# Sistemas lineales

Un sistema lineal de $n \times n$ consiste de $n$ ecuaciones con $n$ incógnitas.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n & = b_1 \\ & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n & = b_n \end{cases}$$

También se puede expresar en forma matricial como $Ax = b$ donde:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \in \mathbb{R}^{n \times n} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n \qquad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \in \mathbb{R}^n$$

Dados los coeficientes $A$ y el término independiente $b$ nos interesa encontrar la solución al sistema, es decir todos los $x \in \mathbb{R}^n$ tal que $Ax = b$ (puede no haber solución).

## Imagen

Definimos la imagen de $A \in \mathbb{R}^{n \times n}$ como:

$$Im(A) = \{y \in \mathbb{R}^n \mid \exists x \in \mathbb{R}^n, Ax = y\}$$

La $Im(A)$ son todas las posibles combinaciones lineales de las columnas de $A$.

## Cuándo hay solución?

Si interpretamos el sistema de ecuaciones de manera geométrica, por ejemplo con $n = 2$, tenemos 2 ecuaciones donde cada una define una recta. Lo que buscamos al resolver el sistema es encontrar todos los puntos donde se intersecan dichas rectas. Con esta interpretación, definimos:

- **Sistema incompatible**: no hay solución (las rectas son paralelas sin intersección).
- **Sistema compatible determinado**: hay una única solución (las rectas intersecan en un único punto).
- **Sistema compatible indeterminado**: hay infinitas soluciones (las rectas intersecan en infinitos puntos, son la misma recta).

Esta interpretación se extiende a $\mathbb{R}^n$ donde las ecuaciones definen hiper-planos.

Para un sistema $Ax = b$ cualquiera:

- Si $b \notin Im(A)$ entonces el sistema **no** tiene solución.
- Si $b \in Im(A)$ entonces el sistema tiene solución. La solución es única sii las columnas de $A$ son linealmente independientes. En ese caso, la matriz $A$ resulta inversible y por lo tanto $Ax = b \iff x = A^{-1}b$. Hay una biyección entre $x$ y $b$, necesariamente hay un único $x$ tal que $Ax = b$.

## Sistemas equivalentes

Sean $A \in \mathbb{R}^{n \times n}$, $a \in \mathbb{R}^n$, $B \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$. Dados 2 sistemas lineales $Ax = a$ y $Bx = b$, decimos que son equivalentes sii tienen el mismo conjunto de soluciones. Es decir, para todo $x \in \mathbb{R}^n$, $Ax = a \iff Bx = b$.

Nos interesan los sistemas equivalentes porque muchas veces resolver el sistema original es muy costoso computacionalmente. Pero quizás podemos encontrar otro sistema equivalente y más fácil de resolver.

# Sistemas fáciles

A continuación vemos algunos sistemas de ecuaciones que son fáciles de resolver por la estructura de la matriz asociada al sistema.

## Matriz diagonal

Supongamos que $D \in \mathbb{R}^{n \times n}$ es la matriz **diagonal** asociada al sistema de ecuaciones y queremos encontrar $x \in \mathbb{R}^n$ tal que $Dx = b$ para algún $b \in \mathbb{R}^n$. Como $D$ es diagonal, cada ecuación tiene una única incógnita.

$$\begin{cases} d_{11}x_1 & = b_1 \\ & \vdots \\ d_{nn}x_n & = b_n \end{cases}$$

**Caso $d_{ii} \neq 0$ para todo $i = 1 \dots n$**

Si $d_{ii} \neq 0$ para todo $i = 1 \dots n$, entonces **existe solución y es única**. Podemos despejar unívocamente cada $x_i$ de la siguiente manera:

$x_i = b_i / d_{ii}$ para todo $i = 1 \dots n$.

Obtenemos la solución en $O(n)$ operaciones elementales.

**Caso existe algún $d_{ii} = 0$**

Si existe algún $d_{ii} = 0$ el sistema puede o no tener solución. Esto depende del valor de $b_i$.

- Si $d_{ii} = 0$ y $b_i = 0$ entonces **hay infinitas soluciones** ya que podemos elegir cualquier valor para $x_i$ tal que $d_{ii}x_i = b_i$ pues $0 * x_i = 0$ se cumple para cualquier $x_i$.
- Si $d_{ii} = 0$ y $b_i \neq 0$ entonces **no hay solución**.

## Matriz triangular superior (backwards substitution)

Supongamos que $U \in \mathbb{R}^{n \times n}$ es la matriz **triangular superior** asociada al sistema de ecuaciones y queremos encontrar $x \in \mathbb{R}^n$ tal que $Ux = b$ para algún $b \in \mathbb{R}^n$.

**Caso $u_{ii} \neq 0$ para todo $i = 1 \dots n$**

Como $U$ es una matriz triangular, si $u_{ii} \neq 0$ para todo $i = 1 \dots n$ podemos despejar cada coordenada de $x$ empezando desde la última hasta la primera, utilizando los resultados previos en cada paso. En este caso **siempre existe solución y es única**. Esto se conoce como **backwards substitution** (sustitución hacia atrás).

Para calcular $x_n$ podemos despejar de forma directa:

$$x_n = b_n / u_{nn}$$

Yendo hacia atrás, si miramos la ecuación $n - 1$ tenemos:

$$u_{n-1n-1}x_{n-1} + u_{n-1n}x_n = b_{n-1}$$

Dado que ya calculamos $x_n$ en el primer paso, podemos despejar $x_{n-1}$ de la siguiente manera:

$$x_{n-1} = (b_{n-1} - u_{n-1n}x_n) / u_{n-1n-1}$$

El caso general para $x_i$ utiliza todos los valores ya calculados previamente para $x_{i+1}, \ldots, x_n$.

$$x_i = (b_i - \sum_{j=i+1}^{n} u_{ij} x_j)/u_{ii}$$

Finalmente obtenemos $x_1$ con la misma estrategia, utilizando todas las coordenadas ya calculados de $x$ desde 2 hasta $n$:

$$x_1 = (b_1 - \sum_{j=2}^{n} u_{1j} x_j)/u_{11}$$

Recordemos que todas estas operaciones están bien definidas ya que $u_{ii} \neq 0$ para todo $i = 1 \ldots n$. Así obtenemos $x \in \mathbb{R}^n$ solución al sistema $Ux = b$ en $O(n^2)$ operaciones elementales.

**Caso existe algún $u_{ii} = 0$**

En este caso **puede o no haber solución**. Supongamos que existe algún $i \in [1, n]$ tal que $u_{ii} = 0$. En el paso $i$-ésimo del backwards substitution tenemos que despejar $x_i$. No podemos dividir por $u_{ii} = 0$ por lo tanto nos queda:

$$0 = x_i u_{ii} = b_i - \sum_{j=i+1}^{n} u_{ij} x_j$$

Solo hay solución si se cumple:

$$b_i - \sum_{j=i+1}^{n} u_{ij} x_j = 0$$

Y en tal caso hay **infinitas** soluciones ya que podemos elegir cualquier valor para $x_i$. Caso contrario el sistema no tiene solución.

### Matriz triangular inferior (forward substitution)

Si en vez de $U$ matriz triangular superior tuviésemos una matriz $L$ **triangular inferior**, se puede aplicar la misma estrategia (con las mismas consideraciones sobre $l_{ii}$) realizando un **forward substitution** (sustitución hacia adelante). Consiste en primero despejar $x_1$ e ir avanzando hasta llegar a $x_n$, utilizando las coordenadas previamente calculadas en cada paso.

## Sistemas generales

En el caso general, si la matriz asociada al sistema no es uno de los casos fáciles, nos gustaría tener un procedimiento que permita encontrar otro sistema equivalente que sí sea fácil de resolver.

### Eliminación Gaussiana

Para esto podemos aplicar el método de **Eliminación Gaussiana** que consiste en sumar/restar ecuaciones (filas), multiplicarlas por un escalar y permutarlas. El objetivo es triangular la matriz del sistema y luego obtener una solución (si existe) con backwards substitution.

Este procedimiento también afecta al término independiente, ya que las transformaciones que realizamos para triangular la matriz tiene que generar un nuevo sistema equivalente al original.

Sean $A \in \mathbb{R}^{n\times n}$ y $b \in \mathbb{R}^n$. Extendemos la matriz $A$ agregando $b$ como la última columna. Llamamos $A^k$ con $k = 0 \dots n-1$ a la matriz extendida en el paso $k$ de la Eliminación Gaussiana.

$$A^0 = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{bmatrix} \in \mathbb{R}^{n\times(n+1)}$$

En el primer paso utilizamos $a^0_{11}$ como pivote para colocar ceros en la primer columna debajo de la diagonal principal.

$$A^0 = \begin{bmatrix} a^0_{11} & a^0_{12} & \dots & a^0_{1n} & b^0_1 \\ a^0_{21} & a^0_{22} & \dots & a^0_{2n} & b^0_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a^0_{i1} & a^0_{i2} & \dots & a^0_{in} & b^0_i \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a^0_{n1} & a^0_{n2} & \dots & a^0_{nn} & b^0_n \end{bmatrix} \begin{matrix} \\ F_2 - (a^0_{21}/a^0_{11})F_1 \\ \vdots \\ \rightarrow \quad F_i - (a^0_{i1}/a^0_{11})F_1 \\ \vdots \\ F_n - (a^0_{n1}/a^0_{11})F_1 \end{matrix} \rightarrow \begin{bmatrix} a^1_{11} & a^1_{12} & \dots & a^1_{1n} & b^1_1 \\ 0 & a^1_{22} & \dots & a^1_{2n} & b^1_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & a^1_{i2} & \dots & a^1_{in} & b^1_i \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & a^1_{n2} & \dots & a^1_{nn} & b^1_n \end{bmatrix} = A^1$$

En el paso $i$-ésimo ya "triangulamos" (pusimos ceros debajo de la diagonal) las columnas 1 a $i-1$. Ahora repetimos el mismo procedimiento para la columna $i$, utilizando como pivote $a^{i-1}_{ii}$. Notemos que en cada paso $i$ usamos la matriz $A^{i-1}$ del paso anterior.

$$A^{i-1} = \begin{bmatrix} a^{i-1}_{11} & a^{i-1}_{12} & \dots & a^{i-1}_{1i} & \dots & a^{i-1}_{1n} & b^{i-1}_1 \\ 0 & a^{i-1}_{22} & \dots & a^{i-1}_{2i} & \dots & a^{i-1}_{2n} & b^{i-1}_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & a^{i-1}_{ii} & \dots & a^{i-1}_{in} & b^{i-1}_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & a^{i-1}_{ni} & \dots & a^{i-1}_{nn} & b^{i-1}_n \end{bmatrix} \begin{matrix} \longrightarrow \\ \\ F_{i+1} - (a^{i-1}_{i+1i}/a^{i-1}_{ii})F_i \\ \vdots \\ F_n - (a^{i-1}_{ni}/a^{i-1}_{ii})F_i \end{matrix}$$

$$\begin{bmatrix} a^i_{11} & a^i_{12} & \dots & a^i_{1i} & a^i_{1i+1} & \dots & a^i_{1n} & b^i_1 \\ 0 & a^i_{22} & \dots & a^i_{2i} & a^i_{2i+1} & \dots & a^i_{2n} & b^i_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & a^i_{ii} & a^i_{ii+1} & \dots & a^i_{in} & b^i_i \\ 0 & 0 & \dots & 0 & a^i_{i+1i+1} & \dots & a^i_{i+1n} & b^i_{i+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & a^i_{ni+1} & \dots & a^i_{nn} & b^i_n \end{bmatrix} = A^i$$

Al concluir los $n-1$ pasos, obtenemos la matriz $A^{n-1}$ que contiene un sistema equivalente al original y además la matriz del sistema es una matriz triangular superior.

$$A^{n-1} = \begin{bmatrix} a^{n-1}_{11} & a^{n-1}_{12} & \dots & a^{n-1}_{1i} & \dots & a^{n-1}_{1n} & b^{n-1}_1 \\ 0 & a^{n-1}_{22} & \dots & a^{n-1}_{2i} & \dots & a^{n-1}_{2n} & b^{n-1}_2 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & a^{n-1}_{ii} & \dots & a^{n-1}_{in} & b^{n-1}_i \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & \dots & a^{n-1}_{nn} & b^{n-1}_n \end{bmatrix}$$

### Condición necesaria

**La Eliminación Gaussiana requiere que $a_{ii}^{i-1} \neq 0$ para todo $i = 1 \dots n-1$.** Notar que **no** es suficiente que $A^0$ (la matriz original) tenga la diagonal sin ceros, esta condición pide que en cada paso $i$ de la Eliminación Gaussiana, el pivote $a_{ii}^{i-1} \neq 0$. Puede suceder que la matriz original no tenga ceros en la diagonal pero que aparezcan luego de algunos pasos.

