MATEMÁTICA UNIVERSITÁRIA
nº28 junho 2000 - pp. 77-92

Aproximando soluções de $AX = b$ com o MATLAB

Hamilton F. Leckar e Rubens Sampaio

Resumo

Dado um sistema linear geral, expresso na forma matricial $AX = b$, uma das três possibilidades pode ocorrer: o sistema *tem uma única solução*, *tem uma infinidade de soluções* ou *não tem solução*. Por outro lado, ao se procurar resolver um sistema linear usando a função interna "inversão à esquerda" do pacote de computação científica MATLAB, obtemos uma resposta, mesmo no caso em que o sistema não tem solução. Para entender a resposta dada pelo MATLAB, usamos um programa que chamaremos de solsis, feito no MATLAB, que simula uma possível resolução à mão, baseado na decomposição de uma matriz em seus valores singulares. Para explicar a resposta dada pelo MATLAB, somos levados a considerar um novo problema, na verdade, uma extensão não linear, que tem uma única solução. Trata-se do seguinte problema de otimização: " Achar o vetor coluna X, de coordenadas $x_1, \ldots, x_n$, mais próximo da origem, tal que os números $b_1^*, \ldots, b_m^*$ dados por $b_1^* = a_{11}\,x_1 + \ldots + a_{1n}\,x_n, \ldots, b_m^* = a_{m1}\,x_1 + \ldots + a_{mn}\,x_n$ sejam as componentes do vetor coluna mais próximo de b". Em seguida, usando o MATLAB, aproximamos a solução desse novo problema, a partir da *peudo-inversa* da matriz do sistema correspondente.

1 Sistemas Lineares

Iniciaremos com um sistema linear geral

$$\left\{ \begin{array}{ccccccc} a_{11}\,x_1 & + & \ldots & + & a_{1n}\,x_n & = & b_1 \\ a_{21}\,x_1 & + & \ldots & + & a_{2n}\,x_n & = & b_2 \\ & & & & & \vdots & \\ a_{m1}\,x_1 & + & \ldots & + & a_{mn}\,x_n & = & b_m \end{array} \right. \tag{1}$$

de m equações nas n incógnitas $x_1, \ldots, x_n$.

Classicamente o problema importante que se coloca para êsse sistema é o seguinte:

"Dadas as constantes reais $a_{11}, \ldots, a_{1n}, a_{21}, \ldots, a_{2n}, \ldots, a_{mn}$; $b_1, \ldots, b_m$, achar números reais $x_1, \ldots, x_n$ para os quais todas as equações em (1) são simultaneamente verdadeiras."

Um tal conjunto $x_1, \ldots, x_n$ satisfazendo (1) é chamado *solução.*

Um sistema linear em geral pode ser *possível ou consistente* ou pode ser *impossível ou inconsistente.*

Definição 1.1 Sistema possível *é aquele que tem ao menos uma solução. Além disso, um sistema é*

- possível determinado, *se tem uma única solução.*
- possível indeterminado, *se tem mais de uma solução.*

Sistema impossível *é aquele que não tem solução.*

O MATLAB opera essencialmente com matrizes. Daí, o primeiro passo é transformar nosso sistema linear de m equações e n incógnitas numa equação matricial equivalente. O sistema é escrito como uma única equação, onde as operações de adição e multiplicação em cada equação do sistema é traduzida como operações correspondentes com matrizes.

Deste modo, o sistema (1) se escreve na forma matricial como

$$AX = b$$

onde

$$A = \begin{bmatrix} a_{11} & \ldots & a_{1n} \\ a_{21} & \ldots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \ldots & a_{mn} \end{bmatrix}, \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \text{ e } b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

A é a matriz do sistema, podendo ser quadrada ou retangular; b é o termo independente do sistema, tem o mesmo número de linhas que A; X, a matriz incógnita do sistema, tem tantas linhas quantas são as colunas de A.

Diversos são os métodos que podemos empregar para estudar um sistema linear dado. Dentre eles, existe um teorema clássico de *Rouché-Capelli*, que nos permite decidir quando um dado sistema linear é possível ou impossível e, se possível, saber se é determinado ou não. Ainda mais, no caso de um sistema "quadrado"(quando o número de equações é igual ao número de incógnitas) *possível determinado*, há a bem conhecida *regra de Cramer* para calcular explicitamente sua única solução.

Hoje em dia, por questões práticas, é preciso resolver sistemas lineares com um grande número de equações e incógnitas. Neste caso, resolução analítica é uma tarefa, no mínimo, desanimadora. Por isso, precisamos escolher um método adequado e usar o computador. Veja mais detalhes em [11, seção 3.2.1] ou [7].

