Calculul valorilor proprii şi vectorilor proprii prin metodele puterii. Metoda Householder

Cuprins

1 Obiective laborator

În urma parcurgerii acestui laborator, studentul va fi capabil să:

- utilizeze metoda puterii directe şi metoda puterii inverse pentru a determina valorile şi vectorii proprii ale unei matrice;
- aplice metoda deflației pentru a determina valorile şi vectorii proprii ale unei matrice;
- aplice proprietățile valorilor şi vectorilor proprii în rezolvarea unor probleme.

2 Noțiuni teoretice

Vom începe prin a prezenta conceptul de valoare proprie și vector propriu pentru o matrice pătrată $A \in \mathbb{C}^{n \times n}$. Se numește *valoare proprie* a lui A orice număr complex $\lambda \in \mathbb{C}$ pentru care există un vector nenul $x \in \mathbb{C}^n$ (*vector propriu*), astfel încât

$$Ax = \lambda x$$

Mulțimea tuturor valorilor proprii ale matricei A se numește *spectrul* matricei și se notează cu

$$\lambda(A) = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}.$$

Două proprietăți importante ale valorilor proprii ale unei matrice $A \in \mathbb{C}^{n \times n}$ sunt:

- $\sum_{i=1}^{n} \lambda_i = \text{tr}(A)$, adică suma valorilor proprii este egală cu urma (suma diagonală) a matricei.
- $\prod_{i=1}^{n} \lambda_i = \det(A)$, adică produsul valorilor proprii este determinantul matricei.

Pentru a determina practic valorile proprii, se consideră polinomul caracteristic $p(\lambda) = \det(\lambda I_n - A)$, iar valorile proprii sunt rădăcinile acestui polinom. Deși acest aspect este esențial din punct de vedere teoretic, în practică se apelează la Metode numerice iterative pentru a deterimna valorile și vectorii proprii.

2.1 Matrice asemenea

Două matrici $A, B \in \mathbb{C}^{n \times n}$ se numesc *asemenea* dacă există o matrice nesingulară T astfel încât

$$B = T^{-1}AT$$

Ideea de asemănare are la bază simplificarea operațiilor. De exemplu, dacă vrem să rotim un vector din $\mathbb{R}^2$ în jurul altui vector, putem ca întâi să schimbăm baza astfel încât vectorul de rotație să devină un vector standard (de exemplu, să-l aducem pe axa x), să aplicăm rotația și apoi să schimbăm baza înapoi.

Două matrice asemenea au același spectru.

Demonstrație. Fie $A, B \in \mathbb{C}^{n \times n}$ asemenea, adică $B = T^{-1}AT$. Fie $\lambda \in \mathbb{C}$ o valoare proprie a lui B.

$$\begin{aligned} Bx &= \lambda x \\ T^{-1}ATy &= \lambda x \\ A(Ty) &= \lambda(Ty) \quad \blacksquare \end{aligned}$$

2.2 Forma Jordan a unei matrice

Forma Jordan a unei matrice este o formă canonică care permite simplificarea analizei matricelor, în special în ceea ce privește valorile și vectorii proprii. O matrice A complexă este mereu similară unei matrice bloc-diagonale de forma:

$$J = \begin{bmatrix} J_1 & 0 & \cdots & 0 \\ 0 & J_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & J_k \end{bmatrix}$$

Fiecare bloc se numește *bloc Jordan* și are forma:

$$J_i = \begin{bmatrix} \lambda_i & 1 & 0 & \cdots & 0 \\ 0 & \lambda_i & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_i & 1 \\ 0 & 0 & \cdots & 0 & \lambda_i \end{bmatrix}$$

2.3 Determinarea vectorilor şi valorilor proprii

În continuare, vom descrie câteva metode numerice utilizate pentru a determina valorile şi vectorii proprii ai unei matrice.

2.3.1 Metoda puterii directe

Fie matricea $A \in \mathbb{R}^{n\times n}$, cu spectrul $\lambda(A) = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ și vectori proprii normalizați $x_1, x_2, \ldots, x_n$. Fie următoarea presupunere:

$$|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$$

Pentru un vector inițial $y^{(0)} \in \mathbb{R}^n$, care are componentă nenulă pe direcția lui x_1, metoda puterii directe construiește șirul

$$y^{(k)} = \frac{Ay^{(k-1)}}{\|Ay^{(k-1)}\|}$$

Șirul $y^{(k)}$ converge către vectorul propriu x_1.

