

Tema nr. 5

Fie $p \in \mathbb{N}^*$ și $n \in \mathbb{N}^*$ dimensiunile matricei A , $p \geq n$, ϵ - precizia calculelor, matricea $A \in \mathbb{R}^{p \times n}$.

- Pentru $p = n$, să se aproximeze valorile și vectorii proprii ale matricei simetrice A ($A = A^T$) folosind metoda Jacobi.

Să se verifice că:

$$A^{init} * U \approx U * \Lambda \quad , \quad U = [u^1 \ u^2 \ \dots \ u^n] \quad , \quad \Lambda = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_n]$$

unde λ_i sunt valorile proprii aproximative, u^i sunt vectorii proprii corespunzători iar A^{init} este o copie a matricei inițiale.

Verificarea $A^{init} * U \approx U * \Lambda$ se va face afișând norma matriceală:

$$||A^{init} * U - U * \Lambda||.$$

- Să se calculeze următorul șir de matrice, $A^{(k)}$:

$A^{(0)} = A = L^0 * (L^0)^T$, se calculează factorizarea Cholesky a matricei $A^{(0)}$

$A^{(1)} = (L^0)^T * L^0$, ($A^{(1)}$ se obține înmulțind matricele $(L^0)^T$ cu L^0)

$A^{(1)} = L^1 * (L^1)^T$, se calculează factorizarea Cholesky a matricei $A^{(1)}$

$A^{(2)} = (L^1)^T * L^1$, ($A^{(2)}$ se obține înmulțind matricele $(L^1)^T$ cu L^1)...

$A^{(k)} = L^k * (L^k)^T$ (factorizare Cholesky a matricei $A^{(k)}$),

$$A^{(k+1)} = (L^k)^T * L^k$$

Calculul se oprește când diferența între două matrice consecutive este suficient de mică ($||A^{(k)} - A^{(k-1)}|| < \epsilon$) sau când s-a depășit un număr maxim de iterații prestabilit ($k > k_{max}$). Nu e nevoie să memorăm toate matricele din șir. Afișăm ultima matrice calculată. Ce formă are? Ce informații se găsesc în această matrice? Pentru calculul descompunerii Cholesky folosiți funcția din biblioteca numerică pe care o folosiți. Dacă nu găsiți o funcție care să calculeze factorizare Choleski, folosiți o funcție care calculează o descompunere LU pentru matrice.

- Pentru $p > n$, utilizând descompunerea după valori singulare (**Singular Value Decomposition**) din biblioteca folosită la [Tema 2](#), să se calculeze și să se afișeze:

- valorile singulare ale matricei A ,
- rangul matricei A ,
- numărul de condiționare al matricei A ,
- pseudoinversa Moore-Penrose a matricei A , $A^I \in \mathbb{R}^{n \times p}$,

$$A^I = V * S^I * U^T$$

- calculați matricea pseudo-inversă în sensul celor mai mici pătrate:

$$A^J = (A^T * A)^{-1} * A^T$$

și afișați norma:

$$\|A^I - A^J\|_1$$

Pentru rangul și numărul de condiționare al matricei să se folosească relațiile descrise în acest fișier și de asemenea funcțiile din bibliotecă, funcții care calculează aceste valori.

Bonus 15 pt. : Pentru memorarea matricei A se va folosi un vector v de dimensiune $\frac{n(n+1)}{2}$ (se memorează doar partea inferior triunghiulară a matricei A , celelalte elemente găsindu-se din relația de simetrie). Algoritmul lui Jacobi se va scrie adaptat pentru acest tip de memorare a matricei A .

Vectori și valori proprii - definiții

Fie $A \in \mathbb{R}^{n \times n}$ o matrice reală de dimensiune n . Se numește *valoare proprie* asociată matricei A , numărul complex $\lambda \in \mathbb{C}$, pentru care există un vector nenul $u \neq 0$ numit și *vector propriu* asociat valorii proprii λ pentru care:

$$Au = \lambda u$$

Valorile proprii ale matricei A pot fi definite și ca rădăcini ale polinomului caracteristic asociat matricei A , $p_A(\lambda)$:

$$p_A(\lambda) = \det(\lambda I - A) = 0$$

Polinomul caracteristic este un polinom de grad n , deci orice matrice de dimensiune n are n valori proprii (reale și/sau complex conjugate).