1.1 Aproximação da solução com o MATLAB

Quando a matriz A do sistema é quadrada e inversível, a solução é dada formalmente por $X = A^{-1}\,b$. No entanto, inverter a matriz A usando, por exemplo, a regra de Cramer, além do número de operações tornar-se computacionalmente impraticável em função do tamanho da matriz, não é uma tarefa computacionalmente muito estável.

Achar $X = A^{-1}b$ com o MATLAB corresponde à operação $X = A\setminus b$ a qual usa eliminação Gaussiana e não calcula a matriz A^{-1} diretamente. Quando a matriz A não é quadrada, a operação $A\setminus b$ varia de acordo com as propriedades da matriz A. Por exemplo, se as colunas de A são linearmente independentes, $A\setminus b$ usa a decomposição QR de A. Veja o comentário B.1 no final do apêndice B.

Podemos observar que a resposta obtida para um dado sistema $AX = b$ usando a função $A\setminus b$ depende do tipo da matriz A:

- A quadrada inversível: $A\setminus b$ é uma aproximação da solução.
- A quadrada não inversível: $A\setminus b$ é uma aproximação da solução ou não.
- A não quadrada com mais linhas do que colunas e o sistema possível determinado: $A\setminus b$ é uma aproximação da solução.

Na tabela 1, a resposta dada pela função $A\setminus b$ a alguns exemplos podem ser comparadas com as respostas dada pelo programa solsis, descrito no apêndice B.

2 Extensão e solução no sentido de mínimos quadrados

Vamos escrever cada coluna $AX \in \mathbb{R}^m$ como uma combinação linear dos vetores colunas de A, tendo as coordenadas $x_1, \ldots, x_n$ do vetor coluna $X \in \mathbb{R}^n$ como coeficientes. O conjunto de tais colunas obtidas, denotado por $Im(A)$, é chamado de *conjunto imagem* de A é identificado à imagem da transformação linear $T_A : \mathbb{R}^n \to \mathbb{R}^m$ associada a matriz A nas bases canônicas desses espaços. Deste modo, um sistema $AX = b$ terá solução se a coluna dada b pertencer ao conjunto imagem de A, caso contrário, não tem solução. Na primeira hipótese, o sistema terá solução única se as colunas de A forem linearmente independentes, caso contrário, terá uma infinidade de soluções. Veja mais detalhes sobre interpretação geométrica de sistemas em [7].

O problema geral enunciado na introdução, expresso na forma matricial $AX = b$, terá *solução única* e sua aproximação numérica poderá ser dada por uma operação simples com o MATLAB.

Em particular:

- Se o sistema $AX = b$ é possível determinado, a solução do problema geral será a própria solução do sistema.
- Se o sistema $AX = b$ é possível indeteminado, a solução do problema geral será aquela de norma mínima, dentre todas as soluções do sistema $AX = b$.
- Se o sistema dado é impossível, a solução do problema geral dependerá da solução do sistema $AX = b^*$, onde b^* é o elemento do conjunto imagem de A tal que a norma de $b - b^*$ é mínima. Neste caso, $AX = b^*$ é sempre possível.
  - Se o sistema $AX = b^*$ é determinado, sua única solução é a solução do problema geral.
  - Se o sistema $AX = b^*$ é indeterminado, a solução do problema geral é aquela de norma mínima, dentre todas as soluções de $AX = b^*$.

Passos da Extensão

Passo 1: Escolher o vetor b^* de $Im(A)$ *mais próximo* de b, ou seja, que torna mínima a distância $\|b^* - b\|$.

Passo 2: Escolher o vetor X de *norma mínima* que satisfaz o sistema $AX = b^*$.

Assim, combinando estes procedimentos, chegamos a um problema bem determinado:

Problema 2.1 *Achar o vetor coluna X de* norma mínima *que* minimiza o erro $\mathbf{e}(Z) = \|AZ - b\|$, *dentre todos os vetores coluna Z.*

Observação 2.1 *Pode acontecer, em função da aplicação que se tenha, de a norma ou noção de distância mais adequada ao problema em questão ser diferente da norma euclideana usual.*

2.1 Decomposição em valores singulares e pseudo-inversa com o MATLAB

É bastante interessante observar que existe uma noção de inversa generalizada[1] de uma matriz A de qualquer tipo, que resolve prontamente o problema acima descrito.

[1]dentre várias outras. Veja por exemplo em [8] ou [1].

Trata-se da inversa generalizada de Moore-Penrose ou pseudo-inversa da matriz A.

