Universidade do Minho
Departamento de Matemática

MMC

ANCP

Ficha de trabalho 2 — 2023/2024

1. Dada uma matriz real $A \in \mathbb{R}^{m\times n}$ e um vetor $\boldsymbol{b} \in \mathbb{R}^m$, o problema dos mínimos quadrados consiste em determinar um vetor $\boldsymbol{x} \in \mathbb{R}^n$ para o qual a função $r(\boldsymbol{x}) = \|A\boldsymbol{x} - \boldsymbol{b}\|_2$ assume um valor mínimo.

(a) Se $m = n$ e A é uma matriz invertível, a solução do sistema obtida por eliminação Gaussiana é igual à solução que se obtém usando o método dos mínimos quadrados (admitindo uma aritmética exata)? Prove a sua resposta.

(b) Para $m \geq n$, mostre que se $A^T\boldsymbol{b} = \boldsymbol{0}$, então $\boldsymbol{b}$ é ortogonal ao subespaço gerado pelas colunas de A, span$(\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\})$, onde $\boldsymbol{a}_i$ são as colunas de A.

(c) Existem muitas identidades envolvendo a pseudoinversa de A, $A^\dagger = \left(A^TA\right)^{-1}A^T$. Mostre **três** (apenas) das identidades seguintes:

i. $A^\dagger AA^\dagger = A^\dagger$
ii. $(A^\dagger A)^T = I$
iii. $AA^\dagger A = A$
iv. $\left(AA^\dagger\right)^T = AA^\dagger$
v. $A(I - A^\dagger A) = O$
vi. $(I - AA^\dagger)A = O$
vii. Se A tem colunas ortonormadas, $A^\dagger = A^T$.

2. Use o método dos mínimos quadrados para obter a reta e a parábola que ajustam os dados seguintes:

x	0	2	3	5	8	11	12	15
y	50	56	60	72	85	100	110	125

Represente num mesmo gráfico os dados e as curvas obtidas.

3. Para $\boldsymbol{x} \in \mathbb{R}^n$, o seguinte algoritmo calcula o vetor de Householder $\boldsymbol{v} = (v_1, \ldots, v_n)^T \in \mathbb{R}^n$, com $\|v\| = 1$, de forma que a matriz ortogonal $H = I_n - 2\boldsymbol{v}\boldsymbol{v}^T$ é tal que $H\boldsymbol{x} = \|x\|\boldsymbol{e}_1$, onde $\boldsymbol{e}_1$ representa a primeira coluna da matriz identidade I_n.

Input: vetor (coluna) $\boldsymbol{x} \in \mathbb{R}^n$
Output: vetor de Householder $\boldsymbol{v} \in \mathbb{R}^n, \beta \in \mathbb{R}$

$n = \texttt{length}(\boldsymbol{x})$
$\sigma = \boldsymbol{x}_{2:n}^T\boldsymbol{x}_{2:n}$
$\boldsymbol{v} = \boldsymbol{x}$
if $\sigma \neq 0$ **then**
  $\mu = \sqrt{x_1^2 + \sigma}$
  if $x_1 \leq 0$ **then**
    $v_1 = x_1 - \mu$
  else
    $v_1 = -\sigma/\left(x_1 + \mu\right)$
  end if
end if
$\boldsymbol{v} = \boldsymbol{v}/\sqrt{\boldsymbol{v}^T\boldsymbol{v}}$

(a) Verifique que o algoritmo calcula sempre o vetor de Householder $\boldsymbol{v} = \boldsymbol{x} - \|x\|_2\boldsymbol{e}_1$ (caso $\boldsymbol{x}$ não seja múltiplo de $\boldsymbol{e}_1$) e comente sobre a forma como se evita o cancelamento subtrativo (subtração de números muito próximos).

(b) Escreva em MATLAB a função `v=housevector(x)` que implemente o algoritmo apresentado e mostre os resultados para diferentes vetores $\boldsymbol{x}$.

(continua)

(c) Altere a função `[Q,R]=qrhouseholder(A)`, desenvolvida nas aulas, por forma a incorporar a chamada à função da alínea anterior.

Considere o sistema $A\boldsymbol{x} = \boldsymbol{b}$, onde A é a matriz de Hilbert de ordem $n = 10$ e $\boldsymbol{b}$ é o vetor com todos os elementos iguais a 1. Use a nova função `qrhouseholder` para obter a decomposição QR da matriz A e obtenha, a partir desta decomposição, a solução do sistema. Compare o resíduo relativo (em norma) da solução assim obtida com o que se obtém usando o operador *backslash* do MATLAB.

4. O seguinte algoritmo permite obter a decomposição LU (sem pivotação) de uma matriz $A \in \mathbb{R}^{n\times n}$, onde U é uma matriz triangular superior e L é uma matriz triangular inferior com 1's na diagonal.

Input: matriz $A \in \mathbb{R}^{n\times n}$
Output: matriz triangular inferior L, matriz triangular superior U

$n = \texttt{size}(A, 1)$
$U = A$; $L = \texttt{eye}(n)$
for $j = 1$ to n **do**
  for $i = j + 1$ to n **do**
    $l_{ij} = u_{ij}/u_{jj}$
    $u_{i,j:n} = u_{i,j:n} - l_{ij} * u_{j,j:n}$
  end for
end for

(a) Escreva uma função em `Matlab` `[L,U]=lunp(A)` que implemente o algoritmo apresentado. Aplique a função à matriz

$$A = \begin{bmatrix} 1 & 1 & 1 & 4 \\ 1 & 2 & 4 & 5 \\ 1 & 3 & 9 & 2 \\ 1 & 5 & 1 & 2 \end{bmatrix}.$$

(b) Escreva uma função em `Matlab` que determine a **decomposição UL** de uma dada matriz $A \in \mathbb{R}^{m\times n}$, onde U é uma matriz triangular superior com 1's na diagonal e L é uma matriz triangular inferior. Aplique a função à matriz da alínea anterior.

Sugestão: podemos pensar em desenvolver um algoritmo que usa operações elementares sobre as linhas de A *de baixo para cima*, à semelhança do algoritmo anterior que usa transformações elementares *de cima para baixo*; obtendo-se assim uma decomposição LU, mas sendo L uma matriz triangular superior com 1's na diagonal (novo U) e U uma matriz triangular inferior (novo L).

DATA LIMITE PARA O ENVIO DA RESOLUÇÃO: 6 DE MAIO DE 2024.