Pentru matricele simetrice se poate arăta că au toate valorile proprii reale.

Metoda lui Jacobi pentru aproximarea valorilor proprii ale matricelor simetrice

Pentru a aproxima valorile proprii ale unei matrice se folosește relația de asemănare. Două matrici A și B se numesc *asemenea* ($A \sim B$) dacă există o matrice nesingulară P astfel încât $A = P * B * P^{-1}$ ($\longleftrightarrow B = P^{-1} * A * P$). Se observă că dacă $A \sim B$ atunci și $B \sim A$. Relația de asemănare se folosește în algoritmi de aproximare a valorilor proprii deoarece matricele asemenea au același polinom caracteristic ($p_A(\lambda) \equiv p_B(\lambda)$) și în consecință au aceleași valori proprii.

Fie $A \in \mathbb{R}^{n \times n}$ o matrice simetrică ($A = A^T$). Se știe că matricele simetrice au toate valorile proprii reale.

Ideea algoritmului lui Jacobi este de a construi un șir de matrice simetrice, asemenea cu matricea inițială, șir care converge la o matrice diagonală. Matricea diagonală limită va fi asemenea cu matricea inițială A și prin urmare pe diagonala acestei matrice limită vom găsi valorile proprii căutate.

Construcția șirului de matrice

O matrice de rotație $R_{pq}(\theta) = R_{pq} = (r_{ij})_{i,j=1,n}$ are următoarea formă :

$$R_{pq}(\theta) = R_{pq} = \begin{pmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & & & & & & & \\ 0 & 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & & & & & & & \\ 0 & 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ \vdots & & & & & & & \\ 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix}$$

$$r_{ij} = \begin{cases} 1 & \text{pentru } i = j, \quad i \neq p \text{ și } i \neq q \\ c & \text{pentru } i = j, \quad i = p \text{ sau } i = q \\ s & \text{pentru } i = p, \quad j = q \\ -s & \text{pentru } i = q, \quad j = p \\ 0 & \text{pentru restul indicilor } i, j \end{cases}$$

unde $p, q \in \{1, \dots, n\}$ sunt indici iar c și s sunt două numere reale care satisfac relația $c^2 + s^2 = 1$ (c și s pot fi alese astfel încât $c = \cos \theta, s = \sin \theta$).

Șirul de matrice $\{A^{(k)}\} \subseteq \mathbb{R}^{n \times n}$ se construiește astfel:

$$A^{(0)} = A \quad , \quad A^{(k+1)} = R_{pq}(\theta) * A^{(k)} * R_{pq}^T(\theta)$$

unde $R_{pq}(\theta)$ sunt matrice de rotație.

- **indicii** (p, q) sunt aleși ca fiind indicii celui mai mare element nediatonal din matrice luat în valoare absolută:

$$\begin{aligned} |a_{pq}^{(k)}| &= \max\{|a_{ij}^{(k)}|; i = 1, \dots, n, j = 1, \dots, n, i \neq j\} = (A = A^T) = \\ &= \max\{|a_{ij}^{(k)}|; i = 2, \dots, n, j = 1, \dots, i-1\} \end{aligned} \tag{1}$$

(datorită simetriei matricelor din șir, se poate căuta elementul $a_{pq}^{(k)}$ de mai sus doar în partea strict inferior triunghiulară a matricei $A^{(k)}$)

- **unghiul** θ ($c = \cos \theta, s = \sin \theta, t = \tan \theta$) este ales astfel ca elementele (p, q) și (q, p) ale matricei $A^{(k+1)}$ să fie zero, i.e.,

$$a_{pq}^{(k+1)} = a_{qp}^{(k+1)} = 0.$$

Schema algoritmului

```

 $k = 0$ ;  $U = I_n$ ;
calculează indicii  $p$  și  $q$  (vezi (1)) ;
calculează unghiul  $\theta$ , adică  $c$ ,  $s$  și  $t$ ;
while ( $A \neq$  matrice diagonală și  $k \leq k_{max}$ )
{
   $A = R_{pq}(\theta) * A * R_{pq}^T(\theta)$  ;
  ( a se vedea formulele (5) de mai jos )
   $U = U * R_{pq}^T(\theta)$  ;
  ( a se vedea formulele (7) de mai jos )
  calculează indicii  $p$  și  $q$  (vezi (1));
  calculează unghiul  $\theta$ , adică  $c$ ,  $s$  și  $t$  ;
  ( a se vedea formulele (3) și (4) de mai jos )
   $k = k + 1$ ;
}