Si en el paso $i$-ésimo sucede que $a_{ii}^{i-1} = 0$ quizás podemos continuar con la Eliminación Gaussiana.

- Si la columna $i$ ya tiene todo ceros debajo de la diagonal, es decir $a_{ji}^{i-1} = 0$ para todo $j = i+1 \dots n$, entonces no hace falta operar con las filas y podemos avanzar al siguiente paso.
- Caso contrario, si existe algún $a_{ji}^{i-1} \neq 0$ con $j > i$ podemos permutar la fila $i$ con la $j$ y continuar con la Eliminación Gaussiana.

### Pivoteo

Las estrategias de pivoteo no solo permiten realizar la Eliminación Gaussiana cuando el pivote es nulo, si no que también se puede utilizar para minimizar los errores numéricos al trabajar con aritmética finita. Notemos que en cada paso estamos diviendo por $a_{ii}^{i-1}$. Para minimizar los errores numéricos conviene siempre dividir por el mayor número posible. En cada paso de la Eliminación Gaussiana podemos intercambiar filas y/o columnas de tal forma que $|a_{ii}^{i-1}|$ sea lo más grande posible.

- Pivoteo parcial (por filas): En el paso $i$-ésimo elegimos como pivote el mayor $|a_{ji}^{i-1}|$ con $j \in [i, n]$. Tenemos que permutamos la fila $j$ con la $i$.
- Pivoteo completo (por filas y columnas): En el paso $i$-ésimo elegimos como pivote el mayor $|a_{jk}^{i-1}|$ con $j, k \in [i, n]$. Es decir miramos ahora toda la submatriz aún no triangulada para elegir el pivote. Luego tenemos que realizar las permutaciones necesarias para colocar el elemento $a_{jk}^{i-1}$ en la posición $i, i$.

### Complejidad

El método de Eliminación Gaussiana tiene una complejidad de $O(n^3)$ operaciones elementales. Dado que triangulamos la matriz del sistema junto al término independiente, si ahora queremos reutilizar la misma matriz pero con otro término independiente necesitamos volver a pagar el mismo costo cúbico.

Cuando tenemos un sistema que vamos a necesitar resolverlo muchas veces para distintos términos independientes, es más eficiente encontrar una descomposición de la matriz, como por ejemplo la descomposición LU o QR.

# Factorización LU

## Motivación

Supongamos que necesitamos resolver el sistema $Ax = b$ varias veces, es decir, para distintos términos independientes $b$. Usando Eliminación Gaussiana nos cuesta $O(n^3)$ **cada vez**, ya que al triangular la matriz del sistema también modificamos el término independiente. Si cambiamos $b$, hay que volver a triangular, por más que sea la misma matriz $A$.

Nos gustaría de alguna forma preservar la secuencia de operaciones que afectan al término independiente $b$ durante la Eliminación Gaussiana para poder reaplicarlas sobre otro término independiente nuevo, y así reutilizar el sistema previamente triangulado.

## Definición

Sean $A \in \mathbb{R}^{n \times n}$, $L \in \mathbb{R}^{n \times n}$ matriz triangular inferior y $U \in \mathbb{R}^{n \times n}$ matriz triangular superior tal que $A = LU$.

$$Ax = b \iff LUx = b$$

Con la factorización LU podemos resolver el sistema en 2 etapas:

$$Ly = bUx = y$$

Como $L$ y $U$ son triangulares (inferior/superior), resolver cada uno de esos sistemas solo cuesta $O(n^2)$ usando backwards/forward substitution.

## Procedimiento

Sea $A \in \mathbb{R}^{n \times n}$ la matriz del sistema. Llamamos $A^k$ con $k = 0 \dots n-1$ a la matriz resultante después del paso $k$-ésimo de la Eliminación Gaussiana. Asumimos que $a_{ii}^{i-1} \neq 0$ para todo $i = 1 \dots n-1$.

En cada paso de la Eliminación Gaussiana colocamos ceros debajo de la diagonal en una columna en particular. Busquemos representar un paso de la Eliminación Gaussiana como una multiplicación de matrices.

Sea $M^1 \in \mathbb{R}^{n \times n}$ la matriz que codifica el paso 1 de la Eliminación Gaussiana. Esta matriz tiene unos en la diagonal y debajo de ella en la columna 1 tiene los multiplicadores necesarios tal que $M^1 A^0 = A^1$.

$$M^1 = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ -m_{21} & 1 & 0 & \dots & 0 \\ -m_{31} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -m_{n1} & 0 & 0 & \dots & 1 \end{bmatrix}$$

$$M^1 A^0 = \begin{bmatrix} a_{11}^0 & \dots & a_{1n}^0 \\ a_{21}^0 - m_{21} a_{11}^0 & \dots & a_{2n}^0 - m_{21} a_{1n}^0 \\ \vdots & \vdots & \vdots \\ a_{n1}^0 - m_{n1} a_{11}^0 & \dots & a_{nn}^0 - m_{n1} a_{1n}^0 \end{bmatrix} = A^1$$

Notemos que $fila_i(A^1) = fila_i(A^0) - m_{i1} fila_1(A^0)$ para todo $i = 2 \dots n$. Basta tomar $m_{i1} = a_{i1}^0 / a_{11}^0$ para colocar ceros debajo de la diagonal en la columna 1 de $A^1$.

$$A^1 = \begin{bmatrix} a_{11}^0 & \dots & a_{1n}^0 \\ 0 & \dots & a_{2n}^0 - m_{21} a_{1n}^0 \\ \vdots & \vdots & \vdots \\ 0 & \dots & a_{nn}^0 - m_{n1} a_{1n}^0 \end{bmatrix}$$

En el caso general del paso $i$-ésimo de la Eliminación Gaussiana tenemos $M^i$ tal que $M^i A^{i-1} = A^i$.

$$M^i = \begin{bmatrix} 1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & \dots & -m_{i+1i} & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \underbrace{-m_{ni}}_{\text{columna } i} & 0 & \dots & 1 \end{bmatrix}$$

Donde $m_{ji} = a_{ji}^{i-1} / a_{ii}^{i-1}$ con $i = 1 \dots n-1$ (la columna donde estamos poniendo los ceros) y $j = i+1 \dots n$.

Todo el método de la Eliminación Gaussiana se codifica como el siguiente producto matricial entre la matriz original $A$ y todas las $M^i$, dando como resultado $U \in \mathbb{R}^{n \times n}$ triangular superior.

$$M^{n-1}M^{n-2}\dots M^1 A = U$$

**Propiedades de $M^i$**

La matriz $M^i$ la podemos representar de la siguiente forma.

$$M^i = I - m_i e_i^t$$

Donde el vector $m_i \in \mathbb{R}^n$ contiene los multiplicadores del paso $i$-ésimo de la Eliminación Gaussiana y $e_i$ es el $i$-ésimo vector canónico.

$$\begin{aligned} m_i &= \begin{bmatrix} 0 & \dots & 0 & 0 & m_{i+1i} & \dots & m_{ni} \end{bmatrix} \\ e_i &= \begin{bmatrix} 0 & \dots & 0 & 1 & 0 & \dots & 0 \end{bmatrix} \end{aligned}$$

$M^i$ es triangular inferior por construcción. Su determinante es 1 ya que es el producto de su diagonal, que son todos unos. Luego $M^i$ es inversible. ¿Quién es su inversa?

$$M^i(M^i)^{-1} = (I - m_i e_i^t)(I + m_i e_i^t) = I + m_i e_i^t - m_i e_i^t - m_i e_i^t m_i e_i^t = I - m_i \underbrace{e_i^t m_i}_{=0} e_i^t = I$$

Entonces $(M^i)^{-1} = I + m_i e_i^t$.

**¿Quién es L?**

$M^{n-1}M^{n-2}\dots M^1 A = UA = (M^1)^{-1}(M^2)^{-1}\dots(M^{n-1})^{-1}UA = (I+m_1e_1^t)(I+m_2e_2^t)\dots(I+m_{n-1}e_{n-1}^t)U$

Observando que $m_i e_i^t m_j e_j^t = 0$ si $j > i$ obtenemos:

$$A = (I + m_1 e_1^t + m_2 e_2^t + \dots + m_{n-1} e_{n-1}^t)U$$

Definimos entonces $L = I + m_1 e_1^t + m_2 e_2^t + \dots + m_{n-1} e_{n-1}^t$ matriz triangular inferior con unos en la diagonal y todos los multiplicadores usados durante la triangulación de $A$. Así obtenemos la factorización $LU$ de $A$.

$$A = LU$$

**Complejidad**

Obtener la factorización $LU$ cuesta $O(n^3)$ operaciones elementales. En esencia es la misma complejidad que la Eliminación Gaussiana.

## Propiedades

Sea $A \in \mathbb{R}^{n\times n}$.

- Si $A$ es inversible y tiene factorización $LU$, entonces es única.
- Si todas las submatrices principales de $A$ son inversibles, entonces tiene factorización $LU$.
- Si $A$ es estrictamente diagonal dominante, entonces tiene factorización $LU$.
- No toda matriz tiene factorización $LU$.

# Factorización PLU

Si existe algún $a_{ii}^{i-1} = 0$ la Eliminación Gaussiana puede continuar permutando filas. Dado que queremos encontrar una factorización que codifique los pasos de la Eliminación Gaussiana para poder resolver el mismo sistema con otros términos independientes de forma más eficiente que repitiendo toda la Eliminación Gaussiana otra vez, necesitamos guardar ahora las permutaciones realizadas.

Partiendo de una matriz de permutación $P = I$ que no permuta ninguna fila, buscamos la factorización $PLU$ tal que:

$$PA = LU$$

En cada paso, si necesitamos permutar alguna fila actualizamos la matriz $P$ con la permutación necesaria. Luego resolver un sistema $Ax = b$ es similar al caso sin permutación:

$$Ax = b \iff PLUx = b \iff LUx = P^{-1}b$$

Notemos que $P^{-1}b$ aplica las permutaciones realizadas durante la Eliminación Gaussiana al término independiente $b$ para que se correspondan las coordenadas de $b$ con las ecuaciones permutadas.

Observación: $P^{-1} = P$ por ser matriz de permutación.

# Normas

# SDP (simétrica definida positiva)

## Definición

Una matriz $A \in \mathbb{R}^{n \times n}$ es **simétrica definida positiva** (SDP) si se cumplen 2 condiciones:

1. $A$ es simétrica.
   $A = A^t$
2. $A$ es definida positiva.
   $x^t Ax > 0$ para todo $x \in \mathbb{R}^n$, $x \neq 0$

## Propiedades

Las matrices SDP tienen varias propiedades muy convenientes.

- A es inversible.
  Si $A$ no fuese inversible entonces existe $x \in \mathbb{R}^n$, $x \neq 0$ tal que $Ax = 0$. Luego $x^t Ax = x^t 0 = 0 \not> 0$. Absurdo pues $A$ es SDP.
- $a_{ii} > 0$ para todo $i = 1 \dots n$.
  Como $A$ es SDP, en particular $e_i^t A e_i = a_{ii} > 0$ para todo $i = 1 \dots n$.
- Toda submatriz principal también es SDP.
  Sea $A^k$ la submatriz de orden $k = 1 \dots n$. Es fácil ver que $A^k$ es simétrica ya que es un "recorte" de la matriz original $A$ que es simétrica. Supongamos que $A^k$ no es definida positiva. Entonces existe $x \in \mathbb{R}^k$, $x \neq 0$ tal que $x^t A^k x \leq 0$. Sea $\overline{x} = (x, 0) \in \mathbb{R}^n$ la extensión de $x$ a $\mathbb{R}^n$. Notemos que $\overline{x} \neq 0$, entonces $\overline{x}^t A \overline{x} > 0$ porque $A$ es definida positiva. Si planteamos el producto matricial por bloques vemos que $\overline{x}^t A \overline{x} = x^t A^k x > 0$. Llegamos a una contradicción, luego $A^k$ también es definida positiva.
- $A$ es SDP sii $B^t AB$ es SDP con $B$ inversible.
  La simetría se ve fácil verificando que $B^t AB = (B^t AB)^t$. Para ver que es definida positiva planteamos

$x^t B^t ABx$ y usando que $B$ es inversible hacemos un cambio de variable $Bx = y$. Luego se verifica que $y^t Ay > 0$ pues $A$ es SDP.

- La submatriz conformada por las filas y columnas 2 a $n$ después del primer paso de la Eliminación Gaussiana es SDP.
  Usando la propiedad anterior vemos que $M_1 AM_1^t$ es SDP y necesariamente la submatriz mencionada es también SDP.
- Se puede realizar el método de Eliminación Gaussiana sin necesidad de permutar filas.
  Aplicando la propiedad anterior recursivamente podemos afirmar esta propiedad.
- Existe factorización $LU$.
  Corolario de las propiedades anteriores.

