

## Clase 17 - Resolución de sistemas lineales (2)

### El problema

Dada  $A \in \mathbb{R}^{n \times n}$  y  $b \in \mathbb{R}^n$  hallar  $x_*$  solución de  $Ax = b$ .

### Factorización LU

Vimos que, usando eliminación gaussiana, esto se puede resolver con un costo computacional de  $\mathcal{O}(\frac{2}{3}n^3)$  flops. Este costo computacional se debe principalmente a las operaciones que se deben realizar para transformar la matriz  $A$  en una matriz triangular superior  $U$ , que es mucho mayor que la cantidad de operaciones que se realizan para transformar el vector  $b$  en otro vector  $y$ .

Ahora supongamos que se deben resolver varios sistemas lineales  $Ax = b$ , con la misma matriz  $A$  y diferentes vectores  $b$ . Sería poco eficiente repetir todas las operaciones elementales por fila del proceso de eliminación gaussiana siendo que sólo el vector  $b$  es modificado y la matriz  $A$  se mantiene igual.

La idea de la factorización LU consiste en factorizar la matriz como un producto de dos matrices más simples:  $A = LU$ , donde  $L$  es una matriz triangular inferior con elementos diagonales iguales a uno ( $l_{ii} = 1, i = 1, \dots, n$ ) y  $U$  triangular superior. Así:

$$Ax = b \iff LUx = b \iff \begin{cases} Ly = b \\ Ux = y \end{cases}$$

Es decir que, conocidas las matrices  $L$  y  $U$  de la factorización de  $A$ , en lugar resolver  $Ax = b$ , se deben resolver dos sistemas triangulares para cada nuevo vector  $b$ .

Estas matrices  $L$  y  $U$  se obtienen a partir de  $A$  de una forma constructiva, como veremos a continuación. La idea es escribir  $A = LU$ , o equivalentemente  $LU = A$ , y tratar de despejar los coeficientes de  $L$  y de  $U$ , usando inducción:

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ l_{21} & 1 & 0 & \dots & 0 \\ l_{31} & l_{32} & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & u_{22} & u_{23} & \dots & u_{2n} \\ 0 & 0 & u_{33} & \dots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & u_{nn} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix}.$$

Vamos a determinar la primera fila de  $U$  y la primera columna de  $L$ .

Si multiplicamos la primera fila de  $L$  por la columna  $j$  de  $U$  para  $j = 1, \dots, n$ , obtenemos:

$$a_{1j} = 1u_{1j} + 0u_{2j} + 0u_{3j} + \dots + 0u_{nj},$$

por lo tanto,

$$u_{1j} = a_{1j}, \quad \text{para } j = 1, \dots, n.$$

Si multiplicamos cada fila de  $L$  por la primera columna de  $U$ , obtenemos:

$$a_{i1} = l_{i1}u_{11}, \quad \text{para } i = 2, \dots, n,$$

por lo tanto,

$$l_{i1} = \frac{a_{i1}}{u_{11}}, \quad \text{para } i = 2, \dots, n.$$

Ahora, supongamos conocidas las  $(k-1)$  filas de  $U$  y  $(k-1)$  columnas de  $L$  y vamos a determinar la fila  $k$  de  $U$  y la columna  $k$  de  $L$ .

Si multiplicamos la fila  $k$  de  $L$  por la columna  $j$  de  $U$  para  $j = k, \dots, n$ , obtenemos:

$$a_{kj} = \sum_{m=1}^{k-1} l_{km}u_{mj} + u_{kj},$$

por lo tanto,

$$u_{kj} = a_{kj} - \sum_{m=1}^{k-1} l_{km}u_{mj}, \quad \text{para } j = k, \dots, n.$$

Si multiplicamos cada fila de  $L$ , desde la  $k+1$  hasta la última, por la columna  $k$  de  $U$ , obtenemos:

$$a_{ik} = \sum_{m=1}^{k-1} l_{im}u_{mk} + l_{ik}u_{kk},$$

por lo tanto,

$$l_{ik} = \frac{1}{u_{kk}}(a_{ik} - \sum_{m=1}^{k-1} l_{im}u_{mk}) \quad \text{para } i = k+1, \dots, n.$$

A continuación veremos el algoritmo para realizar la factorización LU.

**Algoritmo:** Factorización LU.

**input**  $n, A = (a_{ij})$

**for**  $k = 1, \dots, n$  **do**

**for**  $j = k, \dots, n$  **do**

$$u_{kj} \leftarrow a_{kj} - \sum_{m=1}^{k-1} l_{km}u_{mj}$$

**end for** ( $j$ )

**for**  $i = k+1, \dots, n$  **do** (no ejecutar si  $k = n$ )

$$l_{ik} \leftarrow \frac{1}{u_{kk}}(a_{ik} - \sum_{m=1}^{k-1} l_{im}u_{mk})$$

**end for** ( $i$ )

**end for** ( $k$ )

**output**  $L = (l_{ij}), U = (u_{ij})$

**end**

El siguiente resultado da una condición suficiente para poder realizar la factorización LU.

