Tema nr. 5

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

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

  Să se verifice că:

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

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

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

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

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

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

  $$A^{(1)} = (L^0)^T * L^0, (A^{(1)}\text{se obţine înmulţind matricele } (L^0)^T \text{ cu } L^0)$$
  $$A^{(1)} = L^1 * (L^1)^T, \text{ se calculează factorizarea Cholesky a matricei } A^{(1)}$$
  $$A^{(2)} = (L^1)^T * L^1, (A^{(2)}\text{se obţine înmulţind matricele } (L^1)^T \text{ cu } L^1)...$$
  $$A^{(k)} = L^k * (L^k)^T (\text{ factorizare Cholesky a matricei } A^{(k)}) \ ,$$
  $$A^{(k+1)} = (L^k)^T * L^k$$

  Calculele se opresc când diferenţa între două matrice consecutive este suficient de mică ($||A^{(k)} - A^{(k-1)}|| < \epsilon$) sau când s-a depăşit un număr maxim de iteraţii prestabilit ($k > k_{max}$). Nu e nevoie sa memoraţi toate matricele din şir. Afişaţi ultima matrice calculată. Ce formă are? Ce informaţii se găsesc in această matrice? Pentru calculul descompunerii Cholesky folosiţi funcţia din biblioteca numerică pe care o folosiţi. Dacă nu găsiţi o funcţie care să calculeze factorizare Choleski, folosiţi o funcţie care calculează o descompunere LU pentru matrice.

- Pentru $p > n$, utilizând descompunerea după valori singulare (**S**ingular **V**alue **D**ecomposition) din biblioteca folosită la **Tema 2**, să se calculeze şi să se afişeze:
  - valorile singulare ale matricei A,
  - rangul matricei A,
  - numărul de condiţionare al matricei A,
  - pseudoinversa Moore-Penrose a matricei A, $A^I \in \mathbb{R}^{n\times p}$,

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

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

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

  şi afişaţi norma:
  $$\|A^I - A^J\|_1$$

Pentru rangul şi numărul de condiţionare al matricei să se folosească relaţiile descrise în acest fişier şi de asemenea funcţiile din bibliotecă, funcţii care calculează aceste valori.

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

Vectori şi valori proprii - definiţii

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

Valorile proprii ale matricei A pot fi definite şi 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 şi/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 foloseşte relaţia de asemănare. Două matrici A şi 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 şi $B \sim A$. Relaţia de asemănare se foloseşte în algoritmii de aproximare a valorilor proprii deoarece matricele asemenea au acealaşi polinom caracteristic ($p_A(\lambda) \equiv p_B(\lambda)$) şi în consecinţă au aceleaşi valori proprii.

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

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

Construcţia şirului de matrice

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

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

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

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

Şirul de matrice $\{A^{(k)}\} \subseteq \mathbb{R}^{n\times n}$ se construieşte stfel:

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

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

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

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

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

- **unghiul** θ **(**$c = \cos\theta,\ s = \sin\theta,\ t = \mathrm{tg}\theta$**)** este ales astfel ca elementele (p, q) şi (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 şi q (vezi (1)) ;
calculează unghiul θ, adică c, s şi t;
while ($A \neq$ matrice diagonală şi $k \leq k_{max}$)
{
$A = R_{pq}(\theta) * A * R_{pq}^T(\theta)$;
(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 şi q (vezi (1));
calculează unghiul θ, adică c, s şi t ;
(a se vedea formulele (3) şi (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 construieşte matricea B pornind de la matricea A astfel:

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

şi 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:

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

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

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

de unde rezultă :

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

Dacă notăm cu $t = \mathrm{tg}\theta$ avem:

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

rezultă că t satisface ecuaţia:

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

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 ($\theta \in [0, \pi/4]$):

$$t = -\alpha + \mathrm{semn}(\alpha)\sqrt{\alpha^2 + 1} = \left\{ \begin{array}{ll} -\alpha + \sqrt{\alpha^2 + 1} & \text{dacă } \alpha \geq 0 \\ \\ -\alpha - \sqrt{\alpha^2 + 1} & \text{dacă } \alpha < 0 \end{array} \right. \tag{3}$$

$$\mathrm{semn}(\alpha) = \left\{ \begin{array}{ll} 1 & \text{dacă } \alpha \geq 0 \\ -1 & \text{dacă } \alpha < 0 \end{array} \right.$$

Avem:

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

Cazul $a_{pq} = 0$

Ţinând 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 situaţie - pe diagonala matricei A se găsesc aproximările valorilor proprii căutate. Prin urmare testul:

$$A \neq \text{ matrice diagonală}$$

din schema algoritmlui 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 operaţia $A = R_{pq}(\theta) * A * R^T_{pq}(\theta)$ se poate face fără a recurge la matricea auxiliară B astfel:

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

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

$$\left\{ \begin{aligned} v_{ip} &= c\ u_{ip} + s\ u_{iq}\ ,\ i = 1, 2, \ldots, n, \\ v_{iq} &= -s\ u_{ip} + c\ u_{iq}\ ,\ i = 1, 2 \ldots, n. \end{aligned} \right. \tag{6}$$

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

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

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

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

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

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

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

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

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

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

Descompunerea după valori singulare

(**S**ingular **V**alue **D**ecomposition)

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

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

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

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

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

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 condiţionare al matricei A este raportul dintre cea mai mare valoare singulară şi cea mai mică valoare singulară strict pozitivă.

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

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

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

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 $\sigma_1, \sigma_2, \ldots, \sigma_r > 0$ valorile singulare strict pozitive ale matricei A, $r = \text{rang}(A)$.

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

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

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

Exemple

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

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

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

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

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

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

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

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

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

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

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