## Factorización Cholesky

La Factorización de Cholesky es otra manera de descomponer una matriz para poder resolver sistemas de ecuaciones lineales de manera fácil.

Sea $A \in \mathbb{R}^{n \times n}$ una matriz SDP, entonces existe factorización $A = LU$. Como $A$ es simétrica y $L$ inversible por ser triangular inferior con unos en la diagonal:

$$A = A^t \iff LU = (LU)^t \iff LU = U^t L^t \Rightarrow U(L^t)^{-1} = L^{-1} U^t$$

$U(L^t)^{-1}$ es triangular superior por ser producto de triangulares superiores. $L^{-1}U^t$ es triangular inferior por ser producto de triangulares inferiores.

$$U(L^t)^{-1} = L^{-1}U^t = D \Rightarrow U = DL^t$$

Donde $D$ es una matriz diagonal. Volviendo a la factorización $LU$ de $A$ tenemos:

$$A = LU = LDL^t$$

$A$ es SDP y además $L^t$ es inversible. Entonces existe $x \in \mathbb{R}^n$, $x \neq 0$ tal que $L^t x = e_i$ para todo $i = 1 \dots n$.

$$0 < x^t Ax = x^t LDL^t x = (L^t x)^t D(L^t x) = e_i^t De_i = d_{ii}$$

Como todos los elementos de la diagonal $D$ son positivos, podemos descomponer $D$ como el producto de dos matrices diagonales (que son iguales) donde tomamos la raíz cuadrada a toda la diagonal.

$$D = \sqrt{D}\sqrt{D}$$

Donde $(\sqrt{D})_{ii} = \sqrt{d_{ii}}$ para todo $i = 1 \dots n$.

Finalmente encontramos la factorización de Cholesky de $A$.

$$A = L\sqrt{D}\sqrt{D}L^t = L\sqrt{D}\sqrt{D}^t L^t = (L\sqrt{D})(L\sqrt{D})^t = \tilde{L}\tilde{L}^t$$

Donde $\tilde{L} = L\sqrt{D}$.

### Unicidad

Sea $A \in \mathbb{R}^{n \times n}$. $A$ es SDP sii tiene factorización de Cholesky: $A = \tilde{L}\tilde{L}^t$. Si además fijamos el signo de la diagonal de $\tilde{L}$, esta factorización es única.

Se puede probar por el absurdo suponiendo que existen 2 factorizaciones distintas. Finalmente se llega a una relación entre ambas factorizaciones donde quedan igualadas a $I$. De ahí se concluye que necesariamente comparten el mismo signo en la diagonal.

### Algoritmo

Si bien toda matriz SDP tiene factorización $LU$, y desde ahí podemos obtener la factorización de Cholesky, también existe un algoritmo para encontrar la factorización de Cholesky de forma directa.

Sea $A \in \mathbb{R}^{n \times n}$ y $A = \tilde{L}\tilde{L}^t$ su factorización de Cholesky. En líneas generales el algoritmo consiste en plantear el producto matricial $\tilde{L}\tilde{L}^t$ e igualar cada elemento con los de $A$. Luego vamos despejando los elementos de $\tilde{L}$ uno por uno, utilizando los valores previamente despejados (similar al backwards substitution).

### Complejidad

En general obtener la factorización de Cholesky tiene un costo de $O(n^3)$ operaciones elementales. No obstante, posee una mejor constante respecto a la factorización $LU$.

Cholesky es $O(n^3/3)$ mientras que $LU$ es $O(2n^3/3)$.

# QR

## Matrices ortogonales

Una matriz $Q \in \mathbb{R}^{n \times n}$ es ortogonal sii $Q^{-1} = Q^t$.

Equivalentemente $Q$ es ortogonal si:

- Las columnas de $Q$ forman un conjunto ortonormal (todas las columnas son ortogonales entre sí y tienen norma 2 igual a 1). Sean $v_1, \ldots, v_n$ las columnas de $Q$.

$$Q \text{ ortogonal} \iff \begin{cases} v_i^t v_j = 0 & i \neq j \\ v_i^t v_i = ||v_i||_2^2 = 1 & \text{caso contrario} \end{cases}$$

- Las filas de $Q$ forman un conjunto ortonormal.
- $Q$ vista como una transformación lineal es una rotación o reflexión que preserva la norma 2.

$$||Qx||_2 = ||x||_2 \iff ||Qx||_2^2 = ||x||_2^2$$

$$||Qx||_2^2 = (Qx)^t(Qx) = x^t \underbrace{Q^tQ}_{I} x = x^t x = ||x||_2^2$$

### Propiedades

- $||Q||_2 = 1$
- $\kappa_2(Q) = 1$ (las matrices ortogonales son estables numéricamente)
- $det(Q) = \pm 1$
- $Q$ y $P$ ortogonales $\Rightarrow QP$ es ortogonal

## Factorización QR

Sea $A \in \mathbb{R}^{n\times n}$, $Q \in \mathbb{R}^{n\times n}$ ortogonal y $R \in \mathbb{R}^{n\times n}$ triangular superior tal que $A = QR$.

La factorización $QR$ es otra forma de descomponer una matriz para encontrar un sistema equivalente y fácil de resolver. Además, esta factorización tiene otras ventajas asociadas al hecho de que $Q$ es ortogonal y tiene un número de condición igual a 1.

Dado $b \in \mathbb{R}^n$ buscamos todos los $x \in \mathbb{R}^n$ tal que $Ax = b$.

$$Ax = b \iff QRx = b \iff Rx = Q^t b$$

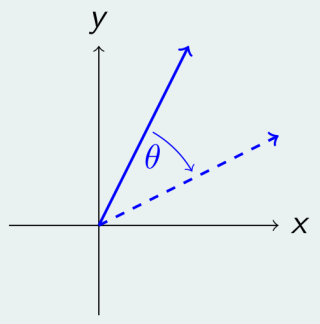
Podemos encontrar $x$ en $O(n^2)$ operaciones elementales con backwards substitution ya que $R$ es triangular superior.

Veamos ahora cómo podemos obtener la factorización $QR$. Existen principalmente 2 métodos.

## Método Givens (rotationes)

El método de Givens consiste en realizar una serie de rotaciones en un subespacio de dimensión 2 para ir colocando ceros estratégicamente debajo de la diagonal principal.

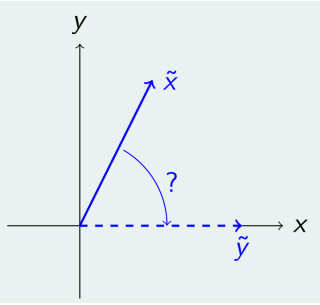
### Rotaciones en el plano




Sea $W \in \mathbb{R}^{2\times 2}$ una matriz ortogonal que rota cualquier vector una cantidad $\theta$ de grados en sentido horario.

$$W = \begin{bmatrix} cos(\theta) & sen(\theta) \\ -sen(\theta) & cos(\theta) \end{bmatrix}$$

La matriz $W$ codifica una rotación para un cierto ángulo $\theta$ el cual está fijo. Es decir, $W$ rota **cualquier** vector siempre por el mismo ángulo. Sin embargo, no nos interesa definir explícitamente el ángulo $\theta$. En cambio, queremos definir $W$ a partir de un vector en particular y hacia dónde lo queremos rotar, con el objetivo final de triangular una matriz de $2 \times 2$.



Dado $\tilde{x} \in \mathbb{R}^2$, queremos construir la matriz de rotación $W \in \mathbb{R}^{2\times 2}$ para que rote al vector $\tilde{x}$ lo necesario para dejarlo "apoyado sobre el eje x". Es decir, $W\tilde{x} = \tilde{y}$ con $\tilde{y} \in \mathbb{R}^2$, $\tilde{y} = (||\tilde{x}||_2, 0)$.

En vez de buscar el ángulo y luego calcular los cosenos y senos, usamos las equivalencias trigonométricas (coseno = adyacente/hipotenusa, seno = opuesto/hipotenusa) para despejar los cosenos y senos sin necesidad de conocer el ángulo de rotación. De esta forma se puede obtener la matriz $W$ para cualquier par de vectores $\tilde{x}$ e $\tilde{y}$ que cumplan con las condiciones antes mencionadas.

$$W = \begin{bmatrix} \tilde{x}_1/||\tilde{x}||_2 & \tilde{x}_2/||\tilde{x}||_2 \\ -\tilde{x}_2/||\tilde{x}||_2 & \tilde{x}_1/||\tilde{x}||_2 \end{bmatrix}$$

**Caso matriz** $2 \times 2$

Si tenemos una matriz cualquiera $A \in \mathbb{R}^{2\times 2}$, podemos triangularla de la siguiente manera. Primero definimos los vectores $\tilde{x}$ e $\tilde{y}$.

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad \tilde{x} = \begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix} \qquad \tilde{y} = \begin{bmatrix} ||\tilde{x}||_2 \\ 0 \end{bmatrix}$$

Por lo expuesto anteriormente, sabemos que existe $W \in \mathbb{R}^{2\times 2}$ tal que $W\tilde{x} = \tilde{y}$. Luego:

$$WA = \begin{bmatrix} ||\tilde{x}||_2 & * \\ 0 & * \end{bmatrix} = R$$

Como $W$ es ortogonal, su inversa es $W^t$.

$$WA = R \iff W^tWA = W^tR \Rightarrow A = QR$$

Donde $Q = W^t$ y $R$ es una matriz triangular superior.

**Caso matriz** $n \times n$

Sea $A \in \mathbb{R}^{n\times n}$. Para el caso general de una matriz de $n \times n$ vamos a necesitar realizar muchas rotaciones para triangular la matriz. Cada rotación coloca un único cero debajo de la diagonal principal.

Para el primer paso definimos $\tilde{x} = \begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix}, \tilde{y} = \begin{bmatrix} ||\tilde{x}||_2 \\ 0 \end{bmatrix}$.

Al igual que el caso de $2 \times 2$ sabemos que existe $W \in \mathbb{R}^{2\times 2}$ ortogonal tal que $W\tilde{x} = \tilde{y}$.

$$W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} = \begin{bmatrix} \tilde{x}_1/||\tilde{x}||_2 & \tilde{x}_2/||\tilde{x}||_2 \\ -\tilde{x}_2/||\tilde{x}||_2 & \tilde{x}_1/||\tilde{x}||_2 \end{bmatrix}$$

Ahora embebemos $W$ en una matriz identidad de $n \times n$ la cual llamamos $W_{12} \in \mathbb{R}^{n\times n}$ (los subíndices hacen referencia a las 2 coordenadas que definen el plano sobre el cual vamos a realizar la rotación).

$$W_{12} = \begin{bmatrix} w_{11} & w_{12} & 0 & \dots & 0 \\ w_{21} & w_{22} & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

El primer paso de la factorización QR utilizando el método de Givens resulta:

$$W_{12}A = \begin{bmatrix} * & * & \dots & * \\ 0 & * & \dots & * \\ a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

Para el segundo paso redefinimos $\tilde{x} = \begin{bmatrix} a_{11}^1 \\ a_{31}^1 \end{bmatrix}, \tilde{y} = \begin{bmatrix} ||\tilde{x}||_2 \\ 0 \end{bmatrix}$. Notemos que siempre usamos los elementos resultantes del paso anterior, al igual que la factorización $LU$. Nuevamente sabemos que existe $W \in \mathbb{R}^{2\times 2}$ ortogonal tal que $W\tilde{x} = \tilde{y}$, y la embebemos junto a la identidad en una matriz de $n \times n$.

$$W_{13} = \begin{bmatrix} w_{11} & 0 & w_{12} & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ w_{21} & 0 & w_{22} & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

Multiplicamos a izquierda el resultado del paso anterior:

$$W_{13}W_{12}A = \begin{bmatrix} * & * & \dots & * \\ 0 & * & \dots & * \\ 0 & * & \dots & * \\ a_{41} & a_{42} & \dots & a_{4n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

Como vamos colocando un cero en cada paso, necesitamos barrer todas las posiciones debajo de la diagonal principal. El paso genérico se define para todo $i = 1 \dots n-1$, para todo $j = i+1 \dots n$. En el paso $ij$ colocamos un cero en la posición $ji$.

$$\tilde{x} = \begin{bmatrix} a_{ii}^{i-1} \\ a_{ji}^{i-1} \end{bmatrix} \qquad \tilde{y} = \begin{bmatrix} ||\tilde{x}||_2 \\ 0 \end{bmatrix} \qquad W_{ij} = \begin{bmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & w_{ii} & \dots & w_{ij} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \dots & w_{ji} & \dots & w_{jj} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{bmatrix}$$

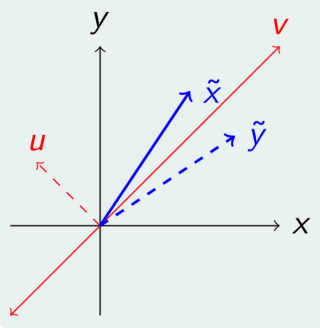
La factorización $QR$ completa se obtiene multiplicando $A$ por todas las matrices de rotación $W_{ij}$ (que como son ortogonales su inveresa es igual a su traspuesta):

$$W_{n-1n}W_{n-2n}W_{n-2n-1}\dots W_{1n}\dots W_{13}W_{12}A = R \iff A = \underbrace{W_{12}^t W_{13}^t \dots W_{1n}^t \dots W_{n-2n-1}^t W_{n-2n}^t W_{n-1n}^t}_{Q} R = QR$$

## Método Householder (reflexiones)

El método de Householder consiste en aplicar sucesivas reflexiones para colocar ceros debajo de la diagonal principal. Una reflexión es una transformación lineal que refleja todo vector respecto a un hiper-plano. Si bien una reflexión concreta se puede modelar como una rotación, donde el método de Givens construye una matriz de rotación que tiene configurada un ángulo de rotación fijo, Householder solo define el plano de reflexión, y por lo tanto cada vector será rotado un ángulo distinto, el que sea necesario para reflejar el vector respecto al plano. En una matriz de Householder lo que queda fijo es justamente el plano de reflexión.

