	ofrece códigos de alta calidad para problemas de optimización de investigación operativa.

He aquí algunas fuentes para algoritmos específicos.

Solución directa de sistemas lineales	
Operaciones básicas de matriz-vector	BLAS
Eliminación con intercambio de filas	LAPACK
Solucionadores directos para sistemas ralos (UMFPACK)	SuiteSparse, SuperLU
QR por Gram-Schmidt y Householder	LAPACK
Autovalores y valores singulares	
Método QR desplazado para autovalores	LAPACK
Método de Golub-Kahan para SVD	LAPACK
Soluciones iterativas	
Gradientes conjugados precondicionados para $S\boldsymbol{x} = \boldsymbol{b}$	Trilinos
GMRES precondicionados para $A\boldsymbol{x} = \boldsymbol{b}$	Trilinos
Krylov-Arnoldi para $A\boldsymbol{x} = \lambda x$	ARPACK, Trilinos, SLEPc
Autovalores extremos de S	vea también BLOPEX
Optimización	
Programación lineal	CLP en COIN/OR
Programación semidefinida	CSDP en COIN/OR
Métodos de punto interior	IPOPT en COIN/OR
Optimización convexa	CVX, CVXR
Álgebra lineal aleatoria	
Factorizaciones aleatorias mediante QR con pivoteo	users.ices.utexas.edu/
$A = CMR$ columnas/mezcla/filas	~pgm/main_codes.html
Descomposición interpolatoria (ID)	
Transformada rápida de Fourier	FFTW.org
Repositorio de códigos de alta calidad	GAMS y Netlib.org
ACM Transactions on Mathematical Software	TOMS
Software de aprendizaje profundo (vea también la página 374)	
Aprendizaje profundo en Julia	Fluxml.ai/Flux.jl/stable
Aprendizaje profundo en MATLAB	Mathworks.com/learn/tutorials/deep–learning–onramp.html
Aprendizaje profundo en Python y JavaScript	Tensorflow.org, Tensorflow.js
Aprendizaje profundo en R	Keras, KerasR

Contar parámetros en factorizaciones básicas

$$\boldsymbol{A = LU} \quad \boldsymbol{A = QR} \quad \boldsymbol{S = Q\Lambda Q^{\mathrm{T}}} \quad \boldsymbol{A = X\Lambda X^{-1}} \quad \boldsymbol{A = QS} \quad \boldsymbol{A = U\Sigma V^{\mathrm{T}}}$$

Este es un repaso de ideas claves en álgebra lineal. Las ideas son expresadas por esas factorizaciones y nuestro plan es sencillo : *contar los parámetros en cada matriz.* Esperamos ver que en cada ecuación como $A = LU$, ambos lados tengan la misma cantidad de parámetros.

Para $A = LU$, ambos lados tienen n^2 parámetros.

$\boldsymbol{L}$: Matriz de $n \times n$ **triangular** con unos en la diagonal	$\frac{1}{2}\,\boldsymbol{n(n-1)}$
$\boldsymbol{U}$: Matriz de $n \times n$ **triangular** con diagonal libre	$\frac{1}{2}\,\boldsymbol{n(n+1)}$
$\boldsymbol{Q}$: Matriz de $n \times n$ **ortogonal**	$\frac{1}{2}\,\boldsymbol{n(n-1)}$
$\boldsymbol{S}$: Matriz de $n \times n$ **simétrica**	$\frac{1}{2}\,\boldsymbol{n(n+1)}$
$\boldsymbol{\Lambda}$: Matriz de $n \times n$ **diagonal**	$\boldsymbol{n}$
$\boldsymbol{X}$: Matriz de $n \times n$ de **autovectores** independientes	$\boldsymbol{n^2 - n}$

Algunos comentarios sobre Q: su primera columna $\boldsymbol{q}_1$ es un punto en una esfera unitaria en $\mathbf{R}^n$. Esa esfera es una superficie $\boldsymbol{n-1}$-dimensional, en el mismo sentido que el círculo unitario $x^2+y^2=1$ en $\mathbf{R}^2$ tiene solo un parámetro (el ángulo θ). El requisito $||\boldsymbol{q}_1|| = 1$ usó uno de los n parámetros en $\boldsymbol{q}_1$. Así $\boldsymbol{q}_2$ tiene $n-2$ parámetros: es un vector unitario y es ortogonal a $\boldsymbol{q}_1$. La suma $(n-1)+(n-2)+\cdots+1$ es igual a $\frac{1}{2}\,\boldsymbol{n(n-1)}$ parámetros libres en Q.

La matriz de autovectores X tiene solo n^2-n parámetros, no n^2. Si $\boldsymbol{x}$ es un autovector, entonces también lo es $c\boldsymbol{x}$ para todo $c \neq 0$. Podríamos requerir que la mayor componente de cada $\boldsymbol{x}$ sea 1. Esto nos deja $n-1$ parámetros para cada autovector (y ningún parámetro libre para X^{-1}).

Ahora concuerdan el número de parámetros en cada lado en las cinco primeras factorizaciones.

Para la SVD, usemos la forma reducida $\boldsymbol{A_{m\times n} = U_{m\times r}\Sigma_{r\times r}V^{\mathrm{T}}_{r\times n}}$ (¡ los ceros conocidos no son parámetros libres !) Supongamos que $m \leq n$ y que A es una matriz de rango completo con $\boldsymbol{r = m}$. La cantidad de parámetros para A es $\boldsymbol{mn}$. *También lo es la cantidad total para* $U, \Sigma,$ *y* V. El razonamiento detrás de las columnas ortonormales en U y V es el mismo que para las columnas ortonormales en Q.

U tiene $\boldsymbol{\frac{1}{2}\,m(m-1)}$, Σ tiene $\boldsymbol{m}$, V tiene $(n-1)+\cdots+(n-m) = \boldsymbol{mn - \frac{1}{2}\,m(m+1)}$.

