



Métodos Matemáticos de Especialidad

Ingeniería Eléctrica

“Algo de álgebra lineal numérica”

José Luis de la Fuente O'Connor

Escuela Técnica Superior de Ingenieros Industriales
Universidad Politécnica de Madrid

jl.delafuente@iberdrola.es



Índice



1. Introducción

2. Operaciones con matrices

- **Producto de una matriz por un vector**
- **Producto de dos matrices**
 - **Fórmula de Strassen**

3. Normas

- **Vectoriales**
- **Matriciales**

4. Condicionamiento de sistemas de ecuaciones



Introducción



3/48

Las manipulaciones numéricas de vectores y matrices son operaciones fundamentales en los métodos que estamos analizando en este curso.

En su versión básica o especializada para diversas estructuras constituyen la base de casi todos los procedimientos o algoritmos que estudiamos.

En este apartado nos centraremos en las implicaciones de cálculo de las formas más eficaces y simples de efectuar productos de vectores por matrices, de dos matrices, cálculo de normas de vectores y matrices y, por último, la importancia que adquieren para determinar el condicionamiento de sistemas de ecuaciones lineales.



Full Scrn

Imprimir

Operaciones con matrices



4/48

Producto de una matriz por un vector

Es una de las operaciones esenciales del álgebra lineal numérica.

Se puede interpretar, si se desea multiplicar A por el vector x , como una combinación lineal de los vectores columna de A o como una serie de productos interiores de vectores: los vectores fila de A y el propio x .

Comenzando por el caso matriz por vector, Ax , la forma usual de considerarlo es la que sigue:

$$y_i = \sum_{j=1}^n A_{ij}x_j,$$

donde $A \in \mathbb{R}^{m \times n}$ y $x \in \mathbb{R}^n$. Para ser factible esta operación



Full Screen

Imprimir

lógicamente, se deben dar las dimensiones correspondientes:

$$[m \times n] \cdot [n \times 1] = [m \times 1].$$

Un algoritmo sencillo para llevarlo a cabo sería el que sigue.

```
*  
* PRODUCTO  $y = Ax$   
*  
Hacer  $y = 0$   
for  $i = 1$  to  $m$   
  for  $j = 1$  to  $n$   
     $y(i) = y(i) + A(i, j)x(j)$   
  end  
end
```





Aun cuando en Matlab este producto sería fácilmente obtenible sin más que hacer $y=A*x$, programar su implementación sin ulteriores complicaciones sería así de sencillo.

```
[m,n]=size(A);  
y=zeros(m,1);  
for i=1:m  
    for j=1:n  
        y(i)=y(i)+A(i,j)*x(j);  
    end  
end
```

Aunque no es necesario pues todo funciona bien, podemos darnos cuenta que el bucle en j materializa el producto interior de la fila i de A por el vector x .



Se podría entonces hacer algo un poquito más sofisticado como lo que sigue.

```
function y = MatVecF(A,x)
%
[m,n] = size(A);
y = zeros(m,1);
for i=1:m
    y(i) = A(i,:)*x;
end;
```

Éste sería el algoritmo en su versión **por filas**: una suma de los productos interiores de los vectores fila de A por x .

Uno **por columnas** debería tener en cuenta que el producto Ax al fin y al cabo es hacer una combinación lineal de los vectores columna de A . En efecto, por ejemplo,

$$y = Ax = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \begin{bmatrix} 7 \\ 8 \end{bmatrix} = \begin{bmatrix} 1 \cdot 7 + 2 \cdot 8 \\ 3 \cdot 7 + 4 \cdot 8 \\ 5 \cdot 7 + 6 \cdot 8 \end{bmatrix} = 7 \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix} + 8 \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} = \begin{bmatrix} 23 \\ 53 \\ 83 \end{bmatrix}.$$



Un algoritmo que plasma la recurrencia general de esto es el que sigue.

```
function y = MatVecC(A,x)
%
[m,n] = size(A);
y = zeros(m,1);
for j=1:n
    y = y + A(:,j)*x(j);
end;
```

Esta operación en los términos que se manejan en la literatura especializada en estas cuestiones se denomina **saxpy**.

El número de operaciones que se realizan en $\mathcal{O}(mn)$ multiplicaciones e igual número de sumas/restas.

El que esta forma de realizar la operación sea más o menos rápida que la anterior dependerá de cómo almacena el ordenador la matriz A .





Si, como es lo habitual, guarda la matriz por columnas, esta operación requerirá menos tiempo que la versión por filas pues los elementos que aquí se van accediendo están contiguos en la memoria.

Hasta ahora hemos estudiado el producto $y = Ax$. Para llevar a cabo $y^T = x^T A$, las versiones de los programas anteriores son las que se indican a continuación.

```
function y = VecMatF(x,A)
%
[m,n] = size(A);
y = zeros(1,n);
for j=1:m
    y = y + x(j)*A(j,:);
end;
```

```
function y = VecMatC(x,A)
%
[m,n] = size(A);
y = zeros(1,n);
for j=1:n
    y(j) = y(j) + x*A(:,j);
end;
```



Producto de dos matrices



10/48

Si $A \in \mathbb{R}^{m \times r}$ y $B \in \mathbb{R}^{r \times n}$, el producto $C \in \mathbb{R}^{m \times n}$ de estas dos matrices se define así:

$$c_{ij} = \sum_{k=1}^r a_{ik} b_{kj},$$

para todo $1 \leq i \leq m$ y $1 \leq j \leq n$. El algoritmo es el que sigue.