### ¿Qué es una reflexión?




Una reflexión se define a partir de un hiper-plano. En el caso de $\mathbb{R}^2$ sería una recta. Sea $H \in \mathbb{R}^{2\times 2}$ una matriz ortogonal que refleja respecto a la dirección $v$.

- $Hv = v$ pues la propia dirección de reflexión se mantiene constante al aplicar la transformación $H$.
- $Hu = -u$ pues la dirección ortogonal a la dirección de reflexión se invierte al aplicar la transformación $H$ (se puede pensar como que se rota 180°).
- $H\tilde{x} = \tilde{y}$ para cualquier otro caso. El ángulo que se forma entre $\tilde{x}$ y $v$ es exactamente igual al que se forma entre $\tilde{y}$ y $v$.

### ¿Cómo se construye la matriz $H$?

Como $u$ y $v$ forman una base por ser vectores ortogonales, podemos escribir $\tilde{x}$ y su reflexión $\tilde{y}$ como combinación lineal de ellos.

$$\tilde{x} = \alpha v + \beta u \qquad \tilde{y} = \alpha v - \beta u$$

Buscamos $H, W \in \mathbb{R}^{2\times 2}$ tal que:

$$H\tilde{x} = \tilde{y} \iff H\tilde{x} = \alpha v - \beta u \iff H\tilde{x} = \underbrace{\alpha v + \beta u}_{I\tilde{x}} - \underbrace{2\beta u}_{W\tilde{x}} \iff H\tilde{x} = I\tilde{x} - W\tilde{x}$$

Donde $W\tilde{x} \overset{\text{def } \tilde{x}}{=} W(\alpha v + \beta u) = \alpha W v + \beta W u = 2\beta u \iff \begin{cases} Wv = 0 \\ Wu = 2u \end{cases}$

Asumiendo que $||u||_2 = 1$, definimos $P \in \mathbb{R}^{2\times 2}$ como $P = uu^t$.

- $P$ es simétrica: $P^t = P$
  $P^t = (uu^t)^t = (u^t)^t u^t = uu^t = P$
- $PP^t = P$
  $PP^t = PP = (uu^t)(uu^t) = u(u^t u)u^t = u\underbrace{||u||_2^2}_{=1}u^t = uu^t = P$

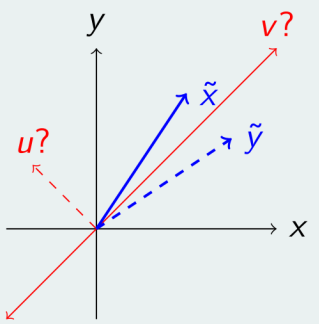
- $Pu = u$
  $Pu = (uu^t)u = u(u^t u) = u\underbrace{||u||_2^2}_{=1} = u$
- $Pv = 0$
  $Pv = (uu^t)v = u(u^t v) = 0$ pues $u \perp v \Rightarrow u^t v = 0$

Si definimos $W = 2P$ tenemos $H = I - 2P$. Veamos que $H\tilde{x} = \tilde{y}$.

$$H\tilde{x} = I\tilde{x} - 2P\tilde{x} = \alpha v + \beta u - 2P(\alpha v + \beta u) = \alpha v + \beta u - 2\alpha \underbrace{Pv}_{=0} - 2\beta \underbrace{Pu}_{u} = \alpha v - \beta u = \tilde{y}$$

Finalmente tenemos $H = I - 2uu^t$ con $H$ simétrica y ortogonal. Este mismo procedimiento se extiende a dimensiones más grandes.

**¿Cómo podemos obtener $H$ a partir de $\tilde{x}$ e $\tilde{y}$?**




Sean $\tilde{x}, \tilde{y} \in \mathbb{R}^n$ tal que $\tilde{x} \neq \tilde{y}$, $||\tilde{x}||_2 = ||\tilde{y}||_2$ (necesitamos que tengan la misma norma pues si no $H$ no podría ser ortogonal ya que estaría deformando el espacio además de reflejarlo). Por lo expuesto anteriormente existe $H \in \mathbb{R}^{n \times n}$ transformación de Householder tal que $H\tilde{x} = \tilde{y}$.

De forma similar al método de Givens en donde no queríamos buscar explícitamente el ángulo $\theta$ porque sabíamos exactamente a dónde queríamos rotar $\tilde{x}$, en el método de Householder pasa algo parecido. Queremos simplemente definir $H$ a partir de $\tilde{x}$ e $\tilde{y}$ (que luego vamos a elegir de forma conveniente para triangular la matriz).

$$v = \tilde{x} + \tilde{y} \qquad u = \frac{\tilde{x} - \tilde{y}}{||\tilde{x} - \tilde{y}||_2} \qquad H = I - 2uu^t = I - 2\frac{(\tilde{x} - \tilde{y})(\tilde{x} - \tilde{y})^t}{||\tilde{x} - \tilde{y}||_2^2}$$

**Ejemplo** $2 \times 2$

Sea $A \in \mathbb{R}^{2\times 2}$. Definimos $\tilde{x}, \tilde{y} \in \mathbb{R}^2$ tal que $\tilde{x} \neq \tilde{y}$, $||\tilde{x}||_2 = ||\tilde{y}||_2$.

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad \tilde{x} = \begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix} \qquad \tilde{y} = \begin{bmatrix} ||\tilde{x}||_2 \\ 0 \end{bmatrix}$$

Observación: si $a_{21} = 0$ entonces $\tilde{x} = \tilde{y}$, pero de ser así $A$ ya estaría triangulada y no es necesario hacer ninguna transformación.

Sabemos que existe una transformación de Householder $H \in \mathbb{R}^{2\times 2}$ tal que $H\tilde{x} = \tilde{y}$.

$$HA = \begin{bmatrix} ||\tilde{x}||_2 & * \\ 0 & * \end{bmatrix} = R$$

Como $H$ es ortogonal, su inversa es $H^t$.

$$HA = R \iff H^tHA = H^tR \Rightarrow A = QR$$

Donde $Q = H^t$ y $R$ es una matriz triangular superior.

**Caso general** $n \times n$

Sea $A \in \mathbb{R}^{n\times n}$. El procedimiento consiste en aplicar distintas transformaciones de Householder de manera recursiva. Primero se aplica a toda la matriz original $A$ para triangular la primer columna. Luego consideramos la submatriz formada desde la fila y columna 2 hasta la $n$ y repetimos el proceso, triangulando la primer columna de esta submatriz que en realidad es la segunda columna de la matriz de $n \times n$.

**Primer paso** Sean $\tilde{x}, \tilde{y} \in \mathbb{R}^n$ tal que:

$$\tilde{x} = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix} \qquad \tilde{y} = \begin{bmatrix} ||\tilde{x}||_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Existe una transformación de Householder $H_1 \in \mathbb{R}^{n\times n}$ tal que $H_1\tilde{x} = \tilde{y}$.

$$H_1 = I - 2u_1u_1^t \qquad u_1 = \frac{\tilde{x} - \tilde{y}}{||\tilde{x} - \tilde{y}||_2}$$

Obtenemos $A^1$ (la matriz resultante luego del primer paso de la factorización) multiplicando a izquierda por $H_1$.

$$H_1A^0 = \begin{bmatrix} ||\tilde{x}||_2 & a_{12}^1 & \dots & a_{1n}^1 \\ 0 & a_{22}^1 & \dots & a_{2n}^1 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & a_{n2}^1 & \dots & a_{nn}^1 \end{bmatrix} = A^1$$

**Segundo paso** Sean $\tilde{x}, \tilde{y} \in \mathbb{R}^{n-1}$ tal que:

$$\tilde{x} = \begin{bmatrix} a_{22}^1 \\ a_{32}^1 \\ \vdots \\ a_{n2}^1 \end{bmatrix} \qquad \tilde{y} = \begin{bmatrix} ||\tilde{x}||_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Existe una transformación de Householder $\tilde{H} \in \mathbb{R}^{(n-1)\times(n-1)}$ tal que $\tilde{H}\tilde{x} = \tilde{y}$.

$$\tilde{H} = I - 2u_2u_2^t \qquad u_2 = \frac{\tilde{x} - \tilde{y}}{||\tilde{x} - \tilde{y}||_2} \in \mathbb{R}^{n-1}$$

Embebemos $\tilde{H}$ adentro de $H_2 \in \mathbb{R}^{n\times n}$.

$$H_2 = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{H} \end{bmatrix}$$

Obtenemos $A^2$ multiplicando a izquierda por $H_2$.

$$H_2 A^1 = \begin{bmatrix} a_{11}^1 & a_{12}^1 & a_{13}^1 & \dots & a_{1n}^1 \\ 0 & ||\tilde{x}||_2 & a_{23}^2 & \dots & a_{2n}^2 \\ 0 & 0 & a_{33}^2 & \dots & a_{3n}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & a_{n3}^2 & \dots & a_{nn}^2 \end{bmatrix} = A^2$$

**Paso $i$-ésimo** En el paso $i$-ésimo trabajamos con una transformación de Householder de $\mathbb{R}^{(n-i+1)\times(n-i+1)}$ que triangula la columna $i$-ésima, que a su vez es la primer columna de la submatriz formada desde la fila y columna $i$ hasta la $n$.

Sean $\tilde{x}, \tilde{y} \in \mathbb{R}^{n-i+1}$ tal que:

$$\tilde{x} = \begin{bmatrix} a_{ii}^{i-1} \\ a_{i+1i}^{i-1} \\ \vdots \\ a_{ni}^{i-1} \end{bmatrix} \qquad \tilde{y} = \begin{bmatrix} ||\tilde{x}||_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Existe una transformación de Householder $\tilde{H} \in \mathbb{R}^{(n-i+1)\times(n-i+1)}$ tal que $\tilde{H}\tilde{x} = \tilde{y}$.

$$\tilde{H} = I - 2u_i u_i^t \qquad u_i = \frac{\tilde{x} - \tilde{y}}{||\tilde{x} - \tilde{y}||_2} \in \mathbb{R}^{n-i+1}$$

Embebemos $\tilde{H}$ adentro de $H_i \in \mathbb{R}^{n\times n}$.

$$H_i = \begin{bmatrix} I & 0 \\ 0 & \tilde{H} \end{bmatrix} \qquad I \in \mathbb{R}^{(i-1)\times(i-1)}$$

Obtenemos $A^i$ multiplicando a izquierda por $H_i$.

$$H_i A^{i-1} = \begin{bmatrix} a_{11}^{i-1} & a_{12}^{i-1} & \dots & a_{1i}^{i-1} & a_{1i+1}^{i-1} & \dots & a_{1n}^{i-1} \\ 0 & a_{22}^{i-1} & \dots & a_{2i}^{i-1} & a_{2i+1}^{i-1} & \dots & a_{2n}^{i-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & ||\tilde{x}||_2 & a_{ii+1}^{i} & \dots & a_{in}^{i} \\ 0 & 0 & \dots & 0 & a_{i+1i+1}^{i} & \dots & a_{i+1n}^{i} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & a_{ni+1}^{i} & \dots & a_{nn}^{i} \end{bmatrix} = A^i$$

**Último paso** Luego de aplicar todas las $n-1$ transformaciones de Householder obtenemos la factorización $QR$. Recordemos que todas las $H_i$ son ortogonales y entonces su inversa es su traspuesta, y además el producto de todas las $H_i$ también es una matriz ortogonal.

$$H_{n-1} \dots H_1 A = R \iff A = \underbrace{H_1^t \dots H_{n-1}^t}_{Q} R = QR$$

### Complejidad

El método de Givens tiene un costo de $O(4/3n^3)$.
El método de Householder tiene un costo de $O(2/3n^3)$.