Demonstrație. Fie $P^{-1}AP = J$, unde J este matricea Jordan corespunzătoare lui A. Fie $y^{(0)} = \alpha_1 p_1 + \alpha_2 p_2 + \cdots + \alpha_n p_n$, unde $\alpha_1 \neq 0$ și p_i sunt coloanele matricei P.

$$\begin{aligned} y^{(k)} = \frac{Ay^{(k-1)}}{\|Ay^{(k-1)}\|} &= \frac{A^k y^{(0)}}{\|A^k y^{(0)}\|} \\ &= \frac{P^{-1}J^k P \sum_i \alpha_i p_i}{\|P^{-1}J^k P \sum_i \alpha_i p_i\|} \\ &= \frac{P^{-1}J^k \sum_i \alpha_i e_i}{\|P^{-1}J^k \sum_i \alpha_i e_i\|} \end{aligned}$$

Din expresia de mai sus, se dă factor comun forțat pe λ_1 și la numitor dar și la numărător și trecând la limita $k \to \infty$, ajungem la $y^{(k)} \to v_1$. Mult mai simplu, dacă A este diagonalizabilă, și $b^{(0)}$ este o combinație liniară a vectorilor proprii, atunci:

$$\begin{aligned}
b^{(k)} &= A^k b^{(0)} = A^k \sum_{i=1}^{n} \alpha_i v_i \\
&= \sum_{i=1}^{n} \alpha_i A^k v_i = \sum_{i=1}^{n} \alpha_i \lambda_i^k v_i \\
&= \alpha_1 \lambda_1^k (v_1 + \sum_{i=2}^{n} \frac{\alpha_i}{\alpha_1} \frac{\lambda_i^k}{\lambda_1^k} v_i) \\
b^{(k)} &\to \alpha_1 \lambda_1^k v_1 \quad \blacksquare
\end{aligned}$$

Rata de convergență este dată de $\frac{|\lambda_2|}{|\lambda_1|}$, iar convergența este rapidă dacă $|\lambda_2| \ll |\lambda_1|$.

Algorithm 1 Metoda puterii directe

$v \leftarrow$ vectorul initial normalizat
for $i = 1$ to max_iter **do**
  $v_{\text{prev}} \leftarrow v$
  $v \leftarrow A \cdot v$
  $v \leftarrow \frac{v}{\|v\|}$
  if $\|v - v_{\text{prev}}\| <$ tol **then**
    break
  end if
end for
$\lambda \leftarrow v^\top A v$

2.3.2 Metoda puterii inverse

Metoda puterii inverse aplică ideea puterii directe la matricea $B = A^{-1}$. Valorile proprii ale lui B sunt inversul valorilor proprii ale lui A, deci cea mai mică valoare proprie a lui A corespunde celei mai mari valori proprii ale matricei B. În plus, se poate introduce o deplasare μ, astfel încât $B = (A - \mu I)^{-1}$. Dacă μ se află în apropierea uneia dintre valorile proprii ale lui A, atunci iterația va converge către vectorul propriu asociat lui λ_j, cu $|\lambda_j - \mu|$ minim.

Algorithm 2 Metoda Puterii Inverse

$v \leftarrow$ vectorul initial normalizat
for $i = 1$ to max_iter **do**
  $v_{\text{prev}} \leftarrow v$
  Rezolv $A \cdot v \leftarrow v_{\text{prev}}$ pentru v
  $v \leftarrow \frac{v}{\|v\|}$
  if $\|v - v_{\text{prev}}\| <$ tol **then**
    break
  end if
end for
$\lambda \leftarrow v^\top A v$

Algorithm 3 Metoda Puterii Inverse cu deplasare

$v \leftarrow$ normalized initial vector
for $i = 1$ to max_iter **do**
  $v_{\text{prev}} \leftarrow v$
  Rezolv $(A - \mu I) \cdot v = v_{\text{prev}}$ pentru v
  $v \leftarrow \frac{v}{\|v\|}$
  if $\|v - v_{\text{prev}}\| < \text{tol}$ **then**
    break
  end if
end for
$\lambda \leftarrow v^\top A v$