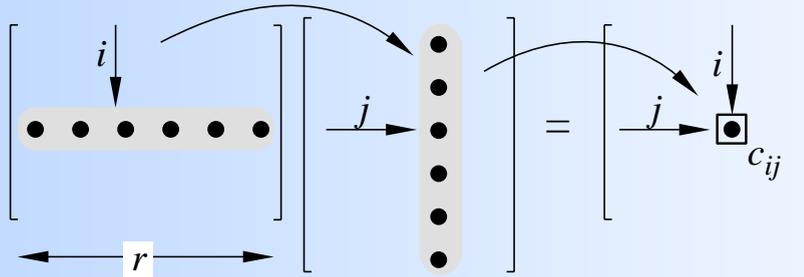
```
*  
* PRODUCTO  $C = AB$   
*  
Hacer  $C = 0$   
for  $j = 1$  to  $n$   
  for  $i = 1$  to  $m$   
    for  $k = 1$  to  $r$   
       $C(i, j) = C(i, j) + A(i, k) * B(k, j)$   
    end  
  end  
end
```



Full Scrn

Imprimir

Cada coeficiente c_{ij} de la matriz resultante se puede contemplar como el resultado de **un producto interior** tal como se indica.

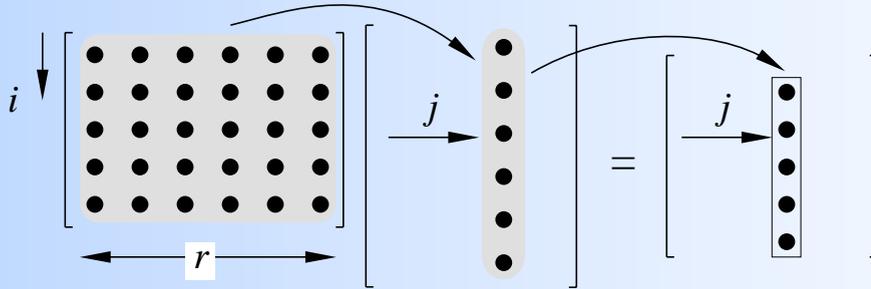


Según esta idea, el algoritmo para obtener el producto $C = AB$ tendría la forma que sigue.

```
function C = MatMatInt(A,B)
%
[m,r] = size(A);
[r,n] = size(B);
C = zeros(m,n);
for j=1:n
    for i=1:m
        C(i,j) = A(i,:)*B(:,j);
    end
end
```



Ahora bien, cada columna j de la matriz C es el resultado de multiplicar la matriz A por el vector columna j -ésimo de B como indica la figura.



Para implementar un algoritmo eficazmente hay que tener en cuenta lo hecho al estudiar el producto de matrices por vectores. El resultado es el que se lista a continuación.





```
function C = MatMatVec(A,B)
%
[m,r] = size(A);
[r,n] = size(B);
C = zeros(m,n);
for j=1:n
    C(:,j) = C(:,j) + A*B(:,j);
end;
```

Por último, la matriz C también se puede ver como el resultado de sumar r matrices de rango uno resultado del **producto exterior** de dos vectores.

Recordemos, el producto exterior de u y v es

$$uv^T = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} [v_1 v_2 \cdots v_n] = \begin{bmatrix} u_1 v_1 & u_1 v_2 & \cdots & u_1 v_n \\ u_2 v_1 & u_2 v_2 & \cdots & u_2 v_n \\ \vdots & \vdots & \ddots & \vdots \\ u_m v_1 & u_m v_2 & \cdots & u_m v_n \end{bmatrix}.$$



Según esto,

$$C = AB = \sum_{k=1}^r A(:, k)B(k, :).$$

Un algoritmo epara Matlab que plasme esta forma de hacerlo es el que sigue.

```
function C = MatMatExte(A,B)
%
[m,r] = size(A);
[r,n] = size(B);
C = zeros(m,n);
for k=1:r
    C = C + A(:,k)*B(k,:);
end;
```

El número de operaciones que realizan todos estos métodos es $\mathcal{O}(mnp)$ multiplicaciones e igual número de sumas/restas.

Como en el caso de los productos de matrices por vectores, el



procedimiento más adecuado dependerá de cómo almacena el ordenador las matrices. Lo más habitual es que el más eficaz se el que denominabamos MatMatVec .



Fórmula de Strassen

La fórmula o esquema de Strassen para multiplicar dos matrices A y B cuadradas y de dimensión 2^p se basa en reordenar los cálculos de una forma muy ingeniosa.

Si A y B son 2×2 , el producto es

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{12}B_{12} + A_{11}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{bmatrix}.$$

Esta operación necesita 8 multiplicaciones y 4 sumas.