Por último, supongamos que A es una *matriz de* m *por* n *de rango* r. **¿ Cuántos parámetros libres hay en una matriz de rango r ?** Podemos contar de nuevo para $U_{m\times r}\Sigma_{r\times r}V^{\mathrm{T}}_{r\times n}$:

U tiene $(m-1)+\cdots+(m-r) = \boldsymbol{mr - \frac{1}{2}\,r(r+1)}$, V tiene $\boldsymbol{nr - \frac{1}{2}\,r(r+1)}$, Σ tiene $\boldsymbol{r}$.

La cantidad total de parámetros para rango r *es* $\boldsymbol{(m+n-r)\,r}$.

Llegamos al mismo número para $A = CR$ en la Sección I.1. Las r columnas de C fueron tomadas directamente de A. La matriz fila R incluye una matriz identidad de r por r (no libre). Así la cantidad para CR concuerda con la cantidad anterior para $U\Sigma V^{\mathrm{T}}$, cuando el rango es r :

C tiene $\boldsymbol{mr}$ parámetros $\quad$ R tiene $\boldsymbol{nr - r^2}$ parámetros $\quad$ Total $\boldsymbol{(m+n-r)\,r}$.

Índice de autores

Para la mayoría de autores mencionados, la página indicada incluye un artículo de una revista científica, un libro o una referencia de arXiv.
También puede consultarse la bibliografía en http://math.mit.edu/~gs/learningfromdata/

Abu-Mostafa, 416
Aggarwal, 416
Alfakih, 259
Alpaydim, 416
Andersson, 108
Arnold, 383
Arnoldi, 115–117, 123

Ba, 366, 367
Bach, 343
Bader, 105, 108
Bahri, 378
Balestriero, 395
Banach, 91
Baraniuk, 196, 395
Bassily, 363
Bau, 115, 117
Bayes, 253, 303
Beckermann, 180, 182, 183
Belkin, 170, 363, 414
Belongie, 395
Ben-David, 416
Bengio, 356, 381, 393, 408, 416
Benner, 182
Bernoulli, 286, 287, 410
Berrada, x, 394
Bertsekas, 188, 356
Bishop, 416
Borre, 165, 302
Bottou, 363
Boyd, 185, 188, 191, 350, 354, 356
Bregman, 185, 192
Bro, 108
Buhmann, 181

Candès, 195, 196, 198, 354
Canny, 390
Carroll, 104
Cauchy, 90, 178, 200
Chang, 104
Chebyshev, 179, 263, 285, 289, 290
Chernoff, 285, 289, 291
Cholesky, 48, 54, 115, 238
Chollet, 416
Chu, 185
Combettes, 191
Cooley, 209
Courant, 174, 176
Courville, 416
Craig, 398
Curtis, 367
Cybenko, 384

Dantzig, 338
Darbon, 193
Daubechies, 237, 384
Davis, 152, 170
De Lathauwer, 107
de Moor, 108
DeVore, 384
Dhillon, 253
Dickstein, 381
Dijksterhuis, 258
Dokmanic, 259
Donoho, 195–198
Douglas, 191
Drineas, 108, 143
Duchi, 366, 367, 416
Durbin, 233

Eckart-Young, 58, 71–74, 179
Eckstein, 185
Efron, 408, 416
Einstein, 91
Eldridge, 170
Elman, 336
Embree, 117, 142
Erdős, 291
Euclides, 259
Euler, 242, 244, 322

Fatemi, 193
Fiedler, 246, 248, 254
Fischer, 174, 176
Fisher, 86, 87
Flyer, 181
Fornberg, 181
Fortin, 188
Fortunato, 182
Fourier, 178, 179, 204, 207, 216, 222, 391
Friedman, 416
Frobenius, 71, 73, 93, 257, 312, 315

Géron, 416
Gangul, 381
Garipov, 365
Gauss, 122, 123, 191, 209, 268, 304, 308
Gibbs, 231
Gilbert, 416
Giles, 272, 382
Gillis, 98
Givens, 119
Glowinski, 188
Goemans, 289
Gohberg, 235
Goldfarb, 193
Goldstein, 193
Golub, 115, 120, 258
Goodfellow, 416
Gordon Wilson, 365

Gower, 258, 261, 363
Gram-Schmidt, 30, 116, 128–130
Gray, 235
Grinfeld, 108

Hölder, 96
Haar, 35
Hadamard, 30, 107, 218
Hager, 167
Hajinezhad, 188
Halko, 143, 151
Hall, 340
Hanin, 378, 383, 384
Hankel, 178, 183
Hansen, 368
Hardt, 356, 367
Harmon, viii
Harshman, 104, 108
Hastie, 99, 100, 198, 199, 416
Hazan, 366, 367
He, 378, 395
Hermite, 206
Hessenberg, 115, 117
Higgs, 268
Higham, 249, 250, 402
Hilbert, 78, 91, 96, 159, 178, 183, 383
Hillar, 104
Hinton, 203, 393, 409, 415
Hitchcock, 104
Hofmann, 414
Hopf, 235
Hornik, 384
Householder, 34, 131, 135

Ioffe, 409
Izmailov, 365

Jacobi, 122, 123, 191, 323
Johnson, 154
Johnson , 406
Jordan, 356

König, 340
Kaczmarz, 122, 185, 193, 363, 368
Kahan, 120, 152, 170
Kale, 367
Kalman, 164, 167, 263, 308, 309
Kalna, 250
Kannan, 151, 155
Karpathy, 392
Khatri-Rao, 105, 106
Kibble, 250
Kingma, 366, 367
Kirchhoff, 18, 241, 243, 336
Kleinberg, 381
Kolda, 105, 108
Kolmogorov, 383
Krizhevsky, 203, 393, 409
Kronecker, 105, 221, 223, 224, 226
Kruskal, 104
Krylov, 115–117, 121, 178, 181, 183
Kullback, 411
Kumar, 367, 394
Kuo, 395

