Laborator 4

Triangularizare ortogonală cu reflectori. Factorizarea QR. Problema celor mai mici pătrate

4.1 Triangularizare ortogonală cu reflectori. Factorizarea QR

4.1.1 Transformări Householder

Definiţia 4.1 Un *reflector elementar* (sau transformare Householder) de ordin m este o matrice $U \in \mathbf{R}^{m\times m}$, de forma

$$U = I_m - \frac{1}{\beta}uu^T \tag{4.1}$$

unde $u \in \mathbf{R}^m$ este un vector nenul şi

$$\beta = \|u\|^2/2.$$

Când $u_i = 0$ pentru $i = 1 : k-1$ reflectorul elementar este numit *de indice* k.

Un reflector elementar U_k de ordinul m şi indice k are structura:

$$U_k = \begin{bmatrix} I_{k-1} & 0 \\ 0 & \bar{U} \end{bmatrix},$$

unde $\bar{U} \in \mathbf{R}^{(m-k+1)\times(m-k+1)}$ este un reflector elementar de ordin $m-k+1$.

Principalele proprietăţi ale transformărilor Housseholder sunt date în următoarea teoremă.

Teorema 4.2 *a) Dacă $U \in \mathbf{R}^{m\times m}$ este un reflector elementar, atunci U este simetrică şi ortogonală.*

b) Fie $k \in 1 : m-1$ şi un vector $x \in \mathbf{R}^m$ astfel încât

$$\sigma^2 = \sum_{i=k}^{m} x_i^2 \neq 0. \tag{4.2}$$

Atunci vectorul $u_k \in \mathbf{R}^m$ dat de

$$u_{ik} \equiv (u_k)_i = \begin{cases} 0, & i = 1 : k-1 \\ x_k + \sigma, & i = k \\ x_i, & i = k+1 : n \end{cases} \tag{4.3}$$

defineşte un reflector elementar de ordin m şi indice k

$$U_k = I_m - \frac{1}{\beta_k} u_k u_k^T, \quad \beta_k = \|u_k\|^2/2, \tag{4.4}$$

astfel încât

$$(U_k x)_i = \begin{cases} x_i, & i = 1 : k-1 \\ -\sigma, & i = k \\ 0, & i = k+1 : n \end{cases}$$

Proprietatea esenţială a reflectorilor elementari este aceea că ei pot fi utilizaţi pentru a introduce zerouri într-un vector.

4.1.2 Triangularizarea ortogonală. Factorizarea QR

Rezolvarea problemei CMMP într-o manieră stabilă din punct de vedere numeric necesită o procedură pentru reducerea matricelor la forma triunghiulară prin transformări ortogonale.

Teorema 4.3 *Fie $A \in \mathbf{R}^{m \times n}$; atunci există o matrice ortogonală $U \in \mathbf{R}^{m \times m}$ astfel încât*

$$UA = R \tag{4.5}$$

este superior triunghiulară, unde U este un produs de reflectori Householder

$$U = U_p U_{p-1} \dots U_2 U_1, \tag{4.6}$$

cu $p = \min m-1, n$.

O aplicaţie imediată a teoremei de mai sus este calculul aşa numitei *factorizări QR* a matricei A. Definiţia şi existenţa factorizării QR este dată de următoarea

Teorema 4.4 *Dacă $A \in \mathbf{R}^{m \times n}$, are coloanele liniar independente, atunci ea poate fi scrisă unic în forma*

$$A = QR, \tag{4.7}$$

unde $Q \in \mathbf{R}^{m \times n}$ are coloanele ortogonale şi $R \in \mathbf{R}^{n \times n}$ este superior triunghiulară cu elementele diagonale pozitive. Factorizarea matriceală (4.7) este numită factorizare QR.

4.1.3 Algoritmi

ALGORITM 4.5 (Calculul unui reflector) (Date întregul $k \in 1 : m-1$ şi un vector x m-dimensional, algoritmul returnează σ, β_k şi u_k astfel încât $[(I - (1/\beta_k)u_k u_k^T)x]_i = 0$, pentru $i = k+1 : m$. Componentele lui u_k suprascriu cele ale lui x.)

1. $\sigma \leftarrow \mathrm{sign}(x_k) \cdot \sqrt{\sum_{i=k}^{m} x_i^2}$
2. $x_k \leftarrow u_{kk} = x_k + \sigma$

(2'. $x_i \leftarrow u_{ik} = x_i$, for $i = k+1 : m$)

3. $\beta_k \leftarrow \sigma u_{kk}$

Pentru triangularizarea ortogonală a matricei $A \in \mathbf{R}^{m\times n}$ putem folosi următoarea schemă:

1. **for** $k = 1 : \min(m-1, n)$
   1. Se determină U_k astfel încât $(U_k a_k)_i = 0$, pentru $i = k+1 : m$
   2. $A \leftarrow U_k A$

Schema de calcul de mai sus conduce la următorul algoritm.

ALGORITM 4.6 *(TORT – triangularizare ortogonală cu reflectori)* (Dată o matrice $A \in \mathbf{R}^{m\times n}$, algoritmul suprascrie matricea A cu matricea superior triunghiulară R şi calculează vectorii $u_k = U(:, k)$ şi scalarii $\beta_k = b(k)$ care definesc reflectorii elementari U_k, $k \in 1 : p$, $p = \min(m-1, n)$, astfel încât $R = U_n U_{n-1} \dots U_2 U_1 A$.)

1. $p = \min(m-1, n)$
2. **for** $k = 1 : p$
   1. $\sigma \leftarrow \mathrm{sign}(a_{kk}) \cdot \sqrt{\sum_{i=k}^{m} a_{ik}^2}$
   2. **dacă** $\sigma = 0$ **atunci** $b(k) \leftarrow \beta_k = 0$

      altfel
      1. $U(k, k) = a_{kk} + \sigma$
      2. **for** $i = k+1 : m$
         1. $U(i, k) = a_{ik}$
      3. $b(k) \leftarrow \beta_k = \sigma u_{kk}$
      4. $a_{kk} \leftarrow r_{kk} = -\sigma$
      5. **for** $i = k+1 : m$
         1. $a_{ik} = 0$
      6. **for** $j = k+1 : n$
         1. $\tau \leftarrow \left(\sum_{i=k}^{m} u_{ik} a_{ij}\right) / \beta_k$
         2. **for** $i = k : m$
            1. $a_{ij} \leftarrow a_{ij} - \tau u_{ik}$

Algoritmul poate fi apelat cu sintaxa $[R, U, b] = TORT(A)$. De notat faptul că algoritmul nu calculează explicit matricea ortogonală U astfel încât UA este superior triunghiulară, ci numai memorează toate elementele necesare pentru a o face.

Pentru factorizarea QR $A = QR$ a unei matrice A fie $UA = R$ triunghiularizarea ortogonală de mai sus a lui A. Atunci $A = U^T R$, deci matricea Q poate fi calculată din:

$$Q = U^T = (U_p \dots U_2 U_1)^T = U_1 U_2 \dots U_p$$

folosind următoarea schemă de calcul:

1. $Q = I_m$
2. **for** $k = p : -1 : 1$
   1. $Q \leftarrow U_k Q$.

Obţinem următorul algoritm pentru factorizarea QR.

ALGORITM 4.7 *(QR - Factorizarea QR)* (Dată o matrice $A \in \mathbf{R}^{m\times n}$, algoritmul calculează matricea ortogonală $Q \in \mathbf{R}^{m\times m}$ şi matricea superior triunghiulară $R \in \mathbf{R}^{m\times n}$ astfel încât $A = QR$.)

1. $[R, U, b] = TORT(A)$
2. $p = \min(m-1, n)$
3. $Q = I_m$
4. **for** $k = p : -1 : 1$
   1. **for** $j = k : m$
      1. $\tau \leftarrow \left(\sum_{i=k}^{m} u_{ik} q_{ij}\right) / \beta_k$
      2. **for** $i = k : m$
         1. $q_{ij} \leftarrow q_{ij} - \tau u_{ik}$

În cazul $m > n$, pentru a obţine factorizarea QR $A = Q'R'$, i.e. a matricei $Q \in \mathbf{R}^{m\times n}$ cu coloanele ortogonale şi a matricei $R \in \mathbf{R}^{n\times n}$ pătrată superior triunghiulară, este suficient să luăm $Q' = Q(:, 1 : n)$ şi $R' = R(1 : n, :)$ unde Q şi R sunt date de algoritmul de mai sus.

4.2 Problema celor mai mici pătrate

Considerăm sistemul liniar

$$Ax = b,$$

unde $A \in \mathbf{R}^{m\times n}$, $b \in \mathbf{R}^m$.

În general, un sistem *supradeterminat* ($m > n$), i.e. cu mai ecuaţii decât necunoscute, nu are soluţii şi o formulare naturală a rezolvării sistemului este problema găsirii unui vector $x \in \mathbf{R}^n$ astfel încât vectorul $y = Ax$ să fie cât mai apropiat posibil de vectorul b. Mai precis, dorim să determinăm un vector $x \in \mathbf{R}^n$ care minimizează funcţia:

$$\rho(x) = \nu(b - Ax),$$

unde ν este o normă vectorială pe $\mathbf{R}^m$. Când ν este norma euclidiană, problema minimizării normei reziduului $r = b - Ax$ se numeşte *problema celor mai mici pătrate* (CMMP).

Problema CMMP este întâlnită în multe domenii de aplicaţii, cum sunt teoria aproximării funcţiilor şi datelor, statistică, etc.

Un sistem *subdeterminat* ($m < n$) are, generic, o infinitate de soluţii. Un criteriu natural de selecţie a unei anumite soluţii este de a cere ca ea să fie de "lungime" minimă, i.e. de a minimiza funcţia:

$$\tau(x) = \mu(x)|_{Ax=b},$$

unde μ este o normă vectorială pe $\mathbf{R}^n$. Când μ este norma euclidiană, soluţia sistemului subdeterminat $Ax = b$ de normă euclidiană minimă se numeşte *soluţie normală.*

4.2.1 Rezolvarea problemei celor mai mici pătrate

Fie problema CMMP a minimizării normei euclidiane a reziduului sistemului liniar supradeterminat:

$$\min_{x \in \mathbf{R}^n} \|r\| = \min_{x \in \mathbf{R}^n} \|b - Ax\|, \tag{4.8}$$

unde $A \in \mathbf{R}^{m \times n}$, $m > n$, şi $b \in \mathbf{R}^m$ sunt date.

Condiţiile în care problema CMMP are o soluţie şi când această soluţie este unică sunt date de următoarea teoremă.

Teorema 4.8 *Fie date $A \in \mathbf{R}^{m \times n}$ şi $b \in \mathbf{R}^m$. Problema celor mai mici pătrate (4.8) admite întotdeauna o soluţie. Soluţia este unică dacă şi numai dacă matricea A are coloanele liniar independente. În acest caz soluţia CMMP a sistemului supradeterminat $Ax = b$ poate fi scrisă în forma:*

$$x^* = A^+ b, \tag{4.9}$$

unde matricea

$$A^+ = (A^T A)^{-1} A^T \tag{4.10}$$

este numită pseudo-inversa *sau* inversa generalizată Moore-Penrose *a lui A.*

Un mijloc de calcul numeric stabil a soluţiei CMMP constă în triangularizarea ortogonală a matricei sistemului (algoritmul $TORT$ din laboratorul precedent). Într-adevăr, fie matricea ortogonală U astfel încât $UA = R$, unde R este o matrice superior triunghiulară. Atunci matricea R poate fi scrisă în forma

$$R = \begin{bmatrix} R' \\ 0 \end{bmatrix},$$

cu R' o matrice pătrată $n \times n$ superior triunghiulară. Matricea R are coloanele liniar independente întrucât A are această proprietate şi, deci, R' este o matricea nesingulară. Transformările ortogonale conservând norma euclidiană, avem:

$$\|b - Ax\| = \|U(b - Ax)\| = \|d - Rx\|, \tag{4.11}$$

unde $d = Ub$. Acum, folosind partiţia

$$d = \begin{bmatrix} d' \\ d'' \end{bmatrix}, \tag{4.12}$$

unde $d' \in \mathbf{R}^n$ şi $d'' \in \mathbf{R}^{m-n}$, relaţia (4.11) poate fi scrisă:

$$\|b - Ax\| = \left\| \begin{bmatrix} d' - R'x \\ d'' \end{bmatrix} \right\| = \sqrt{\|d' - R'x\|^2 + \|d''\|^2},$$

care este minimă când $d' - R'x = 0$. Deci, soluţia CMMP a sistemului supradeterminat $Ax = b$ poate fi calculată ca soluţie a sistemului superior triunghiular nesingular:

$$R'x = d'.$$

Reziduul minimal $r^* = b - Ax^* = b_2$ are semnificaţia de *proiecţie ortogonală* a vectorului b pe subspaţiul liniar $\mathcal{T} = \mathcal{S}^{\perp} = (\mathrm{Im}A)^{\perp} = \mathrm{Ker}A^T$, în timp ce vectorul $b_1 = Ax^*$ are semnificaţia de proiecţie a vectorului b pe subspaţiul liniar $\mathcal{S} = \mathrm{Im}A$. Cele două proiecţii ortogonale ale vectorului b pe $\mathrm{Im}A$ şi $\mathrm{Ker}A^T$ se calculează cel mai bine folosind expresiile:

$$b_1 = U_1 U_2 \dots U_n \begin{bmatrix} d' \\ 0 \end{bmatrix} \qquad b_2 = U_1 U_2 \dots U_n \begin{bmatrix} 0 \\ d'' \end{bmatrix}.$$

4.2.2 Rezolvarea sistemelor liniare subderdeterminate

Condiţiile pentru existenţa şi unicitatea soluţiei normale sunt date de:

Teorema 4.9 *Dacă matricea $A \in \mathbf{R}^{m \times n}$, $m < n$, are liniile liniar independente, atunci sistemul $Ax = b$ are o soluţie normală unică, i.e. un vector unic $x^* \in \mathbf{R}^n$ astfel încât:*

$$\|x^*\| = \min_{Ax=b} \|x\|.$$

Soluţia normală poate fi scrisă în forma:

$$x^* = A^{\#} b, \tag{4.13}$$

unde

$$A^{\#} = A^T (AA^T)^{-1} \tag{4.14}$$

se numeşte pseudo-inversa normală *a lui A.*

Pentru un calcul fiabil al soluţiei normale se poate proceda după cum urmează. Fie $UA^T = R$, unde $U = U_m \cdots U_2 U_1$ şi $R = \begin{bmatrix} R' \\ 0 \end{bmatrix}$ cu R' superior triunghiulară nesingulară, triangularizarea ortogonală a matricei A^T. Atunci sistemul $Ax = b$ se poate scrie $AU^T Ux = b$ sau, notând $y = Ux$ şi folosind partiţia $y = \begin{bmatrix} y' \\ y'' \end{bmatrix}$, $\begin{bmatrix} R'^T & 0 \end{bmatrix} \begin{bmatrix} y' \\ y'' \end{bmatrix} = R'^T y' = b$. Prin urmare, mulţimea soluţiilor sistemului subdeterminat $Ax = b$ se scrie sub forma

$$x = U^T y = U_1 U_2 \ldots U_m \begin{bmatrix} y' \\ y'' \end{bmatrix}$$

unde y' este soluţia sistemului inferior triunghiular $R'^T y' = b$ iar $y'' \in \mathbf{R}^{n-m}$ este arbitrar. Evident soluţia normală se obţine pentru $y'' = 0$.

4.2.3 Algoritmi

Algoritmul dat mai jos este un mijloc foarte stabil de rezolvare a problemei CMMP şi, dacă este necesar, de calcul al proiecţiilor ortogonale ale unui vector dat pe cele două subspaţii liniare complementare definite de o matrice dată.

Algoritm 4.10 *(CMMP – problema celor mai mici pătrate)* (Date o matrice $A \in \mathbf{R}^{m\times n}$, $m > n$, cu coloanele liniar independente, şi un vector $b \in \mathbf{R}^n$, acest algoritm calculează soluţia CMMP a sistemului liniar supradeterminat $Ax = b$ şi proiecţiile ortogonale b_1 şi b_2 ale vectorului b pe subspaţiile liniare ImA şi respectiv KerA^T.)

{ Triangularizarea ortogonală a lui A }
  1. $[R, U, beta] = TORT(A)$
{ Calculul vectorului d care suprascrie b }
  2. **for** $k = 1 : n$
    1. $\tau \leftarrow \left(\sum_{i=k}^m u_{ik} b_i\right) / \beta_k$
    2. **for** $i = k : m$
      1. $b_i \leftarrow b_i - \tau u_{ik}$
{ Calculul soluţiei CMMP }
  3. $x = UTRIS(R(1:n,:), b(1:n))$
{ Calculul proiecţiilor }
  4. $b_1(1:n) \leftarrow b(1:n)$; $b_1(n+1:m) \leftarrow 0$
  5. $b_2(1:n) \leftarrow 0$; $b_2(n+1:m) \leftarrow b(n+1:m)$
  6. **for** $k = n : -1 : 1$
    1. $\tau \leftarrow \left(\sum_{i=k}^m u_{ik} b_{1i}\right) / \beta_k$
    2. **for** $i = k : m$
      1. $b_{1i} \leftarrow b_{1i} - \tau u_{ik}$
    3. $\tau \leftarrow \left(\sum_{i=k}^m u_{ik} b_{2i}\right) / \beta_k$

4. **for** $i = k : m$
   1. $b_{2i} \leftarrow b_{2i} - \tau u_{ik}$

Algoritmul prezentat mai jos este cel mai bun algoritm, din punct de vedere numeric, pentru calculul soluţiei normale a unui sistem subdeterminat cu matricea sistemului de rang maximal.

ALGORITM 4.11 *(SN – rezolvarea sistemelor subdeterminate)* (Dată o matrice $A \in \mathbf{R}^{m \times n}$, $m < n$, cu liniile liniar independente, şi un vector $b \in \mathbf{R}^n$, algoritmul calculează soluţia normală $x \in \mathbf{R}^n$ sistemului liniar subdeterminat $Ax = b$.)

{ Triangularizarea ortogonală a lui A^T }
1. $[R, U, beta] = TORT(A^T)$

{ Rezolvarea sistemului inferior triunghiular $R^T y' = b$, cu y' suprascriind b }
2. $b = LTRIS(R^T, b)$

{ Calculul soluţiei normale }
3. $x(1 : m) \leftarrow b; \quad x(m + 1 : n) \leftarrow 0$
4. **for** $k = m : -1 : 1$
   1. $\tau \leftarrow \left(\sum_{i=k}^{n} u_{ik} x_i\right) / \beta_k$
   2. **for** $i = k : n$
      1. $x_i \leftarrow x_i - \tau u_{ik}$

4.3 Sarcini de lucru

4.3.1 A. In laborator

1. Se elaborează şi editează programele MATLAB pentru implementarea algoritmilor de triangularizare ortogonală şi de factorizare QR. Se testează corectitudinea algoritmilor şi programelor pentru matrice cu elemente aleatoare de dimensiunii 7×5 şi 20×7.

2. Se elaborează şi editează programele MATLAB pentru implementarea algoritmilor de rezolvare în sens CMMP a sistemelor supradeterminate şi subdeterminate.

3. Se testează corectitudinea algoritmilor şi programelor pentru sisteme supradeterminate de 20 de ecuaţii şi 6 necunoscute şi sisteme subdeterminate cu 6 ecuaţii şi 20 de necunoscute.

4. Se testează ortogonalitatea proiecţiilor vectorului b şi faptul că suma proiecţiilor dau vectorul b.

Observaţie. Dacă obiectivele de mai sus nu se ating în laborator atunci se continuă acasă.

4.3.2 B. Acasă

1 Se scriu şi testează programe MATLAB pentru [1]:

a) calculul elementelor definitorii ale unui reflector U_k care anulează componentele $k+1:n$ ale unui vector $x \in \mathbf{R}^n$ dat;

b) înmulţirea unei matrice A pe stânga cu un reflector U_k: $A \leftarrow U_k A$;

c) înmulţirea unei matrice A pe dreapta cu un reflector U_k: $A \leftarrow A U_k$.

2 Se rescriu algoritmii de triangularizare ortogonală şi factorizare QR a unei matrice cu folosirea procedurilor de mai sus.

3 Fie $H \in \mathbf{R}^{m \times n}$ $(m > n)$ o matrice superior Hessenberg ($h_{ij} = 0$ for $i > j+1$). Scrieţi algoritmi eficienţi şi programe pentru:

a. triangularizarea ortogonală a matricei H.

b. factorizarea QR a matricei H.

4 Fie $A \in \mathbf{R}^{m \times n}$, $m > n$, o matrice monică a cărei factorizare QR este cunoscută ($A = QR$, $Q \in \mathbf{R}^{m \times m}$, $R \in \mathbf{R}^{m \times n}$). Fie $\tilde{A} = [A \ z]$, unde $z \in \mathbf{R}^m$. Elaboraţi un algoritm şi un program pentru calculul factorizării QR a lui $\tilde{A}$.

5 Considerăm dat vectorul $x \in \mathbf{R}^n$, de normă unitară. Scrieţi un program de calcul al unei matrice cu coloanele ortogonale $Y \in \mathbf{R}^{n \times (n-1)}$ astfel încât $x^T Y = 0$ (i.e. matricea $[x \ Y]$ să fie ortogonală).

6 Fie $A \in \mathbf{R}^{m \times n}$ o matrice de rang maximal. Elaboraţi un algoritm şi un program de calcul al pseudo-inversei lui A pentru ambele situaţii ce pot apare: $m \geq n$ sau $m \leq n$.

[1] Aceste proceduri vor fi utile la laboratorul dedicat calculului valorilor proprii.