Como el método de Givens coloca un solo cero en cada paso, si la matriz a triangular es rala (ya tiene muchos ceros), Givens puede ser más eficiente que Householder ya que se puede selectivamente solo realizar los pasos para colocar los ceros donde haga falta y evitar pasos redundantes.

### Unicidad

Sea $A \in \mathbb{R}^{n \times n}$ inversible. La factorización $A = QR$ es única si pedimos que $r_{ii} > 0$ para todo $i = 1 \dots n$.

## Autovalores

### Definición

Sea $A \in \mathbb{C}^{n \times n}$. $\lambda \in \mathbb{C}$ es un autovalor de $A$ sii existe $v \in \mathbb{C}^n$, $v \neq 0$ tal que:

$$Av = \lambda v$$

Decimos que $v$ es un autovector asociado al autovalor $\lambda$. Para encontrar los autovalores podemos desarrollar la definición de la siguiente manera:

$$Av = \lambda v \iff Av - \lambda v = 0 \iff (A - \lambda I)v = 0$$

Esto nos dice que $v \in Nu(A - \lambda I)$, y como $v \neq 0$ necesariamente vale que:

$$Nu(A - \lambda I) \neq \{0\} \iff A - \lambda I \text{ no inversible} \iff det(A - \lambda I) = 0$$

Al desarrollar la fórmula del determinante obtenemos el **polinomio característico** de la matriz: $P(\lambda) = det(A - \lambda I)$. Luego $\lambda$ es autovalor de $A$ sii $\lambda$ es raíz de $P(\lambda)$.

Ya sabemos encontrar los autovalores. ¿Cómo podemos obtener los autovectores? Para cada autovalor $\lambda$ volvemos a la definición original y reemplazamos el autovalor en el sistema dado por $(A - \lambda I)v = 0$. Luego resolvemos ese sistema homogéneo para encontrar $v$.

### Autoespacios y multiplicidad

**Los autovectores son las direcciones que se preservan al aplicar la transformación lineal $A$.** El autovalor asociado codifica cuánto se estira o contrae en la dirección del autovector. Por eso, si $v$ es un autovector asociado a $\lambda$, entonces $\alpha v$ para cualquier $\alpha \in \mathbb{R}$ también es un autovector asociado a $\lambda$.

Definimos el **autoespacio** asociado al autovalor $\lambda$ como el subespacio $S_\lambda = Nu(A - \lambda I)$. Todos los autovectores asociados a $\lambda$ están en $S_\lambda$. La multiplicidad geométrica del autovalor $\lambda$ está dada por la dimensión del autoespacio: $m_G(\lambda) = dim(S_\lambda)$.

Por otro lado, el polinomio característico $P(\lambda)$ tiene $n$ raíces contadas con multiplicidad. No obstante, pueden haber raíces complejas no reales y raíces múltiples. Si $\lambda \notin \mathbb{R}$ entonces no hay autoespacio en $\mathbb{R}^n$. Llamamos $m_A(\lambda)$ a la multiplicidad algebraica del autovalor $\lambda$.

La relación entre la multiplicidad algebraica y geométrica es la siguiente:

$$1 \leq m_G(\lambda) \leq m_A(\lambda)$$

Si sucede que $m_G(\lambda) = m_A(\lambda)$ para todos los autovalores, entonces los autovectores forman una base de $\mathbb{R}^n$.

## Radio espectral

Se define el radio espectral de $A$ como:

$$\rho(A) = \max\{|\lambda| : \lambda \text{ autovalor de } A\}$$

El radio espectral cuantifica la máxima deformación que realiza la transformación $A$ en alguna dirección. Algunos casos de uso del radio espectral son:

- Si tenemos un esquema iterativo con una matriz de iteración $T$, el esquema converge para cualquier $x_0$ inicial sii $\rho(T) < 1$.
- La norma matricial de orden 2 de una matriz $A$ coincide con la raíz del radio espectral de $A^tA$: $||A||_2 = \sqrt{\rho(A^tA)}$.

## Discos de Gershgorin

Los discos de Gershgorin nos permiten acotar los autovalores adentro de discos en el plano complejo.

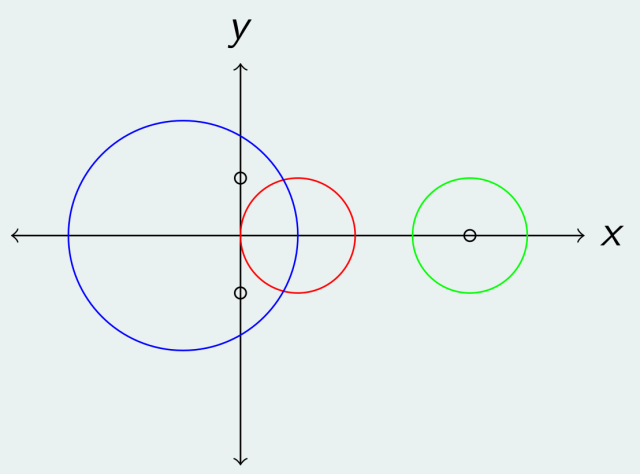
Sea $A \in \mathbb{C}^{n\times n}$ y $r_i = \sum_{j=1, j\neq i}^{n} |a_{ij}|$ para $i = 1 \dots n$. Definimos el disco de Gershgorin como:

$$D_i = \{x \in \mathbb{C} : |x - a_{ii}| \leq r_i\} \text{ para } i = 1 \dots n$$

Luego, si $\lambda$ es autovalor de $A$ entonces $\lambda \in D_i$ para algún $i \in [1, n]$. Todos los autovalores están adentro de algún disco, pero puede pasar que haya discos en donde no hay ningún autovalor.

Si $M = D_{i_1} \cup D_{i_2} \cup \dots \cup D_{i_m}$ es disjunto respecto al resto de los discos entonces en $M$ hay exactamente $m$ autovalores de $A$ contados con multiplicidad.

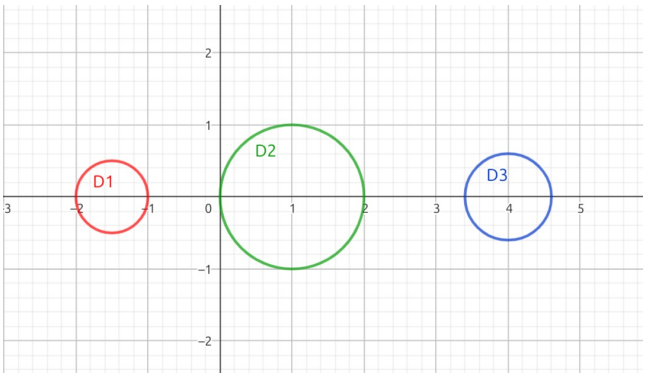
### Ejemplos




Sean $D_1$ (azul), $D_2$ (rojo) y $D_3$ (verde) los discos de alguna matriz $A \in \mathbb{C}^{3\times 3}$.

En $M_1 = D_1 \cup D_2$ tenemos 2 autovalores ya que $M_1 \overset{D}{\cup} D_3$.

En $M_2 = D_3$ tenemos 1 autovalor ya que $M_2 \overset{D}{\cup} (D_1 \cup D_2)$.




Si fuese el caso donde todos los $D_i$ son disjuntos, entonces en cada disco habría un único autovalor, y por lo tanto la multiplicidad algebraica de cada autovalor sería 1, dando lugar a $n$ autovalores distintos.

## Matrices semejantes

$A, B \in \mathbb{C}^{n\times n}$ son matrices semejantes si existe $P \in \mathbb{C}^{n\times n}$ matriz inversible tal que:

$$A = PBP^{-1}$$

Lo interesante es que si $A$ y $B$ son semejantes entonces **tienen los mismos autovalores**. La noción de semejanza viene a raíz de que la transformación que realiza $A$ en la base canónica es la misma que realiza $B$ en otra base definida por $P$. La transformación $PBP^{-1}$ en esencia hace un cambio de base, aplica la transformación $B$ y luego vuelve a la base canónica.

Cuando $A$ es semejante a una diagonal decimos que $A$ es diagonalizable:

$$A = PDP^{-1}$$

**$A$ es diagonalizable sii sus autovectores forman una base.** Este resultado es importante ya que significa que la transformación lineal definida por $A$ la podemos interpretar como deformaciones (estirar/contraer) realizadas sobre cada uno de los ejes canónicos. La matriz $P$ se encarga de hacer el cambio de base hacia/desde la base canónica.

Verificar si $A$ es diagonalizable puede ser muy costoso. Nos gustaría tener una forma más fácil de saber si $A$ tiene una base de autovectores.

## Propiedades

A efectos prácticos de la materia nos enfocamos en matrices con coeficientes reales. Sea $A \in \mathbb{R}^{n\times n}$.

- Si $\lambda$ es autovalor de $A$ con autovector asociado $v$, entonces $\lambda^k$ es autovalor de $A^k$ con el mismo autovector asociado $v$. Volviendo a la interpretación geométrica, aplicar $k$ veces la transformación lineal va a estirar/contraer por $\lambda^k$ a la dirección $v$ que es preservada por $A$.
- Si $A$ es una matriz ortogonal entonces sus autovalores son $\lambda = \pm 1$. Recordemos que las matrices ortogonales son transformaciones que solo rotan/reflejan sin deformar el espacio.
- Si $\lambda$ es autovalor de $A$ entonces también lo es de $A^t$. Esto se deduce a partir de que $A$ y $A^t$ tienen el mismo polinomio característico.
- Si todos los autovalores de $A$ están en $\mathbb{R}$:
  - Todos los autovectores también están en $\mathbb{R}^n$, pues se obtienen resolviendo un sistema definido completamente en $\mathbb{R}$ mediante operaciones cerradas en $\mathbb{R}$.

  - Si además todos los autovalores son distintos entonces los autovectores son linealmente independientes. Luego forman una base de $\mathbb{R}^n$.
  - Existen $Q, T \in \mathbb{R}^{n \times n}$ con $Q$ ortogonal y $T$ triangular superior tal que $A$ es ortogonalmente semejante a $T$. Es decir: $A = QTQ^t$.

- Si $A$ es simétrica:
  - Todos sus autovalores están en $\mathbb{R}$.
  - Si $\lambda_1 \neq \lambda_2$ son autovalores de $A$ entonces sus autovectores asociados $v_1$ y $v_2$ son ortogonales entre sí.
  - Existen $Q, D \in \mathbb{R}^{n \times n}$ con $Q$ ortogonal y $D$ diagonal tal que $A = QDQ^t$. Las columnas de $Q$ son los autovectores de $A$ y $D$ contiene todos los autovalores de $A$.
  - Los autovectores forman una base ortonormal.

## Método de la potencia

Para poder obtener los autovalores y autovectores asociados podemos utilizar un método iterativo llamado método de la potencia.

Sean $A \in \mathbb{R}^{n \times n}$, $\lambda_1 \dots \lambda_n \in \mathbb{R}$ sus $n$ autovalores y $v_1 \dots v_n \in \mathbb{R}^n$ los autovectores asociados que forman una base.

Sin pérdida de generalidad supongamos que ordenamos los autovalores tal que $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$.

A partir de un vector arbitrario inicial $x_0 \in \mathbb{R}^n$ tal que $||x_0||_2 = 1$ definimos la sucesión $\{x_k\}$ como:

$$x_k = \frac{Ax_{k-1}}{||Ax_{k-1}||_2}$$

Dicha sucesión converge al autovector $v_1$. Para obtener el autovalor realizamos la siguiente cuenta:

$$Av_1 = \lambda_1 v_1 \iff v_1^t A v_1 = \lambda_1 v_1^t v_1 \iff v_1^t A v_1 = \lambda_1 \underbrace{||v_1||_2^2}_{=1} \iff v_1^t A v_1 = \lambda_1$$

Dado que los autovectores forman una base, si consideramos a $x_k$ como una combinación lineal de los autovectores, la idea del método es que en cada iteración se "acentúa" la dirección del autovector $v_1$ pues $\lambda_1$ es el autovalor más grande en módulo. Al normalizar $x_k$ para la siguiente iteración, los coeficientes que acompañan a los autovectores $v_2 \dots v_n$ tienden a 0. Después de una cantidad lo suficientemente grande de iteraciones y gracias a la aritmética finita el único coeficiente que sobrevive es el que acompaña a $v_1$.

Generalmente se configura un máximo de iteraciones y también se puede incluir una condición de corte temprano que consiste en ver si la distancia (medida con alguna norma) entre $x_k$ y $x_{k-1}$ es menor que algún umbral. En tal caso se considera $x_k$ como suficientemente bueno y no es necesario realizar más iteraciones.

Hay que mencionar que existe una probabilidad no nula (pero muy baja) de que el $x_0$ inicial elegido de manera arbitraria justo tenga un coeficiente $\approx 0$ acompañando al autovector $v_1$ si consideramos a $x_0$ como combinación lineal de los autovectores. En este caso el método puede no converger o necesitar una cantidad excesivamente grande de iteraciones para hacerlo. Una solución consiste simplemente en reintentar el método con un nuevo $x_0$ inicial.