Lagrange, 69, 173, 185, 322, 333
Lanczos, 115, 118
Laplace, 223, 225, 228, 239, 248
Lathauwer, 108
Le, 415
LeCun, 393, 410
Lee, 97, 108, 198
Lei, 193
Leibler, 411
Lessard, 354, 356
Levenberg, 329
Levinson, 233
Lewis-Kraus, 415
Li, 182
Liang, 412
Liao, 384
Liberty, 151
Lim, 104
Lindenstrauss, 154
Lipschitz, 355
Logan, 196
Lorentz, 91
Lustig, 196
Lyapunov, 180

Müller, 408, 416
Ma, 363, 414
Maggioni, 108
Mahoney, 108, 143, 146, 151, 416
Malik, 247
Mallat, 395
Mandal, 414
Markov, 253, 263, 284, 290, 293, 311, 318
Marquardt, 329
Martinsson, 130, 139, 143, 151, 155
Mazumder, 198
Mhaskar, 384
Minsky, 415, 416
Mirsky, 71, 72
Moitra, 416
Moler, 396
Monro, 361
Montavon, 408, 416
Montufar, 381
Moore-Penrose, 133
Morrison, 160, 162, 309

Nakatsukasa, 152, 200
Nash, 340
Needell, 363
Nesterov, 354, 356
Neumann, 318
Newman, 182, 246
Newton, 165, 327, 330, 332
Nielsen, 411, 416
Nocedal, 356
Nowozin, 416
Nyquist, 195
Nystrom, 253

Ohm, 18, 242, 336
Olah, 402
Orr, 408, 416
Oseledets, 108
Osher, 193

Paatero, 108
Packard, 354, 356
Papert, 415, 416
Parhizkar, 259
Parikh, 185, 191
Pascanu, 381
Peaceman, 191
Pearson, 301
Peleato, 185
Pennington, 378
Pentland, 98
Perron, 312, 315, 319
Pesquet, 191
Pick, 183
Pitágoras, 30
Poczos, 363
Podoprikhin, 365
Poggio, 384
Poisson, 223, 229, 276, 287
Polya, 382
Polyak, 351
Poole, 381

Postnikov, 255
Procrustes, 67, 257

Rachford, 191
Raghu, 381
Ragnarsson, 108
Rakhlin, 412
Ramdas, 363
Ranieri, 259
Rao Nadakuditi, 170, 171
Rayleigh, 68, 81, 87, 173, 249, 290
Recht, 198, 354, 356, 367, 416
Reddi, 363, 367
Ren, 378, 395
Renyi, 291
Richtarik, 363
Robbins, 361
Roelofs, 356, 367
Roentgen, 196
Rokhlin, 151
Rolnick, 378, 383, 385
Romberg, 198
Rosebrock, 416
Ruder, 367
Rudin, 193
Ruiz-Antolin, 178, 179, 182
Rumelhart, 415

Sachan, 367
Salakhutdinov, 409
Schölkopf, 414
Schöneman, 258
Scheinberg, 367
Schmidt, 71
Schoenholz, 378
Schur, 177, 335
Schwarz, 61, 90, 96, 200
Seidel, 122, 123, 191
Semencul, 235
Seung, 97, 108
Shalev-Schwartz, 416
Shannon, 195, 411
Shepp, 196
Sherman, 160, 162, 309
Shi, 188, 247
Silvester, 336
Simonyan, 392, 393
Singer, 366, 367
Smilkov, 386
Smola, 363, 414
Sohl-Dickstein, 378
Song, 108
Sorensen, 142
Sra, viii, 360, 363, 365, 416
Srebro, 75, 198, 356, 363, 367
Srivastava, 409
Stanley, 381
Stern, 356, 367
Stewart, 74, 143
Strohmer, 122, 363
Su, 354
Sun, 378, 395
Sutskever, 203, 393, 409
Sylvester, 180–183
Szegö, 235
Szegedy, 409

Tao, 175, 195, 198
Tapper, 108
Taylor, 179, 323, 407
Tegmark, 385
Tibshirani, 99, 100, 184, 416
Toeplitz, 183, 232, 235, 387, 389
Townsend, 78, 178–183
Trefethen, 115, 117
Tropp, 143, 151, 291
Truhar, 182
Tucker, 107
Tukey, 209
Turing, 413
Turk, 98
Tygert, 151
Tyrtyshnikov, 108

Udell, 181, 182

Van Loan, 108, 115, 120, 226, 258
Vandenberghe, 350, 356
Vandermonde, 178, 180, 181
Vandewalle, 108
Vapnik, 414, 416
Veit, 395
Vempala, 151, 155
Vershynin, 122, 363
Vetrov, 365
Vetterli, 259
Vinyals, 356
Vitushkin, 383
von Neumann, 340

Wakin, 196
Wang, 170
Ward, 363
Wathen, 336
Weyl, 172, 175
Wiener, 235
Wilber, 182, 395
Wilkinson, 119
Williams, 415
Wilson, 356, 367
Woodbury, 160, 162, 309
Woodruff, 108, 151
Woolfe, 151
Wright, 356, 416

Xiao, 378
Xu, 98

Yin, 193
Young, 353
Yu, 98

Zadeh, 198
Zaheer, 367
Zaslavsky, 381
Zhang, 98, 356, 378, 395
Zhong, 108
Zhou, 193, 384, 385
Zisserman, 392–394
Zolotarev, 182
Zou, 99, 100