El esquema de Strassen reagrupa las operaciones de tal manera que



se necesiten 7 multiplicaciones y 18 sumas de esta forma.

$$P_1 = (A_{11} + A_{22})(B_{11} + B_{22})$$

$$P_2 = (A_{21} + A_{22})B_{11}$$

$$P_3 = A_{11}(B_{12} - B_{22})$$

$$P_4 = A_{22}(B_{21} - B_{11})$$

$$P_5 = (A_{11} + A_{12})B_{22}$$

$$P_6 = (A_{21} - A_{11})(B_{11} + B_{12})$$

$$P_7 = (A_{12} - A_{22})(B_{21} + B_{22})$$

$$C_{11} = P_1 + P_4 - P_5 + P_7$$

$$C_{12} = P_3 + P_5$$

$$C_{21} = P_2 + P_4$$

$$C_{22} = P_1 + P_3 - P_2 + P_6$$

Generalizando, las 7 multiplicaciones son de matrices $m \times m$ por lo que se requieren

$$\frac{7}{8}m^3 \text{ multiplicaciones} + \left(\frac{7}{8}m^2(m-2) + \frac{9}{2}m^2 \right) \text{ sumas.}$$





La operación natural requeriría m^3 multiplicaciones y $m^2(m - 1)$ sumas.

Para un m suficientemente grande el procedimiento de Strassen reduce las operación por un factor de alrededor de $7/8$.

Si se aplica recursivamente la idea y se van dividiendo las matrices originales en cuatro bloques, cada unos de estos en otros cuatro y así sucesivamente hasta llegar a 1, el procedimiento mejorará en mucho el número total de operaciones a realizar.

El orden de las matrices menores, no obstante, al que se tiene que llegar es $n_{min} = 16$ pues con bloques de menor tamaño el número de operaciones que se realiza es mayor que el ordinario.

El programa que sigue implementa el método de Strassen para matrices $n \times n$ con n^p .





```
function C = strass(A,B)
%
%   A,B  matrices n x n, n potencia de 2.
%
[n,n] = size(A);
if n <= 16
    C = A*B;
else
    m = n/2; u = 1:m; v = m+1:n;
    P1 = strass(A(u,u)+A(v,v),B(u,u)+B(v,v));
    P2 = strass(A(v,u)+A(v,v),B(u,u));
    P3 = strass(A(u,u),B(u,v)-B(v,v));
    P4 = strass(A(v,v),B(v,u)-B(u,u));
    P5 = strass(A(u,u)+A(u,v),B(v,v));
    P6 = strass(A(v,u)-A(u,u),B(u,u) + B(u,v));
    P7 = strass(A(u,v)-A(v,v),B(v,u)+B(v,v));
    C = [ P1+P4-P5+P7 P3+P5; P2+P4 P1+P3-P2+P6];
end
```



Normas



19/48

Vectoriales

Si en un espacio vectorial E sobre \mathbb{K} (\mathbb{R} o \mathbb{C}) se define una **norma vectorial** como una aplicación $\| \cdot \| : E \rightarrow \mathbb{R}$ que verifica

$$\|v\| = 0 \implies v = \mathbf{0} \text{ y } x \neq \mathbf{0} \implies \|x\| > 0,$$

$$\|\alpha v\| = |\alpha| \|v\| \text{ para } \alpha \in K \text{ y } v \in E,$$

$$\|u + v\| \leq \|u\| + \|v\| \quad \forall u, v \in E,$$

se dice que E es un **espacio vectorial normado**.

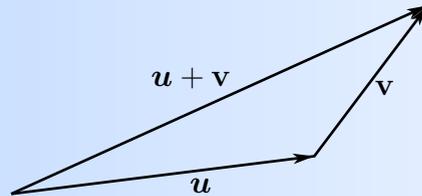
La condición $\|u + v\| \leq \|u\| + \|v\|$ se conoce como **regla del triángulo**.



Full Screen

Imprimir

Es una generalización del hecho de que un lado de un triángulo no puede ser mayor que la suma de los otros dos: ver figura.



Una variante también útil de esta regla es la siguiente:

$$\|u - v\| \geq \|u\| - \|v\|.$$

En el espacio vectorial \mathbb{K}^n , para $1 \leq p < \infty$, se tiene la familia de normas

$$\|x\|_p = \left[|x_1|^p + \cdots + |x_n|^p \right]^{1/p},$$

denominadas **normas p de Hölder**.



Casos particulares lo constituyen las correspondientes a $p = 1$ y $p = 2$:

$$\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$$
$$\|\mathbf{x}\|_2 = \sqrt{|x_1|^2 + \cdots + |x_n|^2}.$$

Esta última se denomina en \mathfrak{R}^n **norma euclídea**.

También en \mathbb{K}^n es una norma la dada por

$$\|\mathbf{x}\|_\infty = \text{máx.}_{1 \leq i \leq n} |x_i|.$$

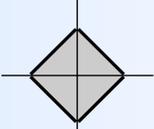
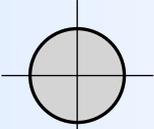
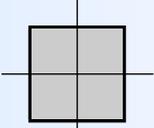
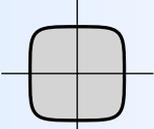
Estas normas cumplen, cualquiera que sea $x \in \mathbb{K}^n$, que

$$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1 \leq n \|\mathbf{x}\|_\infty.$$



Si la bola cerrada unidad en \mathbb{R}^2 es el conjunto $\{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x}\| \leq 1\}$, sus formas para las normas vectoriales 1, 2, ∞ , y p son las que representa la tabla.