Iterarea câtului Rayleigh. O versiune mult mai eficientă a metodei puterii inverse o constituie *iterarea câtului Rayleigh* (sau metoda puterii inverse cu deplasare variabilă). Aceasta se aplică matricelor hermitice (simetrice). Pentru o aproximație curentă $x^{(k)}$ a vectorului propriu, se calculează valoarea:

$$\rho^{(k)} = \frac{x^{(k)T} A x^{(k)}}{x^{(k)T} x^{(k)}}.$$

Această mărime, numită *cât Rayleigh*, reprezintă o estimare a valorii proprii asociate lui $x^{(k)}$. Apoi, în pasul următor, se folosește $\mu = \rho^{(k)}$ drept noua deplasare. Cu alte cuvinte, la fiecare iterație:

$$\begin{aligned} y^{(k)} &= (A - \rho^{(k)} I)^{-1} x^{(k)} \\ (A - \rho^{(k)} I) y^{(k)} &= x^{(k)} \\ x^{(k+1)} &= \frac{y^{(k)}}{\|y^{(k)}\|} \end{aligned}$$

Repetând acest procedeu, vectorul și valoarea proprie estimate converg rapid spre vectorul și valoarea reală asociate lui λ_j. Această strategie este extrem de utilă atunci când avem deja un indiciu despre poziția unei valori proprii, întrucât viteza de convergență devine foarte mare după ce $\rho^{(k)}$ se apropie suficient de λ_j. În practică, iterarea câtului Rayleigh se folosește pe scară largă în algoritmi de diagonalizare numerică și este adesea componenta centrală în metode avansate de calculul valorilor proprii.

Algorithm 4 Iterarea Rayleigh

$v \leftarrow \frac{v}{\|v\|}$
for $i = 1$ to max_iter **do**
  $v_{\text{prev}} \leftarrow v$
  $\mu \leftarrow v^\top A v$
  Rezolv $(A - \mu I) v = v_{\text{prev}}$ pentru v
  $v \leftarrow \frac{v}{\|v\|}$
  if $\|v - v_{\text{prev}}\| < \text{tol}$ **then**
    break
  end if
end for
$\lambda \leftarrow v^\top A v$

2.4 Deflația

Deflația ne permite ca după ce s-a găsit o valoare proprie și un vector propriu asociat, să se "reducă" problema la o submatrice pentru determinarea celorlalte valori proprii.

Fie $\lambda_1, \lambda_2, \ldots, \lambda_n$ valorile proprii ale matricei A cu vectorii proprii asociați $v_1, v_2, \ldots, v_n$. Fie x un vector astfel încât $x^T v_1 = 1$. Se consideră matricea:

$$B = A - \lambda_1 v_1 x^T$$

Spectrul matricei B este dat de:

$$\sigma(B) = \{0, \lambda_2, \lambda_3, \ldots, \lambda_n\}$$

Cum alegem vectorul x? O abordare simplă este *deflația Wielandt*:

- Se consideră că am aflat valoarea proprie dominantă λ_1 și vectorul propriu asociat v_1 (de pildă, prin metoda puterii directe).
- Considerăm $x = \frac{1}{\lambda_1 v_i^{(i)}} A_i$. A_i este linia i din matricea A, iar $v_i^{(i)}$ este componenta i a vectorului propriu v_i. Se obsevă că $x^T c_i = 1$. Construind, matricea B, linia i devine 0.
- Eliminăm linia i și coloana i din matricea B, obținând o matrice de dimensiune $(n-1) \times (n-1)$.

Algorithm 5 MP cu deflație

eigarray $\leftarrow$ zero vector of length n
while size$(A, 1) > 0$ **do**
  $v \leftarrow$ vector de 1 de lungime size$(A, 1)$
  $v \leftarrow v / \|v\|$
  $v \leftarrow$ vector obtinut din MP
  $\lambda \leftarrow v^\top A v$
  eigarray[size$(A, 1)$] $\leftarrow \lambda$
  $y \leftarrow A_{1,:} / (\lambda \cdot v_1)$
  $A \leftarrow A - \lambda \cdot (v \cdot y)$
  $A \leftarrow A_{2:\text{end},2:\text{end}}$
end while

2.5 Algoritmul PageRank

Matricea Google este un exemplu faimos de matrice stocastică, folosită în *algoritmul PageRank*. Ideea de bază este că fiecare pagină web este un nod, iar legăturile (link-urile) către alte pagini definesc o probabilitate de *tranziție* de la o pagină la alta. Pentru a rezolva probleme de tip "pagină izolată" sau sub-componente care nu influențează semnificativ restul rețelei, se folosesc tehnici de *deflație*, prin care se separă anumite sub-blocuri ale matricei și se rezolvă problema pe blocuri mai mici.