Índice

$\frac{1}{2}\boldsymbol{x}^{\mathrm{T}} S\boldsymbol{x}$ cuadrática, 326

Activación, iv, 375, 376
Actualización, 164–166
AD, 397
ADAGRAD, 366
ADAM, 322, 356, 366
Adaptativas, 407
ADMM, 99, 185, 187, 188
Afín, iii
Ajuste de recta, 136
Aleatorización, ix, 108, 146, 155, 368
AlexNet, ix, 373, 415
Algoritmo de partición, 185, 191
Algoritmo MDS, 261
Algoritmo voraz (greedy), 254
Aliasing, 234
AlphaGo Zero, 394, 412
Antisimétrica, 52
Aprendizaje automático, 371, 413
Aprendizaje por refuerzo, 394, 413
Aprendizaje profundo, iii, vi, 371
Aproximación, 384
Aproximación de rango pequeño, 144, 155
Aproximación racional, 182
Árbol, 17, 240, 244
Árbol generador, 256
Argmin, 186, 322
Arnoldi, 116, 117
Arquitectura, 413
Asimetría, 286
Autocorrelación, 220
Autofunción, 229
Autorostro, 98
Autovalor, 1, 12, 36, 39
Autovalor de $A \oplus B$, 224
Autovalor de $A \otimes B$, 224
Autovalor generalizado, 81
Autovalor repetido, 12
Autovalores de A^{T} y A^k, 42, 70
Autovalores de AB y BA, 59, 64
Autovalores reales, 44
Autovalores repetidos, 69
Autovector, 12, 36, 39, 216
Autovectores, viii, 11, 217
Autovectores a izquierda, 43
Autovectores ortogonales, 44

Banco de filtros, 391
Banda, de, 203, 232
Barra invertida, 113, 184
Base, 4, 5, 15, 204, 239
Base ortonormal, 34, 130
Bernoulli, 287
BFGS (Quasi-Newton), 165
Binomial, 270, 271, 275, 287
Bol, 49
Bola pesada, 351, 366
Bola unitaria, 96, 200
Bootstrap, 408
Bosquejo, 151
Bosques aleatorios, 413
Búsqueda de base, 184, 195
Búsqueda lineal, 351

Cadenas de Markov, 311

Caffe, viii, 374
Cálculo, 396
Cálculo de la SVD, 120, 155
Cálculo de variaciones, 322
Cálculo matricial, 163
Canales, 388
Capa interna, 377
Capa oculta, 371, 372, 387, 401
Cápsula convexa, 331
Cauchy-Schwarz, 90, 96, 200
Centrar (media 0), 75, 270
Centroide, 247, 261
Chernoff matricial, 291
Chernoff superior, 289
Chi cuadrado, 275, 280–282
Ciclo, 17, 241, 256
Ciencia de datos, vi, 11, 71
Cinco condiciones, 49
Circulante, x, 213, 220, 234
Circulantes $CD = DC$, 220
Clasificación, 377
CNN, v, 203, 232, 380
Coche sin conductor, 381
Cociente de Rayleigh, 63, 68, 81, 87, 173, 251
Código, 374
Códigos de aprendizaje automático, 374, 418
Cola pesada, 285
Combinatoria, 373, 381
Compartir los pesos, 387
Competencia de Netflix, 199
Complemento de Schur, 177, 335
Completación con ceros, 233
Completación de matrices, viii, 159, 197, 198, 255
Componente informativa, 171
Componentes principales, 71
Composición, iv, 373, 375, 383
Compresión, 230
Compressed sensing, 146, 159, 196
Condiciones de borde, 229
Condición, 47
Conexión salteada, 412
Conglomerar, 245, 246
Congruente, 53, 85, 87, 177
Conjugado complejo, 205, 215
Conjunto de nivel, 194
Conjunto de predictores, 86, 252, 375
Constante de Lipschitz, 194, 355, 365
Continua y lineal a trozos, 372, 375
Contracción, 189, 191, 357, 358
Convergencia cuadrática, 328
Convergencia cúbica, 119
Convergencia en esperanza, 365
Convergencia lineal, 350, 356
Convexo, 293, 321, 324, 325
ConvNets, 378
Convolución, 203, 214, 220, 283, 387
Convolución cíclica, 214, 218, 234
Convolución de funciones, 219
Convolución en 2D, 388
Correlación, 300, 301
Correlación cruzada, 219
Corte mínimo, 246, 340
Costo cuadrático, 411
Courant-Fischer, 174
Covarianza, 76, 289, 294
Covarianza muestral, viii, 76, 296
CPL, 372, 385
Criterio de Sylvester, 180, 181, 183
Cuadrados mínimos no lineales, 329
Cuantización, 231
Cuarto de círculo, 79, 80
Cuatro subespacios, 18
Cuerda, 332
Cumulante, 287, 288
CURT, 108
Curtosis, 286
Curva acampanada, 279
Curva logística, 373
CVX, 326

Datos de entrenamiento, iii, 359
Datos de prueba, iii, ix
Datos faltantes, 197
DCT, 66, 230, 231
Decaimiento de pesos, 412
Definida positiva, viii, 45–49

Densidad de probabilidad, 266, 273, 278
Derivada, 101, 344, 400
Derivada $d\lambda/dt$, 169
Derivada $d\sigma/dt$, 170
Derivada de A^2, 167
Derivada de A^{-1}, 163
Derivadas segundas, 49, 50, 326
Descenso acelerado, 352, 353
Descenso adaptativo, 356, 361
Descenso estocástico, viii, 359, 361, 400
Descenso más pronunciado, 186, 347, 348, 350
Descenso por gradiente, 322, 344, 349
Descenso proximal, 357
Descomposición en valores singulares SVD, 56
Descomposición polar, 67
Descomposición CP , 97, 104
Descomposición interpolante, 139, 155
Desdoblamiento, 108
Desdoblamiento de tensores, 105
Desigualdad de Chebyshev, 285, 290
Desigualdad de Chernoff, 285
Desigualdad de Hölder, 96
Desigualdad de Markov, 284, 290, 293
Desigualdad triangular, 88
Desigualdades de Weyl, 172, 175, 176
Desplazamiento, 213, 235
Desviación estándar, 265
Desviación grande, 285
Detección de gradiente, 389
Detector de bordes de Canny, 390
Determinante, 36, 42, 47, 48, 346
Determinante mayor, 141
DFT no uniforme, 178, 182
Diagonal constante, 213
Diagonalización, 11, 43, 52, 298
Diamante, 88, 89, 184
Diferencia centrada, 345
Diferenciación automática, 371, 397
Diferencias atrasadas, 123
Diferencias segundas, 123
Dígitos, iii
Dimensiones infinitas, 91
Dimensión, 4, 6
Dirección alternante, 185, 191
Discriminante, 86, 87
Distancia a la media, 284
Distancia de Bregman, 192
Distribución acumulada, 266, 269
Distribución exponencial, 278
Distribución normal, 268, 279, 288
Distribución uniforme, 266, 267
Documento, 98
Dropout, 409, 410
Dualidad, ix, 96, 339, 340, 342, 343
Dualidad débil, 339, 343