$\ \mathbf{x}\ _1 = \sum_{i=1}^2 x_i $	
$\ \mathbf{x}\ _2 = \sqrt{ x_1 ^2 + x_2 ^2} = \sqrt{\mathbf{x}^T \mathbf{x}}$	
$\ \mathbf{x}\ _\infty = \max_{1 \leq i \leq 2} x_i $	
$\ \mathbf{x}\ _p = \left[x_1 ^p + x_2 ^p \right]^{1/p}, \quad (1 \leq p < \infty)$	





En el espacio $C[0, 1]$, de funciones continuas del intervalo $[0, 1]$ en \mathbb{C} , son normas las dadas por

$$\|f\|_p = \left[\int_0^1 |f(t)|^p dt \right]^{1/p}$$

y por

$$\|f\|_\infty = \text{máx.}_{t \in [0,1]} |f(t)|.$$

En un espacio vectorial normado se define la **distancia** entre dos elementos u y v mediante

$$d(u, v) = \|u - v\|.$$

Esta definición convierte a cualquier espacio vectorial normado en un **espacio métrico**.



Matriciales



Una **norma matricial** es una norma vectorial compatible con el producto.

Se define formalmente sobre $\mathfrak{R}^{m \times n}$ como una aplicación

$\| \cdot \| : \mathfrak{R}^{m \times n} \rightarrow \mathfrak{R}$ que cumple:

- 1) $\|A\| = 0 \implies A = 0$.
- 2) $\|\lambda A\| = |\lambda| \cdot \|A\|$.
- 3) $\|A + B\| \leq \|A\| + \|B\|$.
- 4) $\|AB\| \leq \|A\| \cdot \|B\|$.

Existen normas sobre el espacio $\mathfrak{R}^{m \times n}$ que no son normas matriciales pues no cumplen la propiedad 4). Así, si se define

$$\|A\| = \text{máx.}_{1 \leq i, j \leq n} |a_{ij}|,$$

se satisfacen 1), 2) y 3); sin embargo, tomando $A = B = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$, es



fácil ver que $\|AB\| = 2 > \|A\| \cdot \|B\| = 1$, por lo que no se cumple 4).

Una norma matricial $\|\cdot\|$ sobre $\mathbb{R}^{m \times n}$ se dice **consistente** con una norma vectorial $\|\cdot\|'$ sobre \mathbb{R}^n cuando para cada matriz A y cada vector x se cumple que

$$\|Ax\|' \leq \|A\| \cdot \|x\|'.$$

Se demuestra que para toda norma matricial es posible construir una norma vectorial consistente. Recíprocamente, a toda norma vectorial sobre \mathbb{R}^n se le puede asociar una norma matricial consistente.

Una norma matricial consistente con una cierta norma vectorial $\|\cdot\|$ se construye mediante la definición

$$\|A\| = \sup_{\mathbf{0} \neq \mathbf{x} \in \mathbb{R}^n} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|}.$$

Esta norma matricial se dice **inducida** por la norma vectorial.



Ejemplo: la norma matricial inducida por la norma euclídea de \mathbb{R}^n es la **norma espectral**, **norma euclídea** o **norma 2**:

$$\|A\|_2 = \sup_{\mathbf{0} \neq \mathbf{x} \in \mathbb{R}^n} \left[\frac{\mathbf{x}^T A^T A \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \right]^{1/2} = \sqrt{\lambda_{\max}(A^T A)} = \sigma_{\max}(A),$$

donde λ designa un autovalor de A y σ un valor singular.

Las normas matriciales inducidas más usadas son

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^m |a_{ij}| \quad \text{y}$$
$$\|A\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}|.$$

Si $\|\cdot\|$ es la norma inducida por una cierta norma vectorial y $\|\cdot\|'$ es una norma matricial cualquiera consistente con esa norma vectorial, se cumple, para toda matriz A , que $\|A\| \leq \|A\|'$.



Un ejemplo importante de norma matricial es la **norma de Frobenius**, definida como:

$$\|A\|_F^2 = \sum_{1 \leq i, j \leq n} a_{ij}^2 = \text{traza}(A^T A),$$

donde la **traza** de una matriz A de orden n es $\sum_{i=1}^n a_{ii}$.

La norma de Frobenius cumple que

$$\|AB\|_F \leq \|A\|_F \cdot \|B\|_F,$$

como se puede comprobar fácilmente.

La norma espectral y la norma de Frobenius cumplen que

$$\|A\|_2 \leq \|A\|_F.$$

Ejemplo

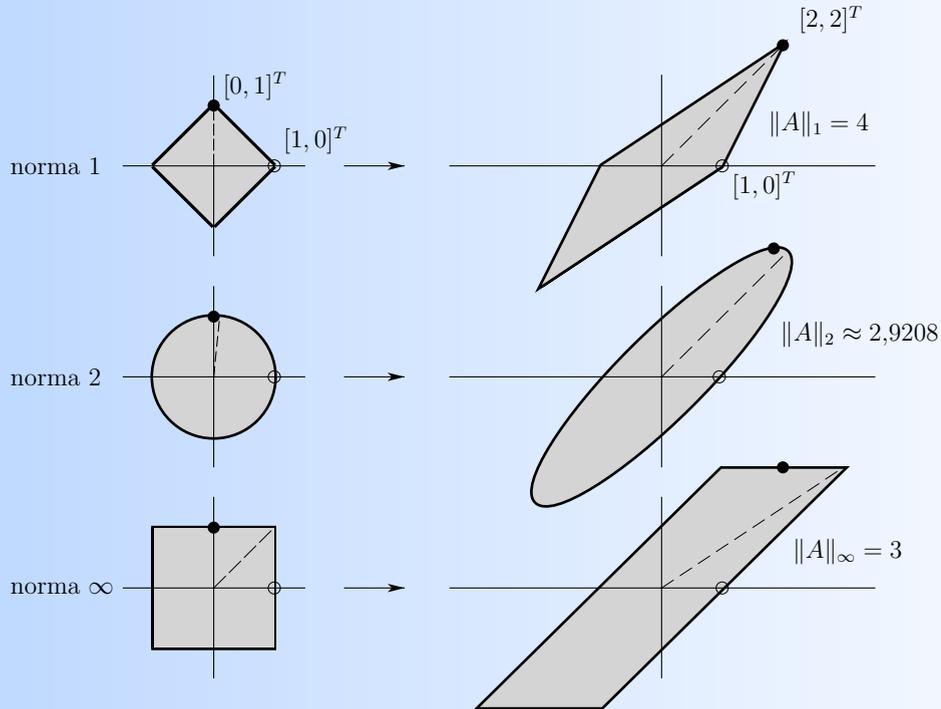
El efecto que produce aplicar la transformación lineal basada en la