**Teorema 1.** Sea  $A \in \mathbb{R}^{n \times n}$  tal que las submatrices  $A_k = A(1:k, 1:k)$  para  $k = 1, \dots, n-1$  son no singulares. Entonces existen únicas matrices  $L, U \in \mathbb{R}^{n \times n}$  tal que  $L$  es triangular inferior con 1 en la diagonal ( $l_{ii} = 1, i = 1, \dots, n$ ) y  $U$  es triangular superior. Además,  $\det(A) = u_{11}u_{22} \dots u_{nn}$ .

*Demostración.* La prueba se hará por inducción en la dimensión de la matriz.

Si  $n = 1$ , entonces si  $A = LU$ , basta tomar  $L = [1]$  y  $U = [a_{11}]$ . También se cumple que  $\det(A) = u_{11}$ .

Supongamos que la factorización es válida para dimensión  $k - 1$ , es decir, dada  $A_{k-1} \in \mathbb{R}^{(k-1) \times (k-1)}$ , existen  $L_{k-1}, U_{k-1} \in \mathbb{R}^{(k-1) \times (k-1)}$  tal que  $A_{k-1} = L_{k-1}U_{k-1}$ . Veamos para  $k$ , o sea, veremos que para una matriz  $A_k \in \mathbb{R}^{k \times k}$  existen  $L_k, U_k \in \mathbb{R}^{k \times k}$  tal que  $A_k = L_k U_k$ . Para determinar esto vamos a particionar convenientemente:

$$\left[ \begin{array}{c|c} A_{k-1} & a \\ \hline b^T & c \end{array} \right] = \left[ \begin{array}{c|c} L_{k-1} & 0 \\ \hline d^T & 1 \end{array} \right] \left[ \begin{array}{c|c} U_{k-1} & e \\ \hline 0^T & u_{kk} \end{array} \right],$$

donde  $a, b, d, e \in \mathbb{R}^{k-1}$  y  $c, u_{kk} \in \mathbb{R}$ .

Realizando los productos en bloques del lado derecho e igualando al lado izquierdo, obtenemos:

$$A_{k-1} = L_{k-1}U_{k-1} \quad (1)$$

$$L_{k-1}e = a \quad (2)$$

$$d^T U_{k-1} = b^T \quad (3)$$

$$d^T e + u_{kk} = c \quad (4)$$

De (2), y como  $L_{k-1}$  es triangular inferior con unos en la diagonal, es claro que existe un único  $e \in \mathbb{R}^{k-1}$ .

De (3), aplicando transpuesta a ambos lados,  $(d^T U_{k-1})^T = b$ , es decir,  $U_{k-1}^T d = b$ , y como  $U_{k-1}$  tiene inversa por hipótesis inductiva y por que las matrices  $A_k$  son no singulares, entonces existe un único  $d \in \mathbb{R}^{k-1}$ .

Luego, de (4),  $u_{kk} = c - d^T e$  está bien definido y está unívocamente determinado.

De esta manera se ha probado que existen  $L_k$  y  $U_k$  tales que  $A_k = L_k U_k$ .

Por último, como  $A = LU$ , entonces

$$\det(A) = \det(LU) = \det(L) \det(U) = 1 \cdot (u_{11}u_{22} \dots u_{nn}) = u_{11}u_{22} \dots u_{nn}.$$

□

**Observación 1:** El costo de realizar la factorización LU es el mismo que para realizar la eliminación gaussiana, es decir,  $\mathcal{O}(\frac{2}{3}n^3)$ .

**Observación 2:** Si la matriz  $A$  no será usada posteriormente, una implementación eficiente de la factorización LU, podría sobrescribir las matrices  $L$  y  $U$  sobre la misma  $A$ , teniendo en cuenta que los coeficientes de la diagonal de  $L$  son todos iguales a 1. Más aún, se puede ver que los elementos de la matriz  $L$  debajo de la diagonal son los multiplicadores que aparecían en la eliminación gaussiana.

**Observación 3:** este es el método más eficiente para calcular el determinante de una matriz. El costo computacional será aproximadamente el mismo que para realizar la factorización LU.

Recordemos la clásica regla para calcular el determinante para una matriz  $2 \times 2$ :

$$\det(A) = \det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$$

Si usáramos esta regla se requieren 2 productos y una resta para calcular el determinante. Lamentablemente, esta cantidad aumenta increíblemente cuando crece el orden  $n$  de la matriz. Por ejemplo, si  $A$  fuera una matriz  $20 \times 20$ . Entonces se requieren del orden de  $20! \approx 2.4 \cdot 10^{18}$  operaciones. Si se usara una computadora que realiza  $10^9$  operaciones en punto flotante por segundo, entonces se necesitarían 76 años para calcular ese determinante.