Definição 2.1 *Dada uma matriz A, retangular de tipo $m \times n$, chamamos* inversa generalizada de Moore-Penrose ou pseudo-inversa *de A a qualquer matriz B de tipo $n \times m$ tendo as quatro propriedades seguintes:*

$$(i)\ A\,B\,A = A \qquad (iii)\ (B\,A)^T = B\,A$$
$$(ii)\ B\,A\,B = B \qquad (iv)\ (A\,B)^T = A\,B\,,$$

onde $(\cdot)^T$ denota a transposta.

Pode ser provado que existe uma única matriz B com estas propriedades[2] a qual denotaremos por $A^\dagger$. A pseudo-inversa $A^\dagger$ pode ser calculada com a função interna `pinv` do MATLAB.

Dada uma matriz A e um vetor coluna b com o mesmo número de linhas que A, a solução do problema 2.1 é

$$X = A^\dagger\, b\,.$$

Este resultado é formalizado de forma equivalente na proposição seguinte.

Proposição 2.1 *$X = A^\dagger\, b$ é o único vetor coluna com a seguinte propriedade:*

1. $(A\,X - b)^T\, Y = \mathbf{0}$, *para todo $Y \in Im(A)$*
2. *Se $(A\,Z - b)^T\, Y = \mathbf{0}$, para todo $Y \in Im(A)$, então $X^T\, X \leq Z^T\, Z$.*

Uma prova dessa proposição é dada no apêndice A.

Observação 2.2 (Problemas de mínimos quadrados) *1. O ponto X do hiperplano de equação $a_1x_1 + a_2x_2 + \ldots + a_nx_n = c$ mais próximo do ponto b de $\mathbb{R}^n$ pode ser calculado simplesmente como o produto $X =$* `pinv`$([a_1a_2\ldots a_n])\ (b - X_0)$, *onde X_0 é um ponto escolhido arbitrariamente no hiperplano.*

2. Sendo os m pontos

$$(a_{11}, a_{12}, \ldots, a_{1n}, b_1)\,,\ (a_{21}, a_{22}, \ldots, a_{2n}, b_2)\,,\ \ldots\ ,\ (a_{m1}, a_{m2}, \ldots, a_{mn}, b_m)\,,$$

dados, onde cada b_j é obtido experimentalmente em função dos $a_{ij}\,, i = 1, \ldots, n$, o hiperplano de $\mathbb{R}^{n+1}$ de equação $x_{n+1} = \alpha_1x_1 + \alpha_2x_2 + \ldots + \alpha_nx_n + \beta$ que melhor ajusta esses dados, no sentido de mínimos quadrados, isto é, minimiza:

$$\begin{aligned} & (a_1\alpha_{11} + a_2\alpha_{12} + \ldots + a_n\alpha_{1n} + \beta - b_1)^2 \\ + \quad & (a_1\alpha_{21} + a_2\alpha_{22} + \ldots + a_n\alpha_{2n} + \beta - b_2)^2 \\ + \ldots + \quad & (a_1\alpha_{m1} + a_2\alpha_{m2} + \ldots + a_n\alpha_{mn} + \beta - b_m)^2\,, \end{aligned}$$

[2] Veja a fórmula (2) do apêndice A.

poderá ser calculado pela fórmula

$$[\alpha_1 \alpha_2 \ldots \alpha_n]^T = \mathtt{pinv}(A)\ [b_1 b_2 \ldots b_m]^T ,$$

onde $A_{ij} = a_{ij}\,; A_{i\ n+1} = 1$ *se* $1 \leq i \leq m\,, 1 \leq j \leq n\,.$

Na tabela 1 temos a solução do problema 2.1 para alguns exemplos com a função pinv do MATLAB .

2.1.1 Aproximação da solução do problema 2.1 com o MATLAB, usando a pseudo-inversa:

Usamos o MATLAB para, através da decomposição em valores singulares , fazer um programa que aproxima a solução de um sistema $AX = b$, com um procedimento análogo ao que faríamos se buscássemos resolvê-lo analiticamente, sem o uso de um computador. Usamos a decomposição em valores singulares, ao invés de outra decomposição de A, porque esta decomposição se aplica a matrizes de todos os tipos e porque ela está ligada à pseudo-inversa de A, através da Eq. (2) do apêndice A.

A idéia do programa é a seguinte: Usando a decomposição em valores singulares de A, conforme a equação (4) do apêndice B, escrevemos $A = U\,S\,V^T$ onde as matrizes U e V são ortogonais, isto é, $U^T U = U U^T = I_{m\times m}$ e $V^T V = V V^T = I_{n\times n}$ e, além disso, S é uma matriz retangular diagonal, conforme (6), (7) do Apêndice B. Daí, $A\,X = b$ é escrita como $U\,S\,V^T X = b$, ou ainda, $S\,V^T X = U^T b$. Então, resolvemos primeiro o sistema $S\,Z = c$, onde $c = U^T b$ e $Z = V^T X$ e em seguida, temos $X = V\,Z$.

Programa SOLSIS com o MATLAB

É um programa para aproximar soluções de sistemas lineares $AX = b$ usando a decomposição em valores singulares de A, sendo A uma matriz m por n.

```
function solsis(A,b)
[m,n]=size(A);
[U,S,V]=svd(A);
C=U'*b;
k=rank(S);
if m==n & k==n
disp('O sistema tem uma unica solucao:');
X=V*inv(S)*C
elseif m==n & k<n
if max(abs(C(k+1:n)))> 10^(-9);
disp('O sistema nao tem solucao.');
else max(abs(C(k+1:n)))<=10^(-9)
```

```
disp('O sistema tem uma infinidade de solucoes. Uma dessas e');
Y=inv(S(1:k,1:k))*C(1:k);
X=V(:,1:k)*Y
disp('e a solucao geral e  X + combinacao linear arbitraria da(s)
coluna(s) de');
M=V(:,k+1:n)
end;
elseif m<n & k==m
disp('O sistema tem uma infinidade de solucoes. Uma dessas e');
Y=inv(S(1:m,1:m))*C(1:m);
X=V(:,1:m)*Y
disp('e a solucao geral e X + combinacao linear arbitraria da(s)
coluna(s) de'):
M=V(:,m+1:n)
elseif m<n & k<m
if max(abs(C(k+1:m)))> 10^(-9);
disp('O sistema nao tem solucao.');
else max(abs(C(k+1:m)))<=10^(-9)
disp('O sistema tem uma infinidade de solucoes. Uma dessas e');
Y=inv(S(1:k,1:k))*C(1:k);
X=V(:,1:k)*Y
disp('e a solucao geral e X + combinacao linear arbitraria da(s)
coluna(s) de');
M=V(:,k+1:n)
end;
elseif m>n & k==n
if max(abs(C(k+1:m)))> 10^(-9);
disp('O sistema nao tem solucao.');
else max(abs(C(k+1:m)))<=10^(-9)
display('O sistema tem uma unica solucao:');
Y=inv(S(1:n,1:n))*C(1:n);
X=V*Y
end;
else m>n & k<n
if max(abs(C(k+1:m)))> 10^(-9);
disp('O sistema nao tem solucao.');
else max(abs(C(k+1:m)))<=10^(-9)
disp('O sistema tem uma infinidade de solucoes. Uma dessas e');
Y=inv(S(1:k,1:k))*C(1:k);
X=V(:,1:k)*Y
disp('e a solucao geral e X + combinacao linear arbitraria da(s)
coluna(s) de');
```

```
M=V(:,k+1:n)
end;
end;
```

Vamos considerar na tabela seguinte, quatro sistemas $AX = b$ onde em cada caso, A e b são:

$$(1) \qquad A = \begin{bmatrix} 0,96 & 1,72 \\ 2,28 & 0,96 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 2 \end{bmatrix};$$

$$(2) \qquad A = \begin{bmatrix} 0,96 & 1,72 & 0,76 \\ 2,28 & 0,96 & -1,32 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 2 \end{bmatrix};$$

$$(3) \qquad A = \begin{bmatrix} 0,96 & 1,72 \\ 2,28 & 0,96 \\ 1,32 & -0,76 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix};$$

$$(4) \qquad A = \begin{bmatrix} 0,96 & 1,72 & 0,76 \\ 2,28 & 0,96 & -1,32 \\ 1,32 & -0,76 & -2,08 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}.$$

Tabela 1

$AX = b$	**Solução**	$A\backslash b$	**solsis**	**pinv**
(1)	$0,82666\ldots$ $0,12000\ldots$	$0,8267$ $0,1200$	$0,8267$ $0,1200$	$0,8267$ $0,1200$
(2)	$0,82666\ldots + c$ $0,12000\ldots - c$ $0\ldots + c$	$0,8267$ $0,1200$ 0	$0,5911 + c0,5774$ $0,3556 - c0,5774$ $-0,2356 + c0,5774$	$0,5911$ $0,3556$ $-0,2356$
(3)	não tem	$1,4222$ $-0,6000$	não tem	$1,4222$ $-0,6000$
(4)	não tem	Atenção: Res. impreciso	não tem	$0,7481$ $0,0741$ $-0,6741$

3 Conclusão

O problema $AX = b$ é um dos mais importantes da Matemática. Em cursos elementares ensina-se que se A for inversível então $A^{-1}b$ é a única solução.

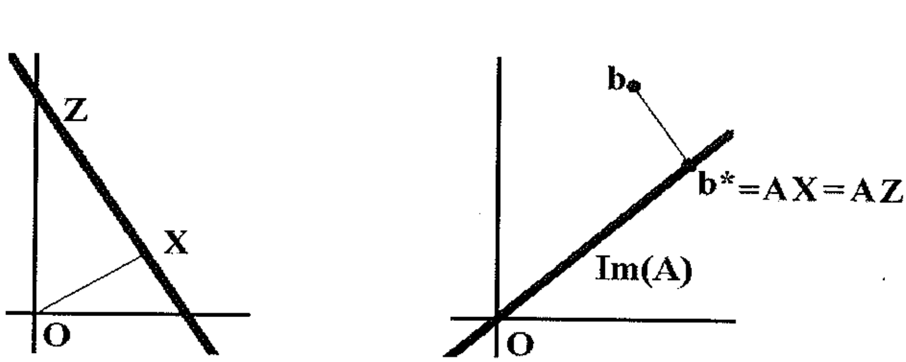
No caso de A ser não inversível ensina-se geralmente, a partir de operações elementares, o método de escalonamento e retro-substituição para resolver o sistema, o que é um processo penoso, mesmo se A é 4×4. Portanto, o uso de um pacote, tipo MATLAB, se faz necessário.

O que procuramos mostrar nestas notas é que, mesmo usando um pacote poderoso como o MATLAB, a reflexão sobre o significado da resposta é imprescindível. Não é possível usar pacotes sem refletir sobre o que se está calculando. É necessário compreender como as respostas são encontradas e com que precisão são calculadas.

Na verdade, só é possível usar pacotes com segurança quando se conhece em profundidade as teorias matemáticas envolvidas e as aproximações que estão sendo feitas.

Pacotes não são substitutos para teorias, e sem estas, os pacotes não são confiáveis.

A Pseudo-inversa e o problema de mínimos quadrados



Vemos na figura que b^* é o único vetor de $Im(A)$ que torna mínimo o erro $\mathbf{e}(X) = ||AX - b||$. Podemos caracterizar b^* por meio da condição seguinte:

$$\text{Para todo } Y \in Im(A), \quad Y^T(b^* - b) = 0$$

que traduz o fato de b^* ser o único vetor coluna de $Im(A)$ tal que $b - b^*$ seja ortogonal à $Im(A)$.

Prova da proposição 2.1:

Vamos provar que, dada uma matriz A e um vetor coluna b com o mesmo número de linhas que A, $X = A^\dagger b$ é o único vetor coluna com a seguinte propriedade:

1. $(A\,X - b)^T\,Y = \mathbf{0}$, para todo $Y \in Im(A)$;
2. Se $(A\,Z - b)^T\,Y = \mathbf{0}$, para todo $Y \in Im(A)$, então $X^T\,X \leq Z^T\,Z$.

Prova: Seja $X = A^\dagger\,b$. Então,

$$\begin{aligned} A^T\,A\,X &= A^T\,A\,(A^\dagger\,b) = A^T\,(A\,A^\dagger)\,b \\ &= A^T\,(A\,A^\dagger)^T\,b = A^T\,(A^\dagger)^T\,A^T\,b = (A\,A^\dagger\,A)^T\,b = A^T\,b\,. \end{aligned}$$

Isto prova o item 1 e mostra também que $A\,(A^\dagger b) = b^*$. Em resumo, temos $A\,X = b^*$ e $||A\,X - b|| = \text{mín}_Z||A\,Z - b||$.

Falta mostrar que $X = A^\dagger\,b$ tem norma mínima. Para isto, observamos que se $AZ \in Im(A)$, isto é, se $A\,Z = b^*$, então podemos escrever $Z = h + A^\dagger\,b$ com h satisfazendo $A\,h = 0$.

Mostremos que $Z^T\,Z \geq (A^\dagger\,b)^T\,A^\dagger\,b$. Com efeito, multiplicação direta nos leva a $Z^T\,Z = (A^\dagger\,b)^T\,A^\dagger\,b + h^T\,h + 2\,h^T\,A^\dagger\,b$.

Usando as propriedades de $A^\dagger$ temos que $h^T\,A^\dagger\,b = 0$. De fato,

$$\begin{aligned} &(h^T\,A^\dagger\,b)^T = (h^T\,(A^\dagger\,A\,A^\dagger)\,b)^T = ((h^T\,A^\dagger\,A)\,(A^\dagger\,b))^T = (A^\dagger\,b)^T\,(h^T\,A^\dagger\,A)^T \\ &= (A^\dagger\,b)^T\,(A^\dagger\,A)^T\,h = (A^\dagger\,b)^T\,(A^\dagger\,A)\,h = (A^\dagger\,b)^T\,A^\dagger\,(A\,h) = 0 \end{aligned}$$

onde usamos $A\,h = 0$. Assim, $Z^T Z = X^T X + h^T h \geq X^T X$. ■

A pseudo-inversa de A

Tomamos a decomposição em valores singulares de A: $A = U\,S\,V^T$, conforme a equação (4) do apêndice B. A *peudo-inversa* $A^\dagger$ tem a seguinte caracterização:

$$A^\dagger = V\,\tilde{S}\,U^T \tag{2}$$

sendo $\tilde{S}$ a matriz n por m obtida da transposta S^T de S, substituindo-se os elementos não nulos S_{ii} de sua diagonal por seus inversos $1/S_{ii}$.

B Decomposição em valores singulares para matrizes reais

Nesta seção, estudamos uma decomposição para uma matriz retangular, a qual tem grande importância em álgebra linear computacional, denominada *decomposição em valores singulares* .

Tomemos uma matriz $m \times n$ A e denotemos por $m \wedge n$, o menor dos inteiros m e n.

A grosso modo, podemos descrever a decomposição em valores singulares de A, da seguinte maneira:

Existem matrizes unitárias U e V de tipos m e n respectivamente, tais que A se escreve na forma

$$A\,V = U\,S\,. \tag{3}$$

onde S é uma matriz $m \times n$ cuja diagonal é constituída por números reais não negativos $\sigma_1, \sigma_2, \dots, \sigma_{m\wedge n}$ e todas as suas demais entradas são nulas.

Em particular, temos

$$A = U\,S\,V^T\,. \tag{4}$$

Mais explicitamente, escrevamos $V = [v_1, v_2, \dots, v_n]$, através de suas colunas $v_i \in \mathbb{R}^n$ e analogamente, $U = [u_1, u_2, \dots, u_m]$, com $u_i \in \mathbb{R}^m$. Neste caso, a equação (3) para $m \wedge n = n$, se escreve como

$$\begin{aligned} A\,v_1 &= \sigma_1\,u_1 \\ A\,v_2 &= \sigma_2\,u_2 \\ &\vdots \\ A\,v_{m\wedge n} &= \sigma_{m\wedge n}\,u_{m\wedge n} \end{aligned} \tag{5}$$

Se $m \wedge n = m$, a equação (3) equivale às equações (5) juntamente com as equações seguintes

$$\begin{aligned} A\,v_{m+1} &= \mathbf{0} \\ A\,v_{m+2} &= \mathbf{0} \\ &\vdots \\ A\,v_n &= \mathbf{0} \end{aligned}$$

ou seja, as últimas $n - m$ colunas em (3) são nulas.

A título de ilustração, notamos que se $m \wedge n = m$, a matriz S tem a forma seguinte:

$$\begin{bmatrix} \sigma_1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_m & 0 & \dots & 0 \end{bmatrix} \tag{6}$$

No caso em que $m \wedge n = n$, a matriz S tem a forma:

$$\begin{bmatrix} \sigma_1 & 0 & \ldots & 0 \\ 0 & \sigma_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \sigma_n \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 \end{bmatrix} \tag{7}$$

Os números reais não negativos $\sigma_1, \sigma_2, \ldots, \sigma_{m\wedge n}$ são denominados *valores singulares de A*. Os vetores $v_1, v_2, \ldots, v_m$ são denominados *vetores singulares à direita* de A, enquanto que $u_1, u_2, \ldots, u_n$ denominam-se *vetores singulares à esquerda de A*.

Observação B.1 *A partir de* (3), *podemos deduzir imediatamente as seguintes propriedades:*

i) O posto *da matriz A,* **posto**(A) *(dimensão da imagem de A), é igual ao número de valores singulares σ_i estritamente positivos de A (Veja (6) e (7)).*

ii) Se denotamos, contando multiplicidade, os auto-valores não nulos de $A^T A$, em ordem decrescente, por $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r > 0$, temos de (3),

$$A\,V = U\,S\,, \quad V^T A^T = S^T U^T$$

e daí,

$$V^T A^T A\,V = S^T U^T U\,S = S^T\,S \tag{8}$$

e

$$A\,A^T = A\,V\,V^T A^T = U\,S\,S^T U^T\,,$$

ou seja,

$$U^T A\,A^T U = S\,S^T\,. \tag{9}$$

Assim, de (8) *e* (9), *temos que os auto-valores não nulos de $A^T A$ e $A\,A^T$ são os mesmos, e na verdade coincidem com os quadrados dos valores singulares de A:*

$$\lambda_1 = {\sigma_1}^2 \geq \lambda_2 = {\sigma_2}^2 \geq \ldots \geq \lambda_r = {\sigma_r}^2 > 0\,.$$

iii) O número de condicionamento *de uma matriz $n\times n$ inversível A,* **cond**(A), *é definido como*

$$\mathbf{cond}(A) = ||A||_2\,||A^{-1}||_2$$

onde
$$\|A\|_2 = \sup_{X \neq 0} \frac{\|A\,X\|}{\|X\|}\,.$$

Mas, de $A = U\,S\,V^T$, *vem* $A^{-1} = V\,S^{-1}\,U^T$.

Sendo
$$\|A\|_2 = \sigma_1 = \textit{ maior valor singular de } A$$

e
$$\|A^{-1}\|_2 = 1/\sigma_n = \textit{ inverso do menor valor singular de } A$$

temos
$$\mathbf{cond}(A) = \sigma_1/\sigma_n\,.$$

As propriedades acima são decorrentes do teorema seguinte.

Teorema B.1 (Decomposição em valores singulares) *Seja A uma matriz $m \times n$ arbitrária, com coeficientes reais. Então existem matrizes ortogonais U e V, de tipos m e n respectivamente, e números reais $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{m \wedge n} \geq 0$ tais que*
$$U^T\,A\,V = S$$
onde S é a matriz $m \times n$ definida por
$$\begin{cases} S_{ij} = \sigma_i & i = j \\ \\ S_{ij} = 0 & i \neq j\,. \end{cases}$$

Prova: (Indução sobre a ordem de A) Seja $\sigma_1 := \|A\|_2 = \sup_{\|X\|=1} \|A\,X\|$.

Como a esfera unitária de $\mathbb{R}^n$, $\mathbf{S} = \{X \in \mathbb{R}^n : \|X\| = 1\}$ é compacta, segue do teorema de Weierstrass que existe um vetor unitário $v_1 \in \mathbb{R}^n$ satisfazendo $\|A\|_2 = \|A\,v_1\|$. Se $A \neq \mathbf{0}$, $\sigma_1 > 0$. (Por quê?)

Observamos que σ_1 corresponde ao comprimento do maior semi-eixo do elipsóide generalizado $\mathbf{E} = \{A\,X \in \mathbb{R}^m : \|X\| = 1\}$.

Obviamente estamos supondo $A \neq \mathbf{0}$. Caso contrário, a decomposição em valores singulares de A pode ser obtida simplesmente com $S = \mathbf{0}$ e quaisquer U e V ortogonais de tipos m e n respectivamente.

Assim, a primeira equação em (5) pode ser imediatamente satisfeita se definimos o vetor unitário u_1 por $u_1 = \dfrac{1}{\sigma_1} A\,v_1$.

Para obtermos o passo de indução, precisamos mostrar que o complemento ortogonal de $\mathbb{R} \cdot \{v_1\} = \{\alpha v_1 : \alpha \in \mathbb{R}\}$ em $\mathbb{R}^n$, tem como imagem por A em $\mathbb{R}^m$, um subespaço ortogonal a $\mathbb{R} \cdot \{u_1\} = \{\beta u_1 : \beta \in \mathbb{R}\}$.

Tomemos $w_2 \in \mathbb{R}^n$ unitário, ortogonal a v_1 ($v_1^T\,w_2 = 0$) e mostremos que $A\,w_2$ é ortogonal a u_1.

Para isto, definamos $w = \sigma_1 v_1 + (u_1^T A w_2) w_2$ e denotemos $u_1^T A w_2 := \delta$. Neste caso, aplicando A, obtemos $A w = \sigma_1{}^2 u_1 + \delta A w_2$. Como w_2 foi tomado unitário, vemos que

$$||w||^2 = \sigma_1{}^2 + \delta^2 ||w_2||^2 = \sigma_1{}^2 + \delta^2 . \tag{10}$$

Por outro lado,

$$\begin{aligned} ||Aw||^2 &= \sigma_1{}^4 + 2\sigma_1{}^2 \delta^2 + \delta^2 ||A w_2||^2 \\ &\geq \sigma_1{}^4 + 2\sigma_1{}^2 \delta^2 + \delta^4 \\ &= (\sigma_1{}^2 + \delta^2)^2 = (\sigma_1^2 + (u_1^T A w_2)^2)^2, \end{aligned}$$

visto que $|u_1^t A w_2| \leq ||Aw_2||$ (desigualdade de Cauchy-Schwarz).

Da definição de σ_1 e da Eq. (10) temos

$$\sigma_1{}^2 \geq \frac{||A w||^2}{||w||^2} \geq \sigma_1{}^2 + (u_1^T A w_2)^2$$

e portanto, temos $u_1^T A w_2 = 0$, ou seja, $A w_2$ é ortogonal a u_1.

Observamos que o mesmo resultado é obtido se w_2 não é unitário.

A partir desse resultado, podemos proceder indutivamente para obter a decomposição em valores singulares de A, determinando o segundo valor singular σ_2 de A. Aplicamos o raciocínio anterior à restrição de A ao complemento ortogonal de v_1 em $\mathbb{R}^n$, cuja imagem está contida no complemento ortogonal de u_1 em $\mathbb{R}^m$.

Notamos daí que $\sigma_2 \leq \sigma_1$, e assim sucessivamente. As matrizes V e U são tomadas, por exemplo, como sendo $V = [v_1, v_2, \ldots, v_n]$ e $U = [u_1, u_2, \ldots, u_m]$, onde se $m \wedge n = m$, então $v_{m+1}, \ldots, v_n$ são quaisquer tais que V é ortogonal, e se $m \wedge n = n$, então $u_{n+1}, \ldots, u_m$ são quaisquer tais que U é ortogonal.

Os σ_i são todos univocamente determinados por A, enquanto que o mesmo não ocorre com as matrizes V e U.

Podemos então, dar por encerrada a prova do teorema. ■

Comentário B.1 *A solução de mínimos quadrados no sentido da norma usual pode também ser calculada a partir da decomposição QR da matriz A. Para simplificar, suponha que A seja de tipo $m \times n$ com $m \geq n$ e posto n. Existe uma matriz ortogonal Q de tipo m e uma matriz triangular superior R tal que $QR = A$, onde R é da forma*

$$\begin{bmatrix} R_1 \\ \mathbf{0} \end{bmatrix}$$

com R_1 de tipo $n \times n$ e posto n.

Dado b de tipo $m \times 1$, seja $Q^T b = \begin{bmatrix} c \\ d \end{bmatrix}$ *com c de tipo $n \times 1$. Temos então,*

$$\begin{aligned} \|AX - b\|^2 &= \|Q(RX - Q^T b)\|^2 \\ &= \|Q\|^2 \left\| \begin{bmatrix} R_1 X \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} c \\ d \end{bmatrix} \right\|^2 = \|R_1 X - c\|^2 + \|d\|^2 . \end{aligned}$$

Temos como conseqüência que a solução de mínimos quadrados é a única solução de $R_1 X = c$. ■

Referências

[1] G.H. Golub, C.F. Van Loan, *Matrix Computations*, Johns Hopkins Series in the Mathematical Sciences **3** **(1996)**.

[2] K. Hoffmann, R. Kunze, *Álgebra Linear*, LTC, Rio de Janeiro (**1979**).

[3] R. A. Horn, C. R. Johnson, *Matrix Analysis*, Cambridge Univ. Press (**1985**).

[4] R. A. Horn, C. R. Johnson, *Topics in Matrix Analysis*, Cambridge Univ. Press (**1991**).

[5] T. Lawson, *Álgebra Linear*, Edgard Blücher, (**1997**).

[6] H. F. Leckar, R. Sampaio, *Problemas de Auto-valores Generalizados e Fatoração de Matrizes*, Relatório Técnico, ILTC- Pós-Grad. Matemática -UFF (**1982**).

[7] E. L. Lima, *Sobre o ensino de sistemas lineares*, Revista do Professor de Matemática,(23) pp. 8-18 (**1993**).

[8] E. L. Lima, *Álgebra Linear*, Coleção Matemática Universitária, IMPA, Rio de Janeiro, 1999.

[9] E. H. Moore, *On the reciprocal of the general algebraic matrix*, Bull. Amer. Math. Soc. **26 (1920)**, pp. 394-395.

[10] R. Penrose, *A generalized inverse for matrices* , Proc. Cambridge Philos. Soc. **51 (1955)**, pp. 406-413.

[11] M. A. G. Ruggiero, V. L. da R. Lopes, *Cálculo Numérico, Aspéctos Teóricos e Computacionais*, McGraw-Hill, S.Paulo (**1988**).

[12] R. Sampaio, E. Cataldo, R. Riquelme, *Introdução ao MATLAB*, PUC-Rio (**1997**). Disponível na página: http://www.mec.puc-rio.br/prof/rsampaio/rsampaio.html

[13] D. Watkins, *Foundations of Matrix Computations*, Wiley (**1991**).

[14] The Math Works, MATLAB User's Guide, (**1994**).

Hamilton F. Leckar
Departamento de Matemática Aplicada-UFF
Rua Mário Santos Braga, s/n, Niterói
24020-140 - RJ, Brasil.
E-mail: hfleckar@vm.uff.br

Rubens Sampaio
Departamento de Engenharia Mecânica-PUC-Rio
Rua Marquês de São Vicente, 225
22453-900 Rio de Janeiro - RJ, Brasil)
E-mail: rsampaio@mec.puc-rio.br
Homepage: http://www.mec.puc-rio.br/prof/rsampaio/rsampaio.html