matriz

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix}$$

sobre la bola unidad definida a partir de las normas $\|\cdot\|_1$, $\|\cdot\|_2$ y $\|\cdot\|_\infty$ en \mathcal{R}^2 , se representa en la figura.





La aplicación transforma el vector $e_1 = [1, 0]^T$ en sí mismo y $e_2 = [0, 1]^T$ en $[2, 2]^T$.

Tomando la norma 1, el vector unitario que más se amplifica al aplicarle la transformación es $[0, 1]^T$ (o $[0, -1]^T$), que pasa a ser $[2, 2]^T$. Su factor de amplificación, en términos de la norma 1, es 4.

Tomando la norma 2, el vector unitario que más se amplifica es el que se representa en la figura con una recta discontinua. El factor de amplificación es 2,9208.

Para la norma ∞ , igualmente, el vector unitario que más se amplifica es el que se representa también con la recta discontinua: $[1, 1]^T$, que pasa a transformarse en $[3, 2]^T$. El factor de amplificación correspondiente es en este caso 3 ya que

$$\begin{aligned}\|[1, 1]^T\|_{\infty} &= 1 \\ \|[3, 2]^T\|_{\infty} &= 3.\end{aligned}$$



Condicionamiento de sistemas de ecuaciones lineales



30/48

El concepto de **condicionamiento de un problema** es algo a lo que se recurre a menudo aun cuando su sentido sea tal vez difuso o vago.

En un sentido general, se dice que un problema está bien condicionado si pequeños cambios en los parámetros que lo definen producen pequeños cambios en los resultados.

Como ejemplo de condicionamiento podríamos considerar el de una carga sujeta a una superficie firme mediante un cable o una barra de hierro.

Aumentando la carga en pequeñas cantidades, el cable o barra sufre unos pequeños estiramientos proporcionales a los incrementos de esa carga.



Full Screen

Imprimir



Alcanzado el umbral que define la zona denominada de fluencia, incrementos muy pequeños de la carga suponen, proporcionalmente, grandes estiramientos del cable. Antes de este umbral, el problema estiramiento/carga se puede decir que está bien condicionado; en la zona de fluencia, por el contrario, el problema está mal condicionado.

El condicionamiento de un sistema de ecuaciones lineales $Ax = b$ lo caracterizará la sensibilidad del vector solución x a pequeños cambios, tanto en el término de la derecha b , como en los coeficientes que definen la matriz A .

Como sabemos, el ordenador o máquina que resuelve $Ax = b$, al no trabajar más que con una precisión determinada, resolverá una aproximación

$$(A + \Delta A)x = b + \Delta b.$$

Si el algoritmo utilizado es estable y el sistema también, cabe esperar que el resultado obtenido sea muy parecido al real.





Sin embargo, si el sistema está mal condicionado, o el algoritmo no es numéricamente estable, la solución puede diferir sustancialmente de la real.

De estas consideraciones se desprende la utilidad de cuantificar el condicionamiento de un sistema de ecuaciones.

Esto se consigue mediante el denominado número de condición de una matriz que veremos inmediatamente.

Antes, consideremos los dos sistemas de ecuaciones siguientes:

$$A\mathbf{x} = \mathbf{b} \rightarrow \begin{bmatrix} 8 & -5 \\ 4 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 14 \end{bmatrix} \quad \text{y} \quad \hat{A}\hat{\mathbf{x}} = \hat{\mathbf{b}} \rightarrow \begin{bmatrix} 0,66 & 3,34 \\ 1,99 & 10,01 \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 12 \end{bmatrix}.$$

La solución de ambos es el vector $[1, 1]^T$.

Si introducimos un $\Delta\mathbf{b} = [-0,04, -0,06]^T$ en el término independiente del primer sistema, su solución pasará a ser $[0,993, 0,9968]^T$.



El cambio relativo en la norma euclídea del vector \mathbf{b} es

$$\frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2} = \frac{\sqrt{0,04^2 + 0,06^2}}{\sqrt{3^2 + 14^2}} \approx 0,0050.$$

Por lo que respecta al vector solución, ese cambio relativo en la norma euclídea es

$$\frac{\|\Delta\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \frac{\sqrt{(1 - 0,993)^2 + (1 - 0,9968)^2}}{\sqrt{1^2 + 1^2}} \approx 0,0054.$$

Como se puede ver, un pequeño cambio en el vector \mathbf{b} induce un cambio pequeño en el vector solución.