**Observación 4:** nunca se debe calcular la inversa de una matriz. Siempre es mucho más eficiente reformular el problema para resolver sistemas lineales. Sin embargo, si el único objetivo del problema es obtener la matriz de una inversa de una matriz, la forma más eficiente consiste en realizar una factorización LU de la matriz  $A$  y luego resolver  $n$  sistemas lineales de la forma  $LUx = e_i$ , donde  $e_i$  es el  $i$ -ésimo vector de la base canónica que tiene un 1 en la posición  $i$ , y 0 en las otras componentes, para  $i = 1, \dots, n$ .

A continuación veremos algunas definiciones y resultados que serán de utilidad para la clase siguiente.

**Definición 1.** Una **norma vectorial** en  $\mathbb{R}^n$  es una función que asigna a cada  $x \in \mathbb{R}^n$  un número real no negativo denotado por  $\|x\|$  y llamado **norma de  $x$** , tal que se satisfacen las siguientes tres propiedades para todo  $x, y \in \mathbb{R}^n$  y para todo  $\alpha \in \mathbb{R}$ :

1.  $\|x\| > 0$  si  $x \neq 0$ , y  $\|0\| = 0$ ;
2.  $\|\alpha x\| = |\alpha| \|x\|$ ;
3.  $\|x + y\| \leq \|x\| + \|y\|$ .

**Ejemplos:**

1. norma euclídeana (o norma 2):  $\|x\|_2 = \left( \sum_{i=1}^n |x_i|^2 \right)^{1/2}$  ;

2. norma 1:  $\|x\|_1 = \sum_{i=1}^n |x_i|$ ;

3. para  $p \geq 1$ , norma  $p$ :  $\|x\|_p = \left( \sum_{i=1}^n |x_i|^p \right)^{1/p}$  ;

4. norma infinito:  $\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$ ;

**Definición 2.** Dados dos vectores  $x, y \in \mathbb{R}^n$  y una norma vectorial  $\|\cdot\|$  se define la **distancia entre  $x$  e  $y$**  por:

$$d(x, y) = \|x - y\|.$$

El concepto de normas puede extenderse al conjunto de matrices.

**Definición 3.** Una **norma matricial** en  $\mathbb{R}^{n \times n}$  es una función que asigna a cada  $A \in \mathbb{R}^{n \times n}$  un número real no negativo denotado por  $\|A\|$  y llamado **norma de A**, tal que se satisfacen las siguientes cuatro propiedades para todo  $A, B \in \mathbb{R}^{n \times n}$  y para todo  $\alpha \in \mathbb{R}$ :

1.  $\|A\| > 0$  si  $A \neq 0$ , y  $\|0\| = 0$ ;
2.  $\|\alpha A\| = |\alpha| \|A\|$ ;
3.  $\|A + B\| \leq \|A\| + \|B\|$ ;
4.  $\|AB\| \leq \|A\| \|B\|$ .

**Ejemplo:** norma de Frobenius

$$\|A\|_F = \left( \sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2 \right)^{1/2}.$$

**Definición 4.** Dada una norma vectorial en  $\mathbb{R}^n$  y una matriz  $A \in \mathbb{R}^{n \times n}$  se define la norma matricial inducida por

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}.$$

**Proposición 1.** Sean  $A \in \mathbb{R}^{n \times n}$  y  $\|\cdot\|$  una norma vectorial en  $\mathbb{R}^n$  que induce una norma matricial, entonces:

1.  $\|Ax\| \leq \|A\| \|x\|$  para todo  $x \in \mathbb{R}^n$ ;
2. existe  $\bar{x}$  con  $\|\bar{x}\| = 1$  tal que  $\|A\bar{x}\| = \|A\|$ .

**Demostración.** - Probemos la primera parte.

Si  $x = 0$ , entonces  $\|Ax\| = 0 = \|A\| \|x\|$ .

Si  $x \neq 0$ , entonces  $\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} \geq \frac{\|Ax\|}{\|x\|}$  para todo  $x$ , y por lo tanto  $\|Ax\| \leq \|A\| \|x\|$  para todo  $x \in \mathbb{R}^n$ .

- Ahora probemos la segunda parte.

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \sup_{x \neq 0} \left\| \frac{Ax}{\|x\|} \right\| = \sup_{x \neq 0} \left\| A \left( \frac{x}{\|x\|} \right) \right\| = \sup_{y: \|y\|=1} \|Ay\|.$$

Como  $S = \{x \in \mathbb{R}^n \mid \|x\| = 1\}$  es cerrado y acotado (compacto), la función continua  $x \rightarrow \|Ax\|$  alcanza sus valores extremos, en particular existe un  $\bar{x}$  tal que  $\|\bar{x}\| = 1$  y

$$\|A\| = \sup_{y: \|y\|=1} \|Ay\| = \max_{y: \|y\|=1} \|Ay\| = \|A\bar{x}\|.$$

□

**Ejemplos:**

$$1. \|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|;$$

---

2.  $\|A\|_{\infty} = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|;$

3.  $\|A\|_2 = \sqrt{\rho(A^T A)},$   
donde  $\rho(B)$  es el radio espectral de  $B$  y se define por  $\max\{|\lambda| : \lambda \text{ es autovalor de } B\}.$

A. Numérico - FAMAF 2022