Eckart-Young, 58, 71, 72, 74, 75
Ecuación de error, 116, 364
Ecuación de estado, 167
Ecuación de Laplace, 229
Ecuación dual, 405
Ecuación en diferencias, 223
Ecuación maestra, 263, 318
Ecuación normal, 113, 127
Ecuación secular, 171
Eje principal, 51
Elemento finito, 336
Eliminación, 11, 21, 23
Elipse, 50, 62
Energía, 46, 49
Entrelazado, 53, 168, 170, 171, 175
Entrelazamiento, 170
Entrenamiento, 412
Entropía, 411
Entropía cruzada, 360
Envolvente, 391
Época, 361
Equilibrio, 242
Erdős-Renyi, 291
Error de redondeo, 145
Escala de grises, 229
Espacio columna, 1–5, 13, 14
Espacio de Banach, 91
Espacio de frecuencias, 219
Espacio de funciones, 91, 92
Espacio de Hilbert, 66, 91

Espacio fila, 5, 14
Espacio nulo, 6, 14
Espacio nulo a izquierda, 14, 17
Espacios completos, 91
Espirales, 386
Estacionariedad débil, 235
Estado estacionario, 311
Estandarizado, 263, 270, 288
Estiramiento, 62, 67
Estrategia mixta, 341
Estrategia óptima, 341, 343
Estrictamente convexa, 49, 323, 325, 355
Estructura local, 387
Euclídea, 88, 259
Eventos infrecuentes, 277
Expresividad, 373, 381

Factor de descenso, 353
Factorización, 5, 11
Fan-in, 378
Fenómeno de Gibbs , 231
FFT, ix, 204, 209, 211, 229, 234
Filtro, 203, 233, 236, 387
Filtro de Kalman, 164, 167, 308–310
Filtro equiripple, 236
Filtro pasobajo, 236, 391
Filtro pasobanda, 233
Finanzas, 321
Flujo máximo, 339, 340
Forma de Tucker, 107
Forma polar, 215
Forma reducida de la SVD, 57
Funciones ortogonales, 205
Funciones periódicas, 204
Función de aprendizaje, iii, vi, 373, 375
Función de base radial, 181
Función de pérdida, 360, 377, 411
Función de salto, 193
Función delta, 193, 219
Función error, 280
Función generadora, 285, 287
Función indicadora, 189
Función núcleo, 414
Función par, 212
Función rampa, 376
Fórmula cuadrática, 38
Fórmula de coárea, 194

GAN, 413
Gaussiana, 271, 275, 389
Gaussiana discreta, 389
Generalizar, 359, 367, 368, 372
Geometría de la SVD, 62
GMRES, 117
Go, ix, 394, 412
Golub-Kahan, 120
Google, 394
GPS, 165, 302
GPU, 393
Gradiente, 323, 344, 345, 347
Gradiente conjugado, 121
Gradiente del costo, 334
Gráfico de sedimentación, 78
Gráfico por filas, 21
Grafo, 16, 203, 239
Grafo aleatorio, 291, 292
Grafo bipartito, 256, 340
Grafo completo, 240, 244
Grafo computacional, 402
Grafo conexo, 292
Gram-Schmidt, 114, 128
Gramática, 415

Habla, 413
He uniform, 378
Hessiano, 55, 323, 326
Hilbert, problema 13º, 383
Hiperparámetro, 407, 412
Hiperplano, 381, 382
Hiperplano de separación, 380
Householder, 131, 135

iid, 277
ICA, 413
Identidad de matrices, 163
Identidad para σ^2, 265, 274
ImageNet, ix, 373
Incoherencia, 195
Independencia conjunta, 289

Independiente, 3–5, 289
Inicialización, 378
Insesgado, 308
Integral de Fourier, 204, 205
Inteligencia artificial, 371
Intercambio de filas, 26
Interpolación, 180, 363
Interrupción temprana, 360
Invariancia con respecto a desplazamientos, 203
Invariancia lineal con respecto al tiempo, 67, 233
Invariante por desplazamientos, 387
Inversa de $A - UV^{\mathrm{T}}$, 162
Inversa de $A \otimes B$, 221
Isométrica restringida, 196

Johnson-Lindenstrauss, 154
JPEG, 66, 229
Juego para dos jugadores, 340

Kaczmarz, 122, 193, 363–364
Kaczmarz aleatorizado, 363
Karhunen-Loève, 61
Keras, viii, 374, 418
Kirchhoff, 241, 242, 336
Kriging, 253
Krylov, 116, 183
Kullback-Leibler, 411

La ruina del jugador, 317
Lagrangiano, 173, 333
Lagrangiano aumentado, 185, 187
Lanczos, 118
Lanzamientos de monedas, 269
Laplaciana normalizada, 248
Laplaciano de una gaussiana, 390
Laplaciano del grafo, 124, 224, 239, 243, 246
LASSO, 100, 184, 190, 357
Lema de inversión de matriz, 160
Levenberg-Marquardt, 329, 330
Ley asociativa, 13, 163
Ley de conteo, 16
Ley de corriente, 18, 241
Ley de inercia, 53, 177, 337
Ley de los grandes números, 264
Ley de voltaje, 18
Lineal a trozos, x, 375, 381, 385
Log-normal, 275, 280, 281
Longitud al cuadrado, 149
Loop, 17, 241
LTI, 233
LU incompleta, 122