## Método de deflación

Utilizando el método de la potencia encontramos $\lambda_1$ y $v_1$. ¿Cómo podemos encontrar todos los autovalores y autovectores? Como ya sabemos obtener el autovalor más grande en módulo, el método de deflación consiste

en obtener una nueva matriz donde el autovalor $\lambda_1$ pasó a ser el más chico y repetir así el método de la potencia sobre esta nueva matriz, hayando el nuevo autovalor más grande en módulo que será $\lambda_2$.

Sean $A \in \mathbb{R}^{n\times n}$, $\lambda_1 \in \mathbb{R}$ autovalor con autovector asociado $v_1 \in \mathbb{R}^n$ tal que $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$..

Sea $H \in \mathbb{R}^{n\times n}$ matriz ortogonal tal que $Hv_1 = e_1$.

$$HAH^t = \begin{bmatrix} \lambda_1 & a^t \\ 0 & \tilde{A} \end{bmatrix}$$

Notemos que $A$ y $HAH^t$ tienen los mismos autovalores. Si planteamos el polinomio característico de $HAH^t$ obtenemos:

$$det(HAH^t - \lambda I) = (\lambda_1 - \lambda) * det(\tilde{A} - \lambda I)$$

Entonces el resto de los autovalores de $A$: $\lambda_2 \dots \lambda_n$ son autovalores de $\tilde{A}$. Para encontrar el resto de los autovalores (o la cantidad que se necesite) simplemente repetimos el método de la potencia sobre esta nueva matriz $\tilde{A}$ para encontrar $\lambda_2$. Luego volvemos a aplicar la deflación y así sucesivamente.

# SVD

## Definición

Sean $A \in \mathbb{R}^{m\times n}$ y $r = rango(A)$. Existen $U \in \mathbb{R}^{m\times m}$ ortogonal, $V \in \mathbb{R}^{n\times n}$ ortogonal y $\Sigma \in \mathbb{R}^{m\times n}$ pseudo-diagonal tales que:

$$A = U\Sigma V^t$$

La descomposición en valores singulares (SVD) utiliza 3 matrices muy específicas para descomponer $A$.

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \sigma_r & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 \end{bmatrix} \qquad U = \begin{bmatrix} | & & | \\ u_1 & \dots & u_m \\ | & & | \end{bmatrix} \qquad V = \begin{bmatrix} | & & | \\ v_1 & \dots & v_n \\ | & & | \end{bmatrix}$$

Realizamos el siguiente desarrollo para encontrar una relación entre las distintas matrices y los valores singulares $\sigma_1 \dots \sigma_r$.

$$A = U\Sigma V^t \iff AV = U\Sigma \iff \begin{cases} Av_i = \sigma_i u_i & i = 1 \dots r \quad (1) \\ Av_i = 0 & i = r+1 \dots n \quad (2) \end{cases}$$

$$A = U\Sigma V^t \iff U^t A = \Sigma V^t \iff A^t U = V\Sigma^t \iff \begin{cases} A^t u_i = \sigma_i v_i & i = 1 \dots r \quad (3) \\ A^t u_i = 0 & i = r+1 \dots m \quad (4) \end{cases}$$

**¿Quiénes son los $v_i$?**

$$Av_i = \sigma_i u_i \iff A^tAv_i = \sigma_i A^t u_i \overset{(3)}{\iff} A^tAv_i = \sigma_i\sigma_i v_i \iff A^tAv_i = \sigma_i^2 v_i \quad \text{para todo } i = 1 \dots r$$

$$Av_i = 0 \iff A^tAv_i = 0 \quad \text{para todo } i = r+1 \dots n$$

Deducimos entonces que los $v_i$ son los autovectores de $A^tA$.

- $A^tA$ es una matriz simétrica entonces existe una base ortonormal de autovectores.
- $A^tA$ es semi definida positiva entonces sus autovalores son $\geq 0$ pues $0 \leq v^tA^tAv = v^t\lambda v = \lambda||v||_2^2 \overset{v\neq 0}{\Longrightarrow} 0 \leq \lambda$.
- $rango(A^tA) = rango(A) = r$ entonces hay $r$ autovalores no nulos y 0 es autovalor con multiplicidad $n - r$.

Sean $\lambda_1 \geq \dots \geq \lambda_r > 0$ los autovalores no nulos de $A^tA$ con $v_1 \dots v_r$ sus autovectores asociados. Sean $v_{r+1} \dots v_n$ los autovectores asociados al autovalor nulo.

Definimos los valores singulares como $\sigma_i = \sqrt{\lambda_i} > 0$ para todo $i = 1 \dots r$, donde $\lambda_i$ son los autovalores de $A^tA$.

Los autovectores $v_1 \dots v_n$ forman un conjunto ortonormal y son los candidatos a ser las columnas de la matriz $V$.

**¿Quiénes son los $u_i$?**

A partir de las definiciones previas y utilizando la ecuación (1) definimos:

$$Av_i = \sigma_i u_i \overset{\sigma_i>0}{\iff} u_i = \frac{1}{\sigma_i}Av_i \quad \text{para todo } i = 1 \dots r$$

Como los $u_i$ son las columnas de $U$ matriz ortogonal, tenemos que verificar que $u_1 \dots u_r$ sea un conjunto ortonormal.

$$u_i^tu_j = \frac{1}{\sigma_i}(Av_i)^t\frac{1}{\sigma_j}Av_j = \frac{1}{\sigma_i\sigma_j}v_i^tA^tAv_j \overset{\overset{v_j \text{ autovector de } A^tA}{\uparrow}}{=} \frac{\lambda_j}{\sigma_i\sigma_j}v_i^tv_j \overset{\overset{v_i\perp v_j}{\uparrow}}{=} 0$$

$$u_i^tu_i = \frac{1}{\sigma_i}(Av_i)^t\frac{1}{\sigma_i}Av_i = \frac{1}{\sigma_i\sigma_i}v_i^tA^tAv_i \overset{\overset{v_i \text{ autovector de } A^tA}{\uparrow}}{=} \frac{\lambda_i}{\sigma_i\sigma_i}v_i^tv_i \overset{\overset{||v_i||_2^2=1}{\uparrow}}{=} \frac{\lambda_i}{\sigma_i\sigma_i} \overset{\overset{\lambda_i=\sigma_i^2}{\uparrow}}{=} 1$$

Los $r$ vectores $u_1 \dots u_r$ pertenecen a $Im(A)$ que tiene $dim(Im(A)) = r$. Como son un conjunto ortonormal entonces son una base ortonormal de $Im(A)$.

El resto de los $u_{r+1} \dots u_m$ tienen que pertenecer al $Nu(A^t)$ por la ecuación (4). $Im(A)$ está en suma directa con su ortogonal: $Im(A) \oplus Im(A)^\perp = \mathbb{R}^m$. Como $Im(A)^\perp = Nu(A^t)$, basta tomar $u_{r+1} \dots u_m$ como una base ortonormal de $Nu(A^t)$ y así el conjunto $u_1 \dots u_m$ forma una base ortonormal de todo $\mathbb{R}^m$ y son los candidatos a ser las columnas de la matriz ortogonal $U$.

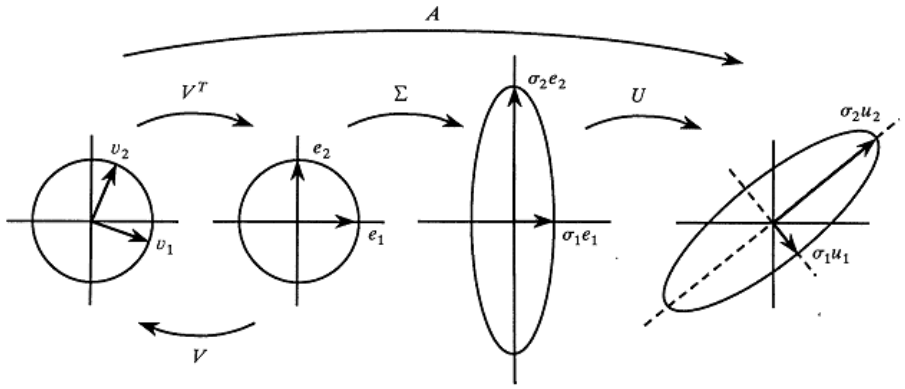
**Conclusión**

- $\sigma_i = \sqrt{\lambda_i} > 0$ con $\lambda_i$ autovalor no nulo de $A^tA$.
- $v_1 \dots v_n$ base ortonormal de autovectores de $A^tA$.
- $u_1 \dots u_r$ base ortonormal de $Im(A)$ con $u_i = \frac{1}{\sigma_i} A v_i$.
- $u_{r+1} \dots u_m$ base ortonormal de $Im(A)^\perp = Nu(A^t)$.

Es fácil verificar que si construimos $U$, $\Sigma$ y $V$ a partir de estas definiciones se cumplen las 4 ecuaciones planteadas anteriormente y las matrices forman la descomposición SVD de $A = U\Sigma V^t$. Además, se puede hacer un análisis similar para concluir que $u_1 \dots u_m$ es una base ortonormal de autovectores de $AA^t$.

## Interpretación geométrica

La descomposición SVD tiene una interpretación geométrica muy interesante. Cada matriz $U$, $\Sigma$ y $V$ cumple una función específica.




La matriz $V^t$ realiza un cambio de base mediante una rotación, en donde los autovectores $v_i$ quedan alineados con los vectores canónicos $e_i$. Luego la matriz $\Sigma$ expande o contrae el espacio sobre cada uno de los vectores canónicos. Finalmente la matriz $U$ realiza otro cambio de base mediante una rotación.

## Propiedades

- $||A||_2 = \sigma_1$
  La norma 2 cuantifica la máxima deformación del espacio realizada por la transformación $A$. El hecho de que sea igual a $\sigma_1$ se corresponde con la interpretación geométrica, en donde dijimos que luego de rotar con $V^t$, la matriz $\Sigma$ expande o contrae el espacio sobre cada dirección canónica acorde a los valores singulares $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$.
- $\kappa_2(A) = \frac{\sigma_1}{\sigma_n}$
- $||A||_F = \sqrt{\sigma_1^2 + \dots + \sigma_r^2}$

## Aplicaciones

La descomposición SVD cuantifica en los valores singulares las características de la matriz original, ya sea una transformación lineal o una matriz que representa un dataset. ¿Qué pasa si descartamos algunos valores singulares? Esto es justamente lo que hace el algoritmo de PCA. Al quedarnos solo con los valores singulares más grandes, podemos reducir la dimensión de la matriz original y aún así preservar las características más importantes. Reducir la dimensión significa menos espacio de memoria y/o tiempo de cómputo más rápido.

# CML

## Introducción

Dado un conjunto de pares ordenados $(x_i, y_i)$ para $i = 1 \dots m$ buscamos una función $f(x)$ perteneciente a una familia de funciones $\mathcal{F}$ tal que mejor aproxime los datos.

Ahora nuestra matriz ya no representa necesariamente una transformación lineal, si no que podría ser un dataset de observaciones. Los valores de $y_i$ podrían representar alguna medición y su $x_i$ asociado el tiempo de dicha medición.

A diferencia de los métodos de interpolación, CML no garantiza que $f(x)$ pase por todos los puntos $x_1 \dots x_m$. En cambio buscamos una función que en su comportamiento general se ajuste lo mejor posible a los datos, con la intención de poder evaluar la función en datos desconocidos y obtener una predicción lo más acertada posible.

## ¿Cómo podemos medir qué tan bien $f(x)$ aproxima los datos?

Como métrica base siempre consideramos el error cometido por la función al evaluarla en los puntos conocidos: $|f(x_i) - y_i|$, es decir la diferencia absoluta entre $f(x_i)$ y el valor real observado $y_i$.

**Tomamos la función que minimice el máximo error.**

$$\min_{f \in \mathcal{F}} \max_{i=1 \dots m} |f(x_i) - y_i|$$

El problema de esta métrica es que la solución es muy sensible a datos atípicos (outliers).

**Tomamos la función que minimice la suma de las diferencias.**

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{m} |f(x_i) - y_i|$$

**Tomamos la función que minimice la suma de las diferencias al cuadrado.**

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{m} (f(x_i) - y_i)^2$$

Este es el método de cuadrados mínimos y el que vamos a utilizar ya que posee propiedades prácticas que facilitan obtener la solución.