Introduciendo el mismo cambio, $\Delta\hat{\mathbf{b}} = [-0,04, -0,06]^T$, en el vector en el término independiente del segundo sistema, $\hat{\mathbf{b}}$, su solución pasa a ser $[6, 0]^T$.





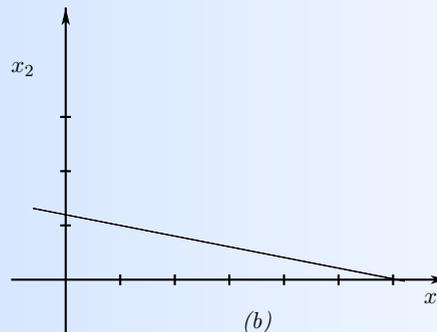
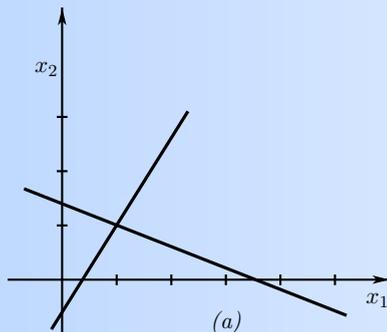
Es decir, un cambio relativo en la norma euclídea de \hat{b} igual a

$$\frac{\sqrt{0,04^2 + 0,06^2}}{\sqrt{4^2 + 12^2}} = 0,0057,$$

produce un cambio en el vector solución igual a:

$$\frac{\sqrt{5^2 + 1^2}}{\sqrt{1^2 + 1^2}} = 3,6055.$$

Evidentemente, el segundo sistema es mucho más sensible a cambios en el término independiente que el primero.





En la figura se representan geoméricamente estos dos sistemas de ecuaciones: en (a) el primero; en (b) el segundo.

Como se puede apreciar, las dos rectas que representan las ecuaciones del primer sistema se cortan nítidamente en el punto $[1, 1]^T$.

En el caso del segundo sistema, aun usando una resolución gráfica mayor, apenas se diferencian las dos rectas y mucho menos dónde se cortan.

Estudiemos cómo cuantificar la sensibilidad de un sistema de ecuaciones lineales a pequeñas modificaciones tanto en el término independiente como en los elementos de la matriz de coeficientes.

Analicemos en primer lugar el caso de una modificación $\Delta \mathbf{b}$ del término independiente. Veamos cómo se relaciona la solución de

$$A(\mathbf{x} + \Delta \mathbf{x}) = \mathbf{b} + \Delta \mathbf{b}$$



con la de

$$A\mathbf{x} = \mathbf{b}.$$

Designemos por $\|\cdot\|$ cualquier norma vectorial y su correspondiente norma matricial consistente. De las igualdades

$$A(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} + \Delta\mathbf{b} \quad \text{y} \quad A\mathbf{x} = \mathbf{b},$$

se obtiene, restando y despejando $\Delta\mathbf{x}$, que

$$\Delta\mathbf{x} = A^{-1}\Delta\mathbf{b}.$$

De la definición de norma matricial consistente con una norma vectorial (ver apéndice ??) se tiene que

$$\|\Delta\mathbf{x}\| \leq \|A^{-1}\| \|\Delta\mathbf{b}\| \tag{1}$$

y que $\|\mathbf{b}\| \leq \|A\| \|\mathbf{x}\|$ o, lo que es lo mismo, que

$$\frac{1}{\|\mathbf{x}\|} \leq \frac{\|A\|}{\|\mathbf{b}\|}. \tag{2}$$

Combinando (1) y (2) se deduce que el error relativo, $\|\Delta\mathbf{x}\|/\|\mathbf{x}\|$, de la solución del sistema $A\mathbf{x} = \mathbf{b}$ al modificar el término independiente



de \mathbf{b} a $\mathbf{b} + \Delta\mathbf{b}$ es

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \|A\| \|A^{-1}\| \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}.$$

Definición 1 Sea $\|\cdot\|$ una norma matricial consistente con una norma vectorial. Asociado a esa norma, el **número de condición de una matriz invertible** A , $\kappa(A)$, es:

$$\kappa(A) = \|A\| \|A^{-1}\|.$$

Si los datos de un sistema $A\mathbf{x} = \mathbf{b}$ son exactos hasta la precisión que permite la máquina, el error relativo de la solución que se calcule cumple que

$$\frac{\|\mathbf{x}^* - \mathbf{x}\|}{\|\mathbf{x}\|} \leq \kappa(A)\epsilon.$$

El concepto de número de condición de una matriz se generaliza a cualquier matriz A (no necesariamente cuadrada) de rango