- **Cum se construiește Matricea Google?** Se consideră că avem n pagini web, notate $P_1, P_2, \ldots, P_n$. Dacă pagina P_i are link-uri către paginile P_j, atunci probabilitatea de a sări din P_i în P_j (prin navigare directă) va fi reprezentată ca un element nenul în matricea noastră. Pentru a asigura că matricea este stocastică (fiecare coloană însumează 1), se normalizează după numărul de link-uri ieșire.
- **Exemplu concret cu $N = 4$ pagini.** Considerăm 4 pagini web: P_1, P_2, P_3, P_4 și următoarea structură de link-uri:

$$P_1 \to \{P_2, P_3\}, \quad P_2 \to \{P_3\}, \quad P_3 \to \{P_1, P_4\}, \quad P_4 \to \{P_2\}.$$

  Matricea de *adiacență* (prin linii) ar putea arăta astfel:

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

Pentru a obține *matricea de stocastică* M, fiecare coloană se normalizează la 1.

$$M = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1/2 & 0 & 0 \end{bmatrix}.$$

- **Calculul PageRank.** PageRank-ul este o distribuție de probabilitate care indică importanța fiecărei pagini web. PageRank-ul rămâne neschimbat indiferent de ce face utilizatorul, deci satisface ecuația:

  $$MR = R$$

  Totuși, există șansa ca utilizatorul să nu continue click-urile. Fie d probabilitatea de a continua să navigheze și $1-d$ probabilitatea de a sări la o pagină aleatorie. Astfel, definim matricea Google:

  $$G = dM + \frac{(1-d)}{N} ONES(N)$$

  PageRank-ul va fi vectorul propriu asociat lui $\lambda = 1$ al matricei Google G. Cum $\lambda = 1$ este și valoare proprie dominantă, se poate aplica metoda puterii.

- **Deflația în PageRank.** Dacă anumite pagini sunt "izolate" sau formează sub-componente, se pot crea blocuri separate pentru care vectorul propriu asociat lui $\lambda = 1$ se calculează ușor, după care aceste blocuri pot fi *deflate* pentru a reduce dimensiunea problemei și, implicit, efortul de calcul.

Algorithm 6 PageRank

$n \leftarrow$ number of rows of M
$v \leftarrow \mathbf{1}_n$
$G \leftarrow d \cdot M + \frac{1-d}{n} \cdot \mathbf{1}_{n\times n}$
while true **do**
    $v_{\text{prev}} \leftarrow v$
    $v \leftarrow G \cdot v$
    $v \leftarrow \frac{v}{\|v\|}$
    if $\|v - v_{\text{prev}}\| <$ tol **then**
        break
    end if
end while
$v \leftarrow \frac{v}{\|v\|_1}$

3 Probleme

1. Fie matricea tridiagonală:

$$A = \begin{bmatrix} a_1 & c_1 & \dots & & \\ b_2 & a_2 & c_2 & \dots & \\ \vdots & \vdots & \ddots & \ddots & \\ & & b_{n-1} & a_{n-1} & c_{n-1} \\ & & & b_n & a_n \end{bmatrix}.$$

Se construieşte şirul de polinoame:

$$p_0(\lambda) = 1, \quad p_1(\lambda) = \lambda - a_1, \quad p_n(\lambda) = (\lambda - a_n)\, p_{n-1}(\lambda) - b_n c_{n-1}\, p_{n-2}(\lambda).$$

1. Arătaţi că $p_n(\lambda)$ este polinomul caracteristic al matricei A.
2. Scrieţi o funcţie OCTAVE care calculează valorile şi vectorii proprii ale matricei A.

2. Să se demonstreze pentru o matrice $A \in \mathbb{R}^{n\times n}$ proprietăţile următoare:
   1. $\sigma(A^{-1}) = \left\{\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \ldots, \frac{1}{\lambda_n}\right\}$;
   2. $\sigma(A - \mu I_n) = \{\lambda_i - \mu\}, \forall \mu \in \mathbb{R}$;
   3. $\sigma(A^k) = \left\{\lambda_i^k\right\}, \forall k \in \mathbb{N}$;
   4. $\sigma((A - \mu I_n)^{-1}) = \left\{\frac{1}{\lambda_i - \mu}\right\}, \forall \mu \in \mathbb{R}$.

3. Calculaţi valorile şi vectorii proprii ai unui reflector Householder.

   Indicaţie: $G^T u = e_1 G$ - este un reflector. Coloanele lui G sunt vectori proprii.

4. Pentru matricea:

$$A = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \in \mathbb{R}^{2\times 2}, \quad c^2 + s^2 = 1,$$

   calculaţi valorile şi vectorii proprii.