## Definición

Dado un conjunto de funciones $\{\phi_1 \dots \phi_n\}$ **linealmente independientes** definimos la familia de funciones como:

$$\mathcal{F} = \{f(x) = \sum_{j=1}^{n} c_j \phi_j(x)\}$$

El problema de cuadrados mínimos lineales consiste en encontrar los coeficientes $c_1 \dots c_n$ tales que $f(x)$ mejor aproxime los datos:

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{m} (f(x_i) - y_i)^2 = \min_{c_1 \dots c_n} \sum_{i=1}^{m} (\sum_{j=1}^{n} c_j \phi_j(x_i) - y_i)^2$$

Notemos que podemos reescribir el problema de forma matricial. Sean $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$ y $b \in \mathbb{R}^m$ tales que:

$$A = \begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) & \dots & \phi_n(x_1) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_1(x_m) & \phi_2(x_m) & \dots & \phi_n(x_m) \end{bmatrix} \qquad x = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} \qquad b = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

$$Ax - b = \begin{bmatrix} \sum_{j=1}^n c_j \phi_j(x_1) - y_1 \\ \vdots \\ \sum_{j=1}^n c_j \phi_j(x_m) - y_m \end{bmatrix}$$

$$||Ax - b||_2^2 = \sum_{i=1}^m (\sum_{j=1}^n c_j \phi_j(x_i) - y_i)^2$$

Luego el problema de CML consiste en buscar $x$ tal que:

$$\min_{x \in \mathbb{R}^n} ||Ax - b||_2^2$$

## Existencia de la solución

Para encontrar la solución nos apoyamos en que $Im(A) \oplus Nu(A^t) = \mathbb{R}^m$. Esto significa que cualquier vector de $\mathbb{R}^m$ podemos escribirlo como una suma directa de 2 vectores, uno de $Im(A)$ y otro de $Nu(A^t)$. En particular vamos a descomponer el vector $b \in \mathbb{R}^m$ como $b = b^1 + b^2$, donde $b^1 \in Im(A)$ y $b^2 \in Nu(A^t)$.

$$\min_{x \in \mathbb{R}^n} ||Ax - b||_2^2 = \min_{x \in \mathbb{R}^n} ||Ax - b^1 - b^2||_2^2$$

$Ax - b^1 \in Im(A)$ mientras que $b^2 \in Nu(A^t) = Im(A)^\perp$. Entonces $(Ax - b^1) \perp b^2$ y por Pitágoras tenemos que:

$$\min_{x \in \mathbb{R}^n} ||Ax - b^1 - b^2||_2^2 = \min_{x \in \mathbb{R}^n} (||Ax - b^1||_2^2 + ||b^2||_2^2)$$

Notemos que $||b^2||_2^2$ es constante, por lo tanto el problema se reduce a buscar $x$ tal que:

$$\min_{x \in \mathbb{R}^n} ||Ax - b^1||_2^2$$

Como $b^1 \in Im(A)$ podemos asegurar que **siempre** existe $x^* \in \mathbb{R}^n$ tal que $Ax^* = b^1$. Y en efecto $x^*$ minimiza la norma pues $||Ax^* - b^1||_2^2 = 0$ es lo mínimo que puede valor una norma.

Luego $x^*$ es la solución al problema de CML. Recordemos que $x^* = (c_1 \dots c_n)$ contiene los coeficientes de la función $f(x) = c_1\phi_1(x) + \dots + c_n\phi_n(x)$ que planteamos para ajustarse a los datos.

## Interpretación geométrica

La noción de buscar una función que se ajuste a los datos se interpreta como buscar la **proyección ortogonal** de $b$ sobre el subespacio $Im(A) \subseteq \mathbb{R}^m$. La proyección ortogonal tiene la mínima distancia euclideana entre $b$ y algún punto de $Im(A)$. No queremos el punto que está en $Im(A)$ sino su pre-imagen, es decir el vector $x^* \in \mathbb{R}^n$ que genera el punto $Ax^*$ (pueden haber varios). De esta forma encontramos los coeficientes $x^*$ que nos dan una función que mejor se ajusta a los datos pues $Ax^*$ es lo más cerca que podemos llegar de $b$ (las observaciones de nuestro dataset).

### Unicidad

Concluimos que $Ax^* = b^1$ es la solución al problema de CML. Como estamos escribiendo a $b^1$ como una combinación lineal de las columnas de $A$, la solución es única sii $A$ tiene rango máximo, o equivalentemente, las columnas de $A$ son linealmente independientes.

### Ecuaciones normales

Si bien logramos caracterizar la solución al problema de CML, nos gustaría poder plantearlo en términos de los datos originales $A$ y $b$.

Recordemos que $b = b^1 + b^2$ con $b^1 \in Im(A)$ y $b^2 \in Nu(A^t)$.

$$b = b^1 + b^2 \iff b^1 = b - b^2$$

Sea $x^*$ solución al problema de CML.

$$Ax^* = b^1 \iff Ax^* = b - b^2 \iff b^2 = b - Ax^*$$

Como $b^2 \in Nu(A^t)$ tenemos que:

$$A^t b^2 = A^t(b - Ax^*) = 0 \iff \underbrace{A^t A x^* = A^t b}_{\text{ecuaciones normales}}$$

De esta forma logramos caracterizar el problema en función de los datos originales. Notemos que la matriz $A^tA$ es simétrica semi definida positiva, o definida positiva si $A$ tiene rango máximo. Podemos resolver el sistema planteado con cualquiera de las técnicas vistas como la factorización de Cholesky.

No obstante, la matriz $A^tA$ puede estar mal condicionada y generar soluciones inestables. Pequeños cambios en $b$ producen grandes cambios en la solución $x^*$. Para intentar remediar esto podemos buscar otra formulación de la solución mediante la factorización $QR$ o $SVD$ en donde no necesitemos usar la matriz $A^tA$.

### Error

### Resolución usando $QR$

### Resolución usando $SVD$

## Métodos Iterativos

### Motivación

Sean $A \in \mathbb{R}^{n \times n}$ y $b \in \mathbb{R}^n$. Buscamos $x \in \mathbb{R}^n$ tal que $Ax = b$.

A diferencia de los métodos exactos que en un número finito de pasos obtienen una solución al sistema (Eliminación Gaussiana, LU, QR), los métodos iterativos llegan a una solución a través de una sucesión $\{x^{(k)}\}$ que converge hacia la solución del sistema.

Este enfoque es especialmente útil cuando la matriz del sistema es muy grande. Intentar obtener una solución exacta puede ser prohibitivo computacionalmente. Si el problema admite soluciones con un cierto margen de error, los métodos iterativos permiten obtener una solución mucho más rápido.

## Esquema básico

Dado un $x^{(0)} \in \mathbb{R}^n$ inicial queremos iterativamente ir refinando este vector hasta obtener una solución al sistema $Ax = b$. Para esto planteamos una sucesión $\{x^{(k)}\}$ definida con el siguiente esquema:

$$x^{(k+1)} = Tx^{(k)} + c$$

En donde $T \in \mathbb{R}^{n \times n}$ es la **matriz de iteración** y $c \in \mathbb{R}^n$ algún vector constante.

## Convergencia

Si $x^* \in \mathbb{R}^n$ es la solución del sistema $Ax = b$, decimos que el método converge si:

$$\lim_{k \to \infty} x^{(k)} = x^*$$

¿Cómo podemos determinar la convergencia a partir de la matriz de iteración $T$? Por suerte tenemos el siguiente resultado principal:

$$\{x^{(k)}\} \text{ converge para cualquier } x^{(0)} \text{ inicial } \iff \rho(T) < 1$$

Recordemos que el radio espectral está definido como $\rho(T) = \max\{|\lambda| : \lambda \text{ autovalor de } T\}$. Si por alguna razón ya estábamos calculando los autovalores de $T$ o tenemos alguna otra propiedad de $T$ que nos permita acotar el máximo autovalor entonces resultaría trivial verificar la convergencia.

Otra opción para determinar si el método converge es utilizar la siguiente propiedad: $\rho(T) < ||T||$ donde $||\cdot||$ es cualquier norma matricial inducida. Notemos que las normas $||\cdot||_\infty$ y $||\cdot||_1$ se pueden calcular sumando filas o columnas lo cual tiene un costo muy bajo. Combinando los 2 teoremas:

$$||T|| < 1 \text{ para alguna norma inducida } \implies \{x^{(k)}\} \text{ converge para cualquier } x^{(0)}$$

## Cota del error

Como mencionamos antes, tenemos una forma de acotar el error para una determinada cantidad de iteraciones del método. Esto nos permite determinar la cantidad de iteraciones que necesitamos para garantizar que la solución sea lo suficientemente buena para el contexto de uso.

Sea $T \in \mathbb{R}^{n \times n}$ la matriz de iteración del método tal que $||T|| < 1$ para alguna norma inducida. El error de la solución obtenida está acotada por:

$$||x - x^{(k)}|| \leq \frac{||T||^k}{1 - ||T||}(x^{(1)} - x^{(0)})$$

## Métodos clásicos

Existen 2 métodos clásicos llamados método de **Jacobi** y método de **Gauss-Seidel**. Ambos se construyen a partir de la siguiente descomposición.

Sea $A \in \mathbb{R}^{n \times n}$ tal que $a_{ii} \neq 0$ para todo $i = 1 \dots n$. Descomponemos $A = D - L - U$ donde:

- $D \in \mathbb{R}^{n \times n}$ es una matriz diagonal que contiene todos los elementos de la diagonal de $A$. Notemos que al pedir que $A$ no tenga elementos nulos en su diagonal la matriz $D$ resulta inversible.
- $L \in \mathbb{R}^{n \times n}$ es una matriz estrictamente triangular inferior (su diagonal son todos ceros) que contiene todos los elementos de $A$ que están **debajo** de la diagonal y con el signo opuesto.

- $U \in \mathbb{R}^{n \times n}$ es una matriz análoga a $L$ pero estrictamente triangular superior que contiene todos los elementos de $A$ que están **arriba** de la diagonal y con el signo opuesto.

En cada iteración ambos métodos van refinando cada coordenada de la solución una por una. La diferencia es que Jacobi refina todas las coordenadas utilizando exclusivamente los valores calculados en el paso anterior, mientras que Gauss-Seidel utiliza las coordenadas ya calculadas en el mismo paso.

Utilizando la descomposición $A = D - L - U$ cada método plantea un esquema de iteración distinto. Veamos las particularidades.

**Jacobi**

La coordenada $i$-ésima de la solución en el paso $k+1$ de iteración se define de la siguiente manera:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \Big( b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j^{(k)} \Big)$$

Si llevamos esta expresión a una forma matricial obtenemos el esquema iterativo de Jacobi:

$$\begin{aligned} x^{(k+1)} &= D^{-1}(b + (L+U)x^{(k)}) \\ x^{(k+1)} &= \underbrace{D^{-1}(L+U)}_{T_j} x^{(k)} + \underbrace{D^{-1}b}_{c_j} \end{aligned}$$

Recordemos que pedimos que $A$ no tenga elementos nulos en su diagonal, luego $D$ resulta inversible. A su vez $L+U$ son todos los elementos de $A$ sin la diagonal y con el signo opuesto.

Si el método converge entonces $\lim_{k \to \infty} x^{(k)} = x^*$. Veamos si este esquema efectivamente obtiene una solución al sistema $Ax = b$.

$$\begin{aligned} x^* &= D^{-1}(b + (L+U)x^*) \\ Dx^* &= b + (L+U)x^* \\ Dx^* - (L+U)x^* &= b \\ (D - L - U)x^* &= b \\ Ax^* &= b \end{aligned}$$

**Gauss-Seidel**

La coordenada $i$-ésima de la solución en el paso $k+1$ de iteración se define de la siguiente manera:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \Big( b_i - \sum_{j=1}^{i-1} a_{ij} \overbrace{x_j^{(k+1)}}^{\text{coordenadas de este paso}} - \sum_{j=i+1}^{n} a_{ij} \overbrace{x_j^{(k)}}^{\text{coordenadas del paso anterior}} \Big)$$

Si llevamos esta expresión a una forma matricial obtenemos el esquema iterativo de Gauss-Seidel:

$$
\begin{aligned}
x^{(k+1)} &= D^{-1}(b + Lx^{(k+1)} + Ux^{(k)}) \\
Dx^{(k+1)} &= b + Lx^{(k+1)} + Ux^{(k)} \\
Dx^{(k+1)} - Lx^{(k+1)} &= b + Ux^{(k)} \\
(D-L)x^{(k+1)} &= b + Ux^{(k)} \\
x^{(k+1)} &= (D-L)^{-1}(b + Ux^{(k)}) \\
x^{(k+1)} &= \underbrace{(D-L)^{-1}U}_{T_{gs}} x^{(k)} + \underbrace{(D-L)^{-1}b}_{c_{gs}}
\end{aligned}
$$

Recordemos que pedimos que $A$ no tenga elementos nulos en su diagonal, luego $D - L$ resulta inversible.