completo mediante la expresión

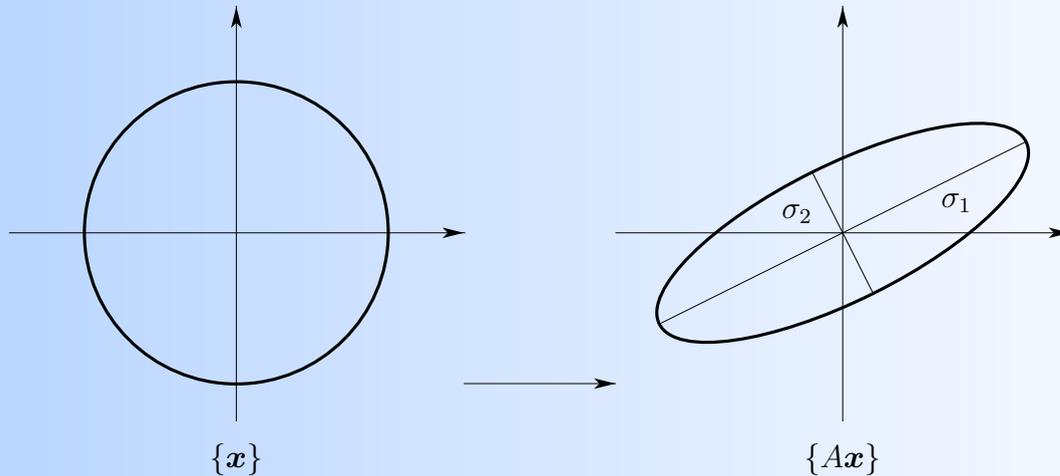
$$\kappa(A) = \|A\| \|A^\dagger\|,$$

donde A^\dagger es la **matriz pseudoinversa** de la matriz A .

El número de condición de una matriz A es un indicador del error de amplificación que produce en un vector x el someterlo a la transformación que define dicha matriz A .

Concretamente, si la esfera unidad se somete a esa transformación, el número de condición 2, asociado a la norma euclídea, será igual al cociente de las longitudes de los semiejes mayor y menor del hiperelipsoide resultante de esa transformación.





De la misma forma que hemos analizado la sensibilidad de un sistema de ecuaciones a pequeños cambios en el término independiente, se pueden estudiar cambios en los elementos de la matriz de coeficientes.

En efecto, comparemos la solución de

$$Ax = b \quad \text{y} \quad (A + \Delta A)(x + \Delta x) = b.$$

De la segunda igualdad, como $Ax = b$, haciendo $\Delta x = -A^{-1}\Delta A(x + \Delta x)$ resulta, despreciando el producto



$\Delta A \cdot \Delta \mathbf{x}$, que

$$\|\Delta \mathbf{x}\| \leq \|A^{-1}\| \|\Delta A\| \|\mathbf{x}\|.$$

Expresión que también se puede escribir como

$$\frac{\|\Delta \mathbf{x}\|}{\|\mathbf{x}\|} \leq \|A^{-1}\| \|A\| \frac{\|\Delta A\|}{\|A\|}.$$

Así pues, el error relativo que resulta de perturbar ligeramente los coeficientes de la matriz del sistema $A\mathbf{x} = \mathbf{b}$ está acotado en términos del número de condición de la matriz A .





Teorema 1 Para toda matriz A de rango completo:

1. Su número de condición $\kappa(A) \geq 1$.
2. $\kappa(A) = \kappa(A^\dagger)$.
3. $\kappa(\alpha A) = \kappa(A)$ para todo escalar $\alpha \neq 0$.
4. $\kappa_2(A) = \frac{\sigma_n(A)}{\sigma_1(A)}$, donde σ_n y σ_1 son, respectivamente, los valores singulares mayor y menor de la matriz A .
5. $\kappa_2(A) = \frac{\text{máx.}_i |\lambda_i(A)|}{\text{mín.}_i |\lambda_i(A)|}$, si A es simétrica.
6. $\kappa_2(A^T A) = \kappa_2^2(A)$.
7. Su número $\kappa_2(A) = 1$ si la matriz es la identidad o se trata de una matriz ortogonal.
8. Su número de condición $\kappa_2(A)$ es invariante frente a transformaciones ortogonales.





1. $AA^\dagger = I \Rightarrow 1 = \|I\| \leq \|A\| \|A^\dagger\|$.
2. $\kappa(A) = \|A\| \|A^\dagger\| = \|A^\dagger\| \|A\| = \kappa(A^\dagger)$.
3. $\kappa(\alpha A) = \|\alpha A\| \|(\alpha A)^{-1}\| = |\alpha| \|A\| |\alpha^{-1}| \|A^{-1}\| = \|A\| \|A^{-1}\| = \kappa(A)$.
4. Recordemos que si σ_i , $1 \leq i \leq n$, son los valores singulares de la matriz A y λ_i , $1 \leq i \leq n$, sus valores propios,

$$\|A\|_2^2 = \max_{1 \leq i \leq n} \lambda_i(A^T A) = \max_i \sigma_i^2 = \sigma_n^2$$

y

$$\|A^\dagger\|_2^2 = \max_{1 \leq i \leq n} \lambda_i((A^T A)^{-1}) = \max_i \frac{1}{\sigma_i^2} = \frac{1}{\min_i \sigma_i^2} = \frac{1}{\sigma_1^2}$$

Por lo tanto

$$\kappa_2(A) = \frac{\sigma_n(A)}{\sigma_1(A)}$$

5. Cuando A es simétrica

$$\sigma_i(A) = \left| \sqrt{\lambda_i(A^T A)} \right| = \left| \sqrt{\lambda_i^2(A)} \right| = |\lambda_i(A)|. \text{ En consecuencia,}$$

$$\kappa_2(A) = \frac{\max_i |\lambda_i(A)|}{\min_i |\lambda_i(A)|}$$





6. De la definición de valor singular, de acuerdo con el punto 4 de este teorema,

$$\kappa_2(A) = \sqrt{\frac{\text{máx.}_i \lambda_i(A^T A)}{\text{mín.}_i \lambda_i(A^T A)}}.$$

Como $A^T A$ es definida positiva, si $B = A^T A$, $B^T B = B^2$ y $\lambda(B^2) = \lambda^2(B)$.