Mal condicionada, 113
Máquina de soporte vectorial, 394
Máquina de Turing, 413
Marco general, 14, 18, 31
Margen, 414
Margen duro, 414
Marginales, 295
Matrices de ceros y unos, 78
Matrices de rango uno, 43
Matrices incompletas, 159
Matrices que conmutan, 43
Matrices semejantes, 38, 43, 85, 119
Matriz bidiagonal, 120
Matriz compañera, 42
Matriz compleja, 45
Matriz de adyacencia, 203, 240, 291
Matriz de conductancia, 124, 336
Matriz de covarianza, 81, 134, 295–297
Matriz de desplazamiento, 387
Matriz de diferencias, 16, 39, 238
Matriz de distancia, 259
Matriz de Fourier, ix, 35, 180, 204, 205, 216
Matriz de ganancia, 164
Matriz de grado, 203, 240
Matriz de Gram, 124
Matriz de Hadamard, 30
Matriz de Hankel, 183
Matriz de Hessenberg, 117
Matriz de Hilbert, 78, 183
Matriz de incidencia, 16, 203, 239, 240
Matriz de Markov, 39, 311–313, 318
Matriz de masa, 81
Matriz de mezcla, 142, 143, 152, 155
Matriz de modularidad, 246

Matriz de pago, 341, 343
Matriz de probabilidad, 295
Matriz de proyección, 127, 153
Matriz de rango dos, 176
Matriz de rigidez, 124, 336
Matriz de Toeplitz, 183, 232, 233, 373, 387, 406
Matriz de transición, 313, 314
Matriz DFT, 205, 207
Matriz DST, 66, 229
Matriz elemental, 244
Matriz estructurada, 180
Matriz hermitiana, 206
Matriz indefinida, 50, 172
Matriz jacobiana, 323
Matriz KKT, 173, 177, 335
Matriz Laplaciana, 203
Matriz normal, 180, 183
Matriz núcleo, 247
Matriz ortogonal, 29, 33, 35, 36, 257
Matriz positiva, 315
Matriz simétrica, 11, 36
Matriz unitaria, 206
Max-min, 174
Max-pooling, 379, 406
Máximo de $R(\boldsymbol{x})$, 81, 172
Mayor varianza, 71
Media, 75, 147, 264, 267
Media de la suma, 299
Media muestral, 264, 296
Método de ADI, 182
Método de las potencias, 95
Método de multiplicadores, 185
Método de Newton, 165, 327, 332
Método DEIM, 142
Método del núcleo, 181, 252
Método multinivel, 249
Método simplex, 338
Métodos de punto interior, 342
Métodos duales, 404
Microarray, 250
Minería de textos, 98
Minilote, ix, 322, 359, 367
Minimax, 174, 335, 342
Minimización alternante, 97, 106, 199, 252
Mínimo, 49, 55, 338
Mínimo aislado, 322
Mínimos cuadrados, iv, 109, 124, 126
Mínimos cuadrados recursivos, 164, 309
MNIST, iii, 355
Modelo cuadrático, 352
Modo hacia adelante, 403
Modo hacia atrás, 397
Modo lote, 361
Modo reverso, 401, 404, 405
Modulación, 208
Momento, 351, 366
Momento central, 286
Momentos, 286
Monte Carlo, 272, 394
Morrison-Woodbury, 160, 162
Movimiento rígido, 259
Muestreo aleatorio, 114, 120, 148, 253, 410
Muestreo de importancia, 363
Muestreo por norma al cuadrado, 122, 146, 149, 156, 363
Multigrilla, 122, 353
Multiplicación, 2, 10, 214
Multiplicación matricial, 7, 10, 13
Multiplicación rápida, 234
Multiplicador de Lagrange, 150, 321, 333
Multiplicidad, 40
Multiplicidad de autovalores, 40
Multivaluado, 192
Multivariada, 275, 280, 304, 305

Neurona, 375
NMF, 97, 98, 190
No diagonalizable, 40
No diagonizable, 40
No negativa, 97
No supervisado, 71
Nodo, 16
norma al cuadrado, 47
Norma de Frobenius, 71, 73, 257
Norma de un tensor, 103
Norma de un vector, 88
Norma del máximo, 95, 96

Norma espectral, 71
Norma matricial, 92, 94
Norma mínima, 126, 356
Norma nuclear, 71, 95, 100, 159, 197, 200
Norma vectorial, 327
Normalización por lotes, x, 409, 412
Normalizar, 128, 270, 409
NP-difícil, 99
Núcleo RBF, 181, 414
Número de condición, 145, 353
Números del pastel, 382
Nyquist-Shannon, 195

Ohm, 336
Ondícula, 391
Ondícula de Haar, 35
OpenCourseWare, x
Optimización, 321
Ortogonal, 11, 29, 52, 128

Partición de Bregman, 192, 193
Parámetros, 70, 419
Parámetros libres, 419
Paso, 379, 390
PCA, 1, 71, 75–77
PCA ralo, 98–100
Penalización, 100, 114
Perceptrón, 415
Pérdida bisagra, 360
Pérdida cuadrática, 360, 411
Pérdida de entropía cruzada, 411
Permutación, 26, 28, 35
Permutación cíclica, 213
Perron-Frobenius, 314, 315
Pesos, 375
Pesos compartidos, 388
Pesos promediados, 365
Pesos que explotan, 378
Pesos que se desvanecen, 378
Pivote, 23, 25, 47, 48
Pivoteo por columnas, 129, 143
Plano de pliegues, 381
Playground, 386
Poisson, 182, 275, 276, 287
Ponderado, 134, 243
Pooling, x, 379
Pooling promedio, 379
Posiciones a partir de distancias, 260
Precisión, 384
Precondicionador, 122, 147
Probabilidad conjunta, 102, 294
Probabilidad de fracaso, 279
Probabilidad total, 268, 274
Problema de máximo, 62, 63, 68
Problema dual, 186, 190, 322, 339
Problema primal, 185
Problemas inversos, 114
Procesamiento de señales, 191, 211, 218
Procesos aleatorios, 235
Procrustes, 257, 258, 261
Producto de Hadamard, 107, 218
Producto de Khatri-Rao, 105, 106
Producto de Kronecker, 105, 221, 227
Producto externo, 9, 10, 103
Producto interno, 9, 10, 91
Programa semidefinido, 198, 342
Programación lineal, 338
Promedio ponderado, 306, 307
Promedios, 236, 365
Proximal, 189, 191
Proyecciones aleatorias, 153
Proyección, 32, 113, 127, 136, 153, 357
Proyectos, vi, viii, 155, 366, 395
Prueba, 412
Prueba de la SVD, 59, 69
Prueba de Logan-Shepp, 196
Prueba del punto medio, 332
Pseudoespectros, 117
Pseudoinversa, 113, 124, 125, 132, 184
Punta, 89, 184
Punto de ensilladura, ix, 50, 53, 81, 168, 172, 174, 186, 335, 341

Quasi-Monte Carlo, 272
Quasi-Newton, 165, 328

Radio espectral, 95
Raíz cuadrada de matrices, 67
Rala, 184
Ralo, 89, 195

Rango, 4, 5, 10, 20
Rango r, 419
Rango asintótico, 79
Rango de $A^{\mathrm{T}}A$ y de AB, 19
Rango de desplazamiento, 182
Rango de un tensor, 103, 104
Rango efectivo bajo, 159, 180
Rango logarítmico, 181
Rango uno, 61, 110, 160, 255, 417
Reconocimiento de imágenes, 387, 389
Recta de nodos, 223
Recta más cercana, 136
Recta tangente, 324, 325, 332
Recursión, 382
Red completamente conectada, v, x, 371, 380
Red convolucional, 380, 387
Red neuronal, iii, v, 377
Red recurrente, 394, 413
Red residual, 395
Reescalar, 300
Reflexión, 33, 34, 131, 237, 391
Reformatear, 226, 227
Regla de convolución, 218, 220
Regla de Cramer, 141
Regla de desplazamiento, 208
Regla de la cadena, 375, 397, 406
Regla del producto, 303
Regresión, 77, 377
Regresión logística, 393
Regresión ridge, 132, 184
Regularización, 132, 410, 412
Relación de recurrencia, 405
Relajar, 184
Relleno de ceros, 391
ReLU, iv, x, 375, 376, 401
Reshape, 226, 227
ResNets, 378
Resonancia magnética (MRI), 196
Respuesta de frecuencia, 232, 236
Retroceso, 328, 351
Retropropagación, 102, 344, 371, 397
Revelación de rango, 138, 143
RNN, 394, 413
Rotaciones de Givens, 119
Rotación, 33, 37, 41, 62, 67
Ruido, 184
Ruido blanco, 134, 306

Saltear conexiones, 395
Salto singular, 79
Saturar, 409, 411
Sección de un tensor, 101
Segmentación, 379
Semiconvergencia, 361
Semidefinida, 47
Semidefinida positiva, 46, 290
Senos y cosenos discretos, 66
Sensibilidad, 406
Separable, 187
Separación de variables, 225
Serie de Chebyshev, 179
Serie de cosenos, 212
Serie de Fourier, 179, 204
Serie de Taylor, 323
Sesgo, iii, 375
Sesgo-varianza, 374, 412
SGD, 359, 361, 367
Sherman-Morrison-Woodbury, 162
Shrinking, 189, 191, 357
SIAM News, 373
Sigmoide, iv, 252
Símbolo, 232, 238
Sobel, 390, 396
Sobreajustar, iii, vi, ix, 359, 360, 409
Sobrerrelajación, 353
Softmax, 393
Solapamiento, 234
Soluciones de $z^N = 1$, 206, 215
Sparse, 89, 184, 195
Spline, 395
Suavidad, 92
Suavizado, 389
Subajustar, 374
Subespacios fundamentales, 14
Subespacios ortogonales, 29
Subgradiente, 188, 191, 192, 355
Submatriz, 65
Submuestreo, 379

Suma de autovalores, 42
Suma de cuadrados, 51
Suma de Kronecker, 223–225, 228
Sumandos de la SVD, 57
Sustitución hacia atrás, 25
SVD, vi, ix, 1, 5, 11, 31, 56, 57, 60, 144
SVD aproximada, 144, 155
SVD generalizada, 85
SVD para derivadas, 65
SVM, 181, 394, 413, 414
SWA, 365
Szegö, 235

Tamaño del paso, vii, 186, 344, 407
Tasa de aprendizaje, vii, 344, 407
Tensor, x, 101, 110
Tensor matricial, 105
Tensorflow, viii, 374, 418
Teorema binomial, 385
Teorema BLUE, 308
Teorema de Bayes, 303
Teorema de Schur, 317
Teorema del límite central, 267, 270, 271, 288
Teorema espectral, 12, 44
Teorema MELI, 308
Teoría de juego, 340
Teoría del campo medio, 378
Toeplitz por bloques, 389
Traducción automática, 394
Traductor de Google, 415
Tramos lineales, 381
Transformada de Fourier discreta, 203, 207
Transformada de Fourier rápida, 178
Transformada de ondículas, 237
Transformada de seno, 229
Transformada inversa, 204
Transformada inversa de Fourier, 217
Transformadas de la mitad de tamaño, 209
Traza, 36, 42, 77
Tren de tensores, 108
Tres bases, 138
Triangular superior, 23
Tridiagonal, 28, 118, 232
Truco del núcleo, 414

Umbral suave, 189, 192, 357
Un pixel, 196
Unidad lineal rectificada, 376
Unitariamente invariante, 72
Universalidad, 384

Validación cruzada, 408, 412
Valor esperado, 149, 264
Valor muestral, 264
Valores singulares, 56
Vandermonde, 178, 180
Variable latente, 182
Variación acotada, 193, 194
Variación total, 190, 193
Varianza, ix, 76, 134, 147, 150, 264, 265, 267
Varianza de $\widehat{x}$, 307
Varianza de la suma, 299
Varianza muestral, 265
Varianza mínima, 150
Varianza total, 77
Vector de Fiedler, 246, 248, 249, 254
Vectores singulares, ix, 56, 59
Vectores singulares a derecha, 60
Vectores singulares a izquierda, 60
Vectorizar (vec), 225
Ventana móvil, 237, 390
Vértice, 338, 343
Vídeo, 226

Wiener-Hopf, 235
Wikipedia, 30, 275, 394, 408

YOGI, 367

Zigzag, 348, 349
Zolotarev, 182

Índice de símbolos

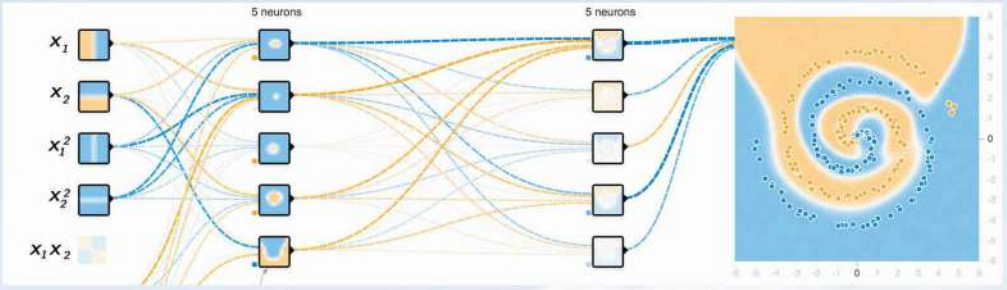
$(AB)\,C$ o $A(BC)$, 399
18.06-18.065, vi, viii, x, 155
$A = CMR$, 142, 151, 156
$A = CR$, 5, 7, 245
$A = LU$, 11, 24, 27
$A = QR$, 11, 129, 143, 156
$A = QS$, 67
$A = UV$, 97
$A = U\Sigma V^{\mathrm{T}}$, 11, 57, 64, 69, 120
$A = X\Lambda X^{-1}$, 11, 39
AB, 9, 10, 64
$AV = U\Sigma$, 56
$AV_r = U_r\Sigma_r$, 57
$A \oplus B$, 223
$A \otimes B$, 221
$A^{\mathrm{T}}CA$, 242, 243
$A^{+} = A^{\dagger} = V\Sigma^{+}U^{\mathrm{T}}$, 125, 132
$A^k = X\Lambda^k X^{-1}$, 39
$H_k = Q_k^{\mathrm{T}} A Q_k$, 117
M-ortogonales, 83
QQ^{T}, 32
$Q^{\mathrm{T}}Q$, 32
$Q^{\mathrm{T}} = Q^{-1}$, 29
$S = A^{\mathrm{T}}A$, 47, 48
$S = A^{\mathrm{T}}CA$, 54
$S = Q\Lambda Q^{\mathrm{T}}$, 11, 12, 44, 51
$\boldsymbol{\nabla F}$, 323, 347
$\boldsymbol{a} * *\boldsymbol{a}$, 220
$\boldsymbol{c} * \boldsymbol{d}$, 214, 220
$\boldsymbol{c} \circledast \boldsymbol{d}$, 214, 218
$\boldsymbol{x}^{\mathrm{T}} S \boldsymbol{x}$, 46, 55
$\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}$, 9
ℓ^1 contra ℓ^2, 308
$\mathrm{C}(\boldsymbol{A})$, 3
$\mathrm{N}\,(0, 1)$, 288, 304
$\mathrm{N}\,(m, \sigma)$, 268
$\mathrm{S} + \mathrm{T}, \mathrm{S} \cap \mathrm{T}, \mathrm{S}^{\perp}$, 20
$\mathsf{Kron}(A, B)$, 221
vec, 225–227
$\mathbf{C}(A^{\mathrm{T}})$, 9
$\mathbf{N}(A)$, 14
$\mathbf{N}(A^{\mathrm{T}}A) = \mathbf{N}(A)$, 20, 135
$\log(\det X)$, 346, 358
$\odot$, 105
$|\lambda_1| \leq \sigma_1$, 61
$||A\boldsymbol{x}||/||\boldsymbol{x}||$, 62
$||\boldsymbol{f}||_C$, 92
k-medias, 97, 245, 247, 251, 252, 254
MATLAB, 45, 82, 108, 221, 249, 418
Julia, 45, 82, 418
.* o $\circ$, 107, 218

Algoritmo QR, 119, 123

Curva en forma de S, 376

Matriz $-1, 2, -1$, 238

Norma ℓ^0, 89, 159
Norma S, 90
Normas $\ell^1, \ell^2, \ell^\infty$, 88, 94, 159, 327



Linear Algebra
Four fundamental subspaces
$A = LU$, $A = QR$, $S = Q\Lambda Q^T$, $A = U\Sigma V^T$
Singular vectors and the SVD
Columns times rows: $A \approx CR$ or CMR

Probability and Statistics
Mean and variance
Covariance and joint probability
Markov chains
Randomized linear algebra

Optimization
Convexity and sparsity
Gradient descent and momentum
Stochastic gradient descent
LASSO and ℓ^1 versus ℓ^2

Deep Learning
Piecewise linear functions
Convolutional neural nets
Backpropagation
Hyperparameters

Support for the reader

ocw.mit.edu (videos for 18.065)
see also Youtube.com/mitocw

math.mit.edu/learningfromdata
math.mit.edu/weborder.php

978-161197738-7
90000
978-1611977387