Si el método converge entonces $\lim_{k\to\infty} x^{(k)} = x^*$. Veamos si este esquema efectivamente obtiene una solución al sistema $Ax = b$.

$$
\begin{aligned}
x^* &= (D-L)^{-1}Ux^* + (D-L)^{-1}b \\
(D-L)x^* &= Ux^* + b \\
(D-L)x^* - Ux^* &= b \\
(D-L-U)x^* &= b \\
Ax^* &= b
\end{aligned}
$$

### Matrices particulares

Si bien ambos métodos (Jacobi y Gauss-Seidel) son parecidos, la convergencia de uno no implica la del otro. Sin embargo, si tenemos hipótesis adicionales sobre la matriz del sistema $A \in \mathbb{R}^{n\times n}$ podemos afirmar la convergencia de forma directa.

- Si $A$ es estrictamente diagonal dominante (EDD) entonces Jacobi converge.
- Si $A$ es estrictamente diagonal dominante (EDD) entonces Gauss-Seidel converge.
- Si $A$ es simétrica definida positiva (SDP) entonces Gauss-Seidel converge.

Si además $a_{ii} > 0$ para todo $i = 1 \dots n$ y $a_{ij} < 0$ para todo $i \neq j$ se cumple una sola de las siguientes propiedades:

- $\rho(T_{gs}) < \rho(T_j) < 1$
- $\rho(T_{gs}) = \rho(T_j) = 0$
- $\rho(T_{gs}) > \rho(T_j) > 1$
- $\rho(T_{gs}) = \rho(T_j) = 1$

Lo cual nos dice o que ambos métodos convergen (y Gauss-Seidel lo hace más rápido) o que ninguno converge.

## Interpolación

### Motivación

Dado un conjunto de datos compuesto de pares ordenados $\{(x_0, y_0), \dots, (x_n, y_n)\}$ buscamos un polinomio $P(x)$ tal que:

$$
P(x_i) = y_i \qquad \text{para todo } i = 0 \dots n
$$

Decimos que el polinomio **interpola** los datos porque pasa por todos los puntos de nuestro conjunto. Si éstos describen algún fenómeno, nos gustaría utilizar el polinomio para predecir puntos desconocidos.

## Polinomio interpolante

Definimos $n+1$ polinomios de grado $n$ de la siguiente manera:

$$L_{nk} = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{x - x_i}{x_k - x_i}$$

Observemos que cada $L_{nk}$ tiene un comportamiento particular al ser evaluado en los puntos del dataset:

- $L_{nk}(x_i) = 0$ para todo $i = 0 \dots n, i \neq k$
- $L_{nk}(x_k) = 1$

Luego definimos el polinomio interpolante de grado $\leq n$ como:

$$P(x) = \sum_{k=0}^{n} y_k L_{nk}(x)$$

Al evaluar $P(x_i)$ para algún $i = 0 \dots n$ lo que sucede es que $y_i L_{ni}(x_i) = y_i * 1 = y_i$. Todos los otros $n$ términos de la sumatoria se anulan pues $L_{nk}(x_i) = 0$ si $i \neq k$.

$$P(x_i) = y_i \qquad \text{para todo } i = 0 \dots n$$

Luego $P(x)$ es el polinomio interpolante y además siempre existe.

## Error

Sean $f \in C^{n+1}[a,b]$ una función definida en el intervalo $[a,b]$ y el conjunto de pares ordenados $\{(x_i, f(x_i)) : x_i \in [a,b] \text{ para todo } i = 0 \dots n\}$. Sabemos que existe un polinomio interpolante $P(x)$ de grado $\leq n$ tal que $P(x_i) = f(x_i)$ para todo $i = 0 \dots n$.

Dado $\overline{x} \in [a,b]$, $\overline{x} \neq x_i$ para todo $i = 0 \dots n$, queremos conocer el error cometido al aproximar $f(\overline{x})$ con $P(\overline{x})$.

Para esto sabemos que existe una función $\xi(\overline{x})$ tal que:

$$f(\overline{x}) = P(\overline{x}) + \frac{f^{n+1}(\xi(\overline{x}))}{(n+1)!} \prod_{i=0}^{n} (\overline{x} - x_i)$$

## Unicidad

Dado un conjunto de datos $\{(x_0, y_0), \dots, (x_n, y_n)\}$ el polinomio interpolante de grado $\leq n$ es único.

Supongamos que existen 2 polinomios interpolantes distintos $P_1$ y $P_2$. Podemos considerar a $P_2$ como un polinomio interporlante de $P_1$ pues interpolan a los mismos datos: $P_2(x_i) = P_1(x_i)$ para todo $i = 0 \dots n$.

Como $P_1$ es de grado $\leq n$ su derivada de orden $n+1$ se anula. A partir de la fórmula del error podemos deducir que entonces $P_1 = P_2$.

## Métodos para obtener el polinomio interpolante

Ya sabemos que siempre existe el polinomio interporlante y que además es único salvo reescrituras. Supongamos que ya tenemos construido el polinomio interpolante para nuestros datos y ahora obtenemos nuevos puntos. ¿Cómo podemos actualizar el polinomio interporlante si se agregan nuevos puntos al dataset?

Si lo construimos a partir de los polinomios $L_{nk}$ tenemos que volver a repetir todo el proceso desde cero. Para evitar esto existen 2 formas (similares) de construcción del polinomio interporlante tal que permiten actualizarlo a partir de nuevos puntos sin necesidad de repetir pasos ya realizados.

**Diferencias divididas**

Dados $(x_i, f(x_i))$ para $i = 0 \dots n$ definimos las diferencias divididas como:

- Orden 0: $f[x_i] = f(x_i)$
- Orden 1: $f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}$
- Orden $k$: $f[x_i, x_{i+1}, \dots, x_{i+k}] = \frac{f[x_{i+1}, \dots, x_{i+k}] - f[x_i, \dots, x_{i+k-1}]}{x_{i+k} - x_i}$

Luego construimos el polinomio interporlante de la siguiente manera:

$$P_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots + f[x_0, \dots, x_n](x - x_0) \dots (x - x_{n-1})$$

Si ahora tenemos un nuevo punto $(x_{n+1}, y_{n+1})$ podemos actualizar el polinomio interpolante $P_n$ simplemente agregando un nuevo término que utiliza las diferencias divididas de orden $n + 1$. Calcular este término es fácil por la definición recursiva de las diferencias divididas (suponiendo que nos guardamos los resultados intermedios).

$$P_{n+1}(x) = P_n(x) + f[x_0, \dots, x_{n+1}](x - x_0) \dots (x - x_n)$$

**Proceso recursivo (Neville)**

Otra forma de construir el polinomio interpolante es a partir de otros 2 polinomios que interpolan en un punto menos.

Dados $(x_i, f(x_i))$ para $i = 0 \dots n$ notamos $P_{m_1, m_2, \dots, m_k}(x)$ al polinomio interpolante en los puntos $x_{m_1}, x_{m_2}, \dots, x_{m_k} \in \{x_0, \dots, x_n\}$.

Sean $i, j \in \{0, \dots, k\}$. Podemos expresar el polinomio interpolante $P_{0,\dots,k}(x)$ como:

$$P_{0,\dots,k} = \frac{(x - x_j) \overbrace{P_{0,\dots,j-1,j+1,\dots,k}(x)}^{\text{no interpola el punto } x_j} - (x - x_i) \overbrace{P_{0,\dots,i-1,i+1,\dots,k}(x)}^{\text{no interpola el punto } x_i}}{x_i - x_j}$$

Ahora definimos $Q_{ij} = P_{i-j,\dots,i}$ para $i \geq j$ como el polinomio interpolante de grado $\leq j$ en los puntos $x_{i-j}, x_{i-j+1}, \dots, x_i$.

$$Q_{ij}(x) = \frac{(x - x_{i-j})Q_{ij-1}(x) - (x - x_i)Q_{i-1j-1}(x)}{x_i - x_{i-j}}$$

Esto nos brinda una definición recursiva en donde el polinomio interpolante en todos los puntos del conjunto está dado por:

$$P(x) = P_{0,\dots,n}(x) = Q_{nn}(x)$$

## Interpolación segmentaria

Si tenemos muchos datos el polinomio interpolante tendrá un grado muy alto, y estos polinomios tienen el problema de presentar muchas oscilaciones. Si bien el polinomio evaluado en los puntos del dataset dan un resultado exacto, al evaluarlo en puntos desconocidos pueden arrojar valores muy "lejos" de los datos originales debido a estas oscilaciones.

Para remediar esta situación introducimos la interpolación segmentaria. Consiste en definir el polinomio interpolante como una función partida (pero contínua) construida a partir de polinomios interpolantes entre cada par de puntos consecutivos.

Ahora consideramos un conjunto de datos $\{(x_0, y_0), \ldots, (x_n, y_n)\}$ que está ordenado: $x_i < x_{i+1}$ para todo $i = 0 \ldots n-1$. Definimos la interpolación segmentaria como:

$$P(x) = \begin{cases} S_0(x) & x \in [x_0, x_1] \\ S_1(x) & x \in [x_1, x_2] \\ \vdots & \\ S_{n-1}(x) & x \in [x_{n-1}, x_n] \end{cases}$$

Donde $S_i$ interpola los puntos $(x_i, y_i)$ y $(x_{i+1}, y_{i+1})$. Además siempre pedimos que $S_i(x_{i+1}) = S_{i+1}(x_{i+1})$ para todo $i = 0 \ldots n-2$ así cada segmento se une con el siguiente.

**Interpolación lineal segmentaria**


Definimos $S_i$ como una recta que conecta los extremos de su respectivo intervalo.

$$S_i(x) = a_i + b_i(x - x_i)$$

Para cada $i = 0 \ldots n-1$ tenemos:

- 2 incógnitas: $a_i$ y $b_i$
- 2 ecuaciones:
  $S_i(x_i) = y_i$
  $S_i(x_{i+1}) = y_{i+1}$

Tenemos $2n$ incógnitas y $2n$ ecuaciones. Cada $S_i$ queda unívocamente determinado. No obstante, esta interpolación no es muy buena porque el polinomio interpolante $P(x)$ no resulta derivable en los puntos $x_0, \ldots, x_n$ ya que en esos puntos la función no es suave.

**Interpolación cuadrática segmentaria**


Definimos $S_i$ como un polinomio cuadrático que conecta los extremos de su respectivo intervalo.

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2$$

Para cada $i = 0 \ldots n-1$ tenemos:

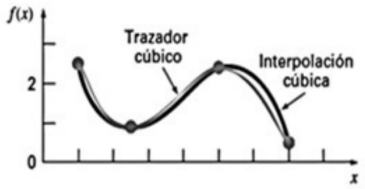
- 3 incógnitas: $a_i$, $b_i$ y $c_i$
- 2 ecuaciones:
  $S_i(x_i) = y_i$
  $S_i(x_{i+1}) = y_{i+1}$

Como ahora el polinomio es suave en los puntos interpolantes podemos agregar más ecuaciones usando las derivadas. Para cada $i = 0 \dots n-2$ tenemos:

- 1 ecuación: $S_i'(x_{i+1}) = S_{i+1}'(x_{i+1})$

En total tenemos $3n$ incógnitas y $2n + n - 1 = 3n - 1$ ecuaciones. Para determinar unívocamente al polinomio interpolante podríamos pedir algúna condición más respecto a la derivada en los extremos del conjunto de datos.

**Interpolación cúbica segmentaria (splines)**




La interpolación cúbica segmentaria utiliza polinomios cúbicos para interpolar cada par de puntos. Ahora cada segmento tiene derivada primera y segunda.

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^2$$

Para cada $i = 0 \dots n-1$ tenemos:

- 4 incógnitas: $a_i$, $b_i$, $c_i$ y $d_i$
- 2 ecuaciones:
  $S_i(x_i) = y_i$
  $S_i(x_{i+1}) = y_{i+1}$

Para cada $i = 0 \dots n-2$ tenemos:

- 2 ecuaciones:
  $S_i'(x_{i+1}) = S_{i+1}'(x_{i+1})$
  $S_i''(x_{i+1}) = S_{i+1}''(x_{i+1})$

En total tenemos $4n$ incógnitas y $2n + 2(n-1) = 4n - 2$ ecuaciones. Nos faltan 2 ecuaciones para que el polinomio interpolante quede unívocamente determinado.

La estrategia común suele ser agregar 2 condiciones respecto a las derivadas en los extremos.

- Opción 1: $S_0''(x_0) = S_{n-1}''(x_n) = 0$
- Opción 2: si conocemos la función $f$ que estamos queriendo interpolar podemos usarla para pedir que $S_0'(x_0) = f'(x_0)$ y $S_{n-1}'(x_n) = f'(x_n)$

Si desarrollamos todas las ecuaciones y las llevamos a una forma matricial obtenemos un sistema que resulta estrictamente diagonal dominante. Por lo cual existe solución al sistema y es única. Consecuentemente el polinomio interpolante construido a partir de la interpolación cúbica segmentaria siempre existe y es único.