Algebră Liniară și Optimizare

Curs 9

Conf. dr. Anca Ignat

Conf. dr. Corina Forăscu

April 25, 2026

Valori și vectori proprii
(Eigenvalues, eigenvectors)

Valori și vectori proprii

Definiție

Fie $A \in \mathbb{R}^{n \times n}$. Numărul complex $\lambda \in \mathbb{C}$ se numește **valoare proprie** a matricei A dacă există un vector $\mathbf{u} \in \mathbb{C}^n$, $\mathbf{u} \neq \mathbf{0}$, astfel încât:

$$A\mathbf{u} = \lambda \mathbf{u}$$

Vectorul $\mathbf{u}$ se numește *vector propriu* asociat valorii proprii λ.

Pentru existența unui vector propriu $\mathbf{u} \neq 0$ este necesar și suficient ca matricea $\lambda I_n - A$ să fie singulară, adică: $\det(\lambda I_n - A) = 0$.

Valori şi vectori proprii

Polinomul de grad n:

$$p_A(\lambda) = \det(\lambda I_n - A)$$

se numește **polinomul caracteristic** al matricei A.

Propoziția 1

Fie rădăcinile polinomului caracteristic $\lambda_1, \lambda_2, \ldots, \lambda_n$ distincte, adică $\lambda_i \neq \lambda_j$ pentru $1 \leq i < j \leq n$, și fie $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ vectorii proprii corespunzători. Atunci vectorii $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ sunt liniar independenți.

(Demonstrația Propoziţiei se face prin inducție.)

Diagonalizarea matricelor

Propoziţia 2

Fie valorile proprii λ_i ale matricei $A \in \mathbb{R}^{n\times n}$ distincte. Atunci există o matrice nesingulară T astfel ca:

$$T^{-1}AT = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$$

Demonstrație: Fie $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ vectorii proprii ai matricei A. Construim matricea T ale cărei coloane sunt vectorii proprii $\mathbf{u}_i$, $T = [\mathbf{u}_1\ \mathbf{u}_2\ \ldots\ \mathbf{u}_n]$.

Deoarece vectorii proprii sunt liniar independenți (conform Propoziției 1), rezultă că matricea T este nesingulară.

Diagonalizarea matricelor

Demonstraţie: (continuare)

Avem:

$$AT = [A\mathbf{u}_1\ A\mathbf{u}_2\ \dots\ A\mathbf{u}_n] = [\lambda_1\mathbf{u}_1\ \lambda_2\mathbf{u}_2\ \dots\ \lambda_n\mathbf{u}_n]$$

$$= T * \mathrm{diag}[\lambda_1\ \lambda_2\ \dots\ \lambda_n]$$

Înmulțind la stânga cu T^{-1}, obținem:

$$T^{-1}AT = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_n).$$

Matrici asemenea

Definiţie

Matricele A şi B sunt **asemenea** (notaţie $A \sim B$) dacă şi numai dacă există o matrice nesingulară T ($\det T \neq 0$) astfel ca:

$$A = TBT^{-1}.$$

Propoziţia 3

$$A \sim B \Rightarrow p_A(\lambda) = p_B(\lambda).$$

Demonstraţie:

$$p_A(\lambda) = \det(\lambda I_n - A) = \det(\lambda I_n - TBT^{-1}) = \det(\lambda TT^{-1} - TBT^{-1})$$
$$= \det\left(T(\lambda I_n - B)T^{-1}\right) = \det(T)\det(\lambda I_n - B)\det(T^{-1}) = p_B(\lambda)$$

Propoziţia 3 ne spune că matricele asemenea au acelaşi polinom caracteristic şi aceleaşi valori proprii.

Teorema lui Gershgorin

Teorema lui Gershgorin

Fie $A \in \mathbb{R}^{n \times n}$ şi $\lambda \in \mathbb{C}$ o valoare proprie oarecare a matricei A. Atunci:

$$\exists\, i_0 \in \{1, 2, \ldots, n\} \text{ astfel încât } |\lambda - a_{i_0 i_0}| \leq r_{i_0},\, r_{i_0} = \sum_{\substack{j=1 \\ j \neq i_0}}^{n} |a_{i_0 j}|.$$

(Valoarea proprie λ se află în cercul din planul complex de centru $a_{i_0 i_0}$ şi rază r_{i_0}.)

Teorema lui Gershgorin

Demonstraţie: Fie $\lambda \in \mathbb{C}$ o valoare proprie a matricei A şi $\mathbf{u} \neq \mathbf{0}$ un vector propriu asociat valorii proprii λ, $A\mathbf{u} = \lambda\mathbf{u}$. Avem

$$\lambda \mathbf{u}_i = \sum_{j=1}^{n} a_{ij}\mathbf{u}_j \iff (\lambda - a_{ii})\mathbf{u}_i = \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}\mathbf{u}_j, \quad i = 1, \ldots, n.$$

Fie i_0 astfel ca:

$$|\mathbf{u}_{i_0}| = \|\mathbf{u}\|_\infty = \max\{|\mathbf{u}_k|\ ;\ k = 1, \ldots, n\} > 0, (\mathbf{u} \neq \mathbf{0})$$

Vom avea:

$$|\lambda - a_{i_0 i_0}| = \left|\sum_{\substack{j=1 \\ j\neq i_0}}^{n} a_{i_0 j}\frac{u_j}{u_{i_0}}\right| \leq \sum_{\substack{j=1 \\ j\leq i_0}}^{n} |a_{i_0 j}|\frac{|u_j|}{|u_{i_0}|} \leq r_{i_0} \text{ deoarece } \frac{|u_j|}{|u_{i_0}|} \leq 1.$$

$\square$

Observaţie: Presupunem că matricea A are n vectori proprii liniar independenţi $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n$ asociaţi valorilor proprii $\lambda_1, \lambda_2, \dots, \lambda_n$.

Fie: $U = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_n]$ Datorită independenţei vectorilor $\mathbf{u}_k$, rezultă că matricea U este nesingulară şi avem:

$$\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_n) \qquad\qquad U^{-1}AU = \Lambda.$$

Considerăm matricea perturbată:

$$A(\varepsilon) = A + \varepsilon B$$

$$U^{-1}A(\varepsilon)U = \Lambda + \varepsilon U^{-1}BU == \Lambda + \varepsilon C$$

Rezultă că $A(\varepsilon) \sim U^{-1}A(\varepsilon)U \Rightarrow$ au aceleaşi valori proprii $\lambda_i(\varepsilon)$.

$$|\lambda_i(\varepsilon) - \lambda_i - \varepsilon c_{ii}| \leq \varepsilon \sum_{\substack{j=1 \\ j\neq i}}^{n} |c_{ij}| \Rightarrow |\lambda(\varepsilon) - \lambda_i| = \mathcal{O}(\varepsilon).$$

Metoda puterii pentru matrice simetrice

Metoda puterii pentru matrice