Se tiene entonces que

$$\begin{aligned} \kappa_2(A^T A) &= \sqrt{\frac{\text{máx.}_i \lambda_i(B^2)}{\text{mín.}_i \lambda_i(B^2)}} = \sqrt{\frac{\text{máx.}_i \lambda_i^2(B)}{\text{mín.}_i \lambda_i^2(B)}} \\ &= \frac{\text{máx.}_i \lambda_i(B)}{\text{mín.}_i \lambda_i(B)} = \kappa_2^2(A). \end{aligned}$$

7. Si la matriz es ortogonal o la identidad,

$\|A\|_2^2 = \text{máx.}_{1 \leq i \leq n} \lambda_i(A^T A) = 1$, lo que implica que su número de condición κ_2 es 1.

8. Esta última propiedad se deduce inmediatamente recordando que $QQ^T = I$ y que, por tanto, $\|QA\|_2 = \|AQ\|_2 = \|A\|_2$. ■



El número de condición de una matriz indica también lo cerca que esa matriz está de la singularidad.



Volvamos al ejemplo que utilizábamos para introducir estos conceptos. La matriz

$$A = \begin{bmatrix} 8 & -5 \\ 4 & 10 \end{bmatrix},$$

cuya inversa es

$$A^{-1} = \begin{bmatrix} 0,10 & 0,05 \\ -0,04 & 0,08 \end{bmatrix},$$

tiene un número de condición

$$\kappa_1(A) = \|A\|_1 \|A^{-1}\|_1 = 15 \cdot 0,14 = 2,1.$$

El de

$$\hat{A} = \begin{bmatrix} 0,66 & 3,34 \\ 1,99 & 10,01 \end{bmatrix},$$

cuya inversa es

$$\hat{A}^{-1} = \begin{bmatrix} 250,25 & 83,5 \\ 49,75 & -16,5 \end{bmatrix},$$





es $\kappa_1(\hat{A}) = \|\hat{A}\|_1 \|\hat{A}^{-1}\|_1 = 13,35 \cdot 300 = 4005$: tres órdenes de magnitud superior.

Un error que se comete con frecuencia es asimilar el concepto de número de condición de una matriz con el de su determinante y que, en ese sentido, a mayor determinante, mayor número de condición; nada más lejos de la realidad.

Ejemplo

Sea A una matriz diagonal de orden 100 definida por

$$a_{11} = 1; \quad a_{ii} = 0,1 \quad 2 \leq i \leq 100.$$

De esta matriz, $\|A\|_2 = 1$ y $\|A^{-1}\|_2 = 10$.

El número de condición $\kappa_2(A) = 10$. Por el contrario, su determinante es $\det(A) = 1 \cdot (0,1)^{99} = 10^{-99}$.



Ejemplo

Sea A una matriz bidiagonal de la forma

$$\begin{bmatrix} 1 & 2 & & & & \\ & 1 & 2 & & & \\ & & 1 & 2 & & \\ & & & \ddots & \ddots & \\ & & & & 1 & 2 \\ & & & & & 1 \end{bmatrix}.$$

Su inversa es

$$\begin{bmatrix} 1 & -2 & 4 & \cdots & (-2)^{n-1} \\ & 1 & -2 & & (-2)^{n-2} \\ & & 1 & & \vdots \\ & & & \ddots & \vdots \\ & & & & 1 \end{bmatrix}.$$



Se tiene que

$$\|A\|_\infty = \|A\|_1 = 3 \quad \text{y} \quad \|A^{-1}\|_\infty = \|A^{-1}\|_1 = 1+2+4+\dots+2^{n-1} = 2^n - 1.$$

Los números de condición de A son:

$$\kappa_\infty(A) = \kappa_1(A) \approx 3 \cdot 2^n.$$

Su determinante en cambio es 1. ●

Los distintos números de condición de una matriz $A \in \mathfrak{R}^{n \times n}$ asociados con las normas matriciales más habituales cumplen que:

$$\begin{aligned} \kappa_2(A)/n &\leq \kappa_1(A) \leq n \kappa_2(A); \\ \kappa_\infty(A)/n &\leq \kappa_2(A) \leq n \kappa_\infty(A); \\ \kappa_1(A)/n^2 &\leq \kappa_\infty(A) \leq n^2 \kappa_1(A). \end{aligned}$$

Las matrices con números de condición pequeños, próximos a la unidad, se dicen bien condicionadas; las que tienen números de condición altos, mal condicionadas.



Cálculo del número de condición



48/48

Calcular directamente el número de condición de una matriz es una **tarea muy costosa** desde el punto de vista numérico: $\mathcal{O}(n^3)$.

Dado que, en general, $\kappa(A)$ da un diagnóstico de las implicaciones que el uso de esa matriz puede comportar, su cálculo no suele ser necesario; puede que sí la estimación de su orden de magnitud.

Existen diversos algoritmos para estimar el número de condición de una matriz que necesitan $\mathcal{O}(n^2)$ operaciones.

En Matlab existen varias funciones que lo estiman: `condest` y `rcond`; y otros que lo evalúan directamente, `cond`.

Siempre que se usa el operador “\”, Matlab usa `rcond` para comprobar la condición de la matriz afectada.



Full Screen

Imprimir