```

La finalul acestui algoritm vom avea în matricea $A = A^{final}$ o matrice (aproximativ) diagonală, valorile de pe diagonală fiind aproximări ale valorilor proprii, iar coloanele matricei U (matrice ortogonală) sunt aproximări ale vectorilor proprii corespunzători.

$$A^{final} = U^T * A^{init} * U$$

Pasul k al algoritmului

La acest pas se construiește matricea B pornind de la matricea A astfel:

$$B = R_{pq}(\theta) * A * R_{pq}^T(\theta)$$

și matricea V pornind de la matricea U :

$$V = U * R_{pq}^T(\theta).$$

Trecerea de la matricea A la matricea B se face după următoarele formule:

$$\left\{ \begin{array}{l} b_{pj} = b_{jp} = c a_{pj} + s a_{qj} , \quad j = 1, 2, \dots, n, \quad j \neq p, \quad j \neq q \\ b_{qj} = b_{jq} = -s a_{pj} + c a_{qj} , \quad j = 1, 2, \dots, n, \quad j \neq p, \quad j \neq q \\ b_{pp} = c^2 a_{pp} + s^2 a_{qq} + 2 c s a_{pq} \\ b_{qq} = s^2 a_{pp} + c^2 a_{qq} - 2 c s a_{pq} \\ b_{pq} = b_{qp} = (c^2 - s^2) a_{pq} + c s (a_{qq} - a_{pp}) \\ b_{ij} = a_{ij} \quad \text{în rest} \end{array} \right. \quad (2)$$

Pentru a deduce unghiul θ se impune condiția $b_{pq} = b_{qp} = 0$, adică :

$$(c^2 - s^2)a_{pq} + c s (a_{qq} - a_{pp}) = 0$$

de unde rezultă :

$$\alpha = \cotg(2\theta) = \frac{(a_{pp} - a_{qq})}{2a_{pq}}$$

Dacă notăm cu $t = \tg\theta$ avem:

$$\cotg(2\theta) = \frac{(1 - t^2)}{2t}$$

rezultă că t satisface ecuația:

$$t^2 + 2\alpha t - 1 = 0$$

deci

$$t = -\alpha + (\alpha^2 + 1)^{1/2} \text{ sau } t = -\alpha - (\alpha^2 + 1)^{1/2}.$$

Dintre cele două valori de mai sus ale lui t se alege rădăcina de modul minim ($\theta \in [0, \pi/4]$):

$$t = -\alpha + \text{semn}(\alpha)\sqrt{\alpha^2 + 1} = \begin{cases} -\alpha + \sqrt{\alpha^2 + 1} & \text{dacă } \alpha \geq 0 \\ -\alpha - \sqrt{\alpha^2 + 1} & \text{dacă } \alpha < 0 \end{cases} \quad (3)$$

$$\text{semn}(\alpha) = \begin{cases} 1 & \text{dacă } \alpha \geq 0 \\ -1 & \text{dacă } \alpha < 0 \end{cases}$$

Avem:

$$c = \frac{1}{\sqrt{1+t^2}} \quad , \quad s = \frac{t}{\sqrt{1+t^2}} \quad (4)$$

Cazul $a_{pq} = 0$

Ținând cont că a_{pq} este cel mai mare element nediagonal în valoare absolută, cazul $a_{pq} = 0$ înseamnă că matricea A la care s-a ajuns, este matrice diagonală, algoritmul oprindu-se în această situație - pe diagonala matricei A se găsesc aproximările valorilor proprii căutate. Prin urmare testul:

$$A \neq \text{matrice diagonală}$$

din schema algoritmului de mai sus se poate înlocui cu testul:

$$|a_{pq}| > \epsilon$$

unde ϵ este precizia calculelor.

Se observă că:

$$\begin{aligned} b_{pp} - a_{pp} &= s^2(a_{qq} - a_{pp}) + 2c s a_{pq} = 2s (c - \alpha s)a_{pq} = \\ &= 2s [c - (c^2 - s^2)s/(2c s)]a_{pq} = t a_{pq} \end{aligned}$$

La fel se deduce că :

$$b_{qq} - a_{qq} = -t a_{pq}$$

La pasul k operația $A = R_{pq}(\theta) * A * R_{pq}^T(\theta)$ se poate face fără a recurge la matricea auxiliară B astfel:

$$\begin{aligned} a_{pj} &= c a_{pj} + s a_{qj}, \quad j = 1, 2, \dots, n, \quad j \neq p, \quad j \neq q, \\ a_{qj} &= a_{jq} = -s a_{jp} + c a_{qj}, \quad j = 1, 2, \dots, n, \quad j \neq p, \quad j \neq q, \\ a_{jp} &= a_{pj}, \quad j = 1, 2, \dots, n, \quad j \neq p, \quad j \neq q, \\ a_{pp} &= a_{pp} + t a_{pq} \\ a_{qq} &= a_{qq} - t a_{pq} \\ a_{pq} &= a_{qp} = 0 \end{aligned} \tag{5}$$

Trecerea de la matricea U la matricea V se face schimbând doar coloanele p și q astfel:

$$\begin{cases} v_{ip} &= c u_{ip} + s u_{iq}, \quad i = 1, 2, \dots, n, \\ v_{iq} &= -s u_{ip} + c u_{iq}, \quad i = 1, 2, \dots, n. \end{cases} \tag{6}$$

Operația se poate face direct în matricea U , fără a recurge la matricea V :

$$\begin{cases} u_{ip} &= c u_{ip} + s u_{iq}, \quad i = 1, 2, \dots, n, \\ u_{iq} &= -s u_{ip}^{veche} + c u_{iq}, \quad i = 1, 2, \dots, n. \end{cases} \quad (7)$$

Observație: Matricea A fiind simetrică se poate memora într-un vector v de dimensiune $\frac{n(n+1)}{2}$. În acest fel se memorează doar partea inferior triunghiulară a matricei. Vectorul v va conține elementele:

$$v : a_{11}, a_{21}, a_{22}, \dots, a_{r1}, a_{r2}, \dots, a_{rr}, \dots, a_{n1}, a_{n2}, \dots, a_{nn}$$

$$v_1 = a_{11}, v_2 = a_{21}, \dots, v_{\frac{n(n+1)}{2}} = a_{nn}$$

restul elementelor din matricea A se regăsesc din relația de simetrie:

$$a_{ij} = a_{ji}$$

Problema ce trebuie rezolvată pentru a ușura scrierea algoritmului lui Jacobi pentru valori proprii, cu memorare vectorială este următoarea:

Pentru orice indici $i, j \in \{1, 2, \dots, n\}$ dați (indici ai elementelor din matricea A) să se găsească indicele $k(i, j) \in \{1, 2, \dots, \frac{n(n+1)}{2}\}$ (indice pentru elemnte din vectorul v) astfel ca:

$$a_{ij} = v_{k(i,j)} \quad (a[i][j] = v[k(i, j)])$$

Descompunerea după valori singulare

(Singular Value Decomposition)

Fie $A \in \mathbb{R}^{p \times n}$. Se numește descompunere după valori singulare a matricei:

$$A = U * S * V^T, \quad U \in \mathbb{R}^{p \times p}, \quad S \in \mathbb{R}^{p \times n}, \quad V \in \mathbb{R}^{n \times n}$$

cu $U = [u_1 \ u_2 \ \dots \ u_p]$ (vectorii u_i sunt coloanele matricei U) și $V = [v_1 \ v_2 \ \dots \ v_n]$ matrice ortogonale iar S matrice de forma:

$$\text{pentru } p \leq n \quad S = \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 & \cdots & 0 \\ \vdots & & & & & \\ 0 & 0 & \cdots & \sigma_p & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{p \times n}$$

$$\text{pentru } p > n \quad S = \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & \sigma_n \\ \vdots & & & \\ 0 & 0 & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{p \times n}$$

unde numerele nenegative $\sigma_i \geq 0, \forall i$ sunt valorile singulare ale matricei A .

Rangul matricei A este numărul de valori singulare strict pozitive:

$$\text{rang}(A) = \text{numărul de valori singulare } \sigma_i > 0.$$

Numărul de condiționare al matricei A este raportul dintre cea mai mare valoare singulară și cea mai mică valoare singulară strict pozitivă.

$$k_2(A) = \frac{\sigma_{\max}}{\sigma_{\min}},$$

$$\sigma_{\max} = \max\{\sigma_i; \sigma_i \text{ valoare singulară}\},$$

$$\sigma_{\min} = \min\{\sigma_i; \sigma_i > 0 \text{ valoare singulară}\}$$

Pseudoinversa Moore-Penrose a matricei A se calculează folosind formula:

$$A^I = V * S^I * U^T.$$

Matricea S^I se calculează folosind formula descrisă mai jos. Presupunem că am calculat pentru matricea $A \in \mathbb{R}^{p \times n}$ descompunerea după valori singulare. Fie $\sigma_1, \sigma_2, \dots, \sigma_r > 0$ valorile singulare strict pozitive ale matricei A , $r = \text{rang}(A)$.

Matricea $S^I \in \mathbb{R}^{n \times p}$ se definește astfel:

$$S^I = \begin{pmatrix} \frac{1}{\sigma_1} & 0 & \dots & 0 & \dots & 0 \\ 0 & \frac{1}{\sigma_2} & \dots & 0 & \dots & 0 \\ \vdots & & & & & \\ 0 & 0 & \dots & \frac{1}{\sigma_r} & \dots & 0 \\ & & & & & \\ 0 & 0 & \dots & 0 & \dots & 0 \\ \vdots & & & & & \\ 0 & 0 & \dots & 0 & \dots & 0 \end{pmatrix} \in \mathbb{R}^{n \times p}.$$

Vectorul $x^I = V * S^I * U^T * b$ poate fi considerat soluția sistemului $Ax = b$ chiar și când $p \neq n$ iar sistemul nu are soluție clasică. Când $p = n$ și matricea A este nesingulară vectorul x^I coincide cu soluția clasică a sistemului $Ax = b$.

Exemple

$$A = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \text{ are valorile proprii } \lambda_1 = -1, \lambda_2 = 0, \lambda_3 = 2$$

$$\lambda_1 = -1, u^1 = \begin{pmatrix} a \\ a \\ -a \end{pmatrix} \quad a \in \mathbb{R} \quad a \neq 0, \quad \lambda_2 = 0, u^2 = \begin{pmatrix} b \\ -b \\ 0 \end{pmatrix} \quad b \in \mathbb{R} \quad b \neq 0$$

$$\lambda_3 = 2 \quad u^3 = \begin{pmatrix} c \\ c \\ 2c \end{pmatrix} \quad c \in \mathbb{R} \quad c \neq 0$$

$$A = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 1 & 2 \\ 2 & 2 & 2 \end{pmatrix} \text{ are valorile proprii } \lambda_1 = 0, \lambda_2 = 2(1-\sqrt{2}), \lambda_3 = 2(1+\sqrt{2})$$

$$2(1 - \sqrt{2}) \approx -0.828427, 2(1 + \sqrt{2}) \approx 4.828437$$

$$\lambda_1 = 0 \quad u^1 = \begin{pmatrix} a \\ -a \\ 0 \end{pmatrix} \quad a \in \mathbb{R} \quad a \neq 0$$

$$A = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \text{ are valorile proprii } \lambda_1 = \lambda_2 = 0, \lambda_3 = \lambda_4 = 2$$

$$\lambda_{1,2} = 0 \quad u^{1,2} = \begin{pmatrix} a \\ b \\ -a \\ -b \end{pmatrix} \quad a, b \in \mathbb{R} \quad , \quad \lambda_{3,4} = 2 \quad u^{3,4} = \begin{pmatrix} c \\ d \\ c \\ d \end{pmatrix} \quad c, d \in \mathbb{R}$$

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 7 \end{pmatrix} \text{ are valorile proprii } \lambda_1 = \lambda_2 = 0, \lambda_{3,4} = 2(4 \pm \sqrt{21})$$

$$2(4 + \sqrt{21}) \approx 17.165151, 2(4 - \sqrt{21}) \approx -1.165151$$

$$\lambda_{1,2} = 0 \quad u^{1,2} = \begin{pmatrix} a \\ b \\ -3a - 2b \\ 2a + b \end{pmatrix} \quad a, b \in \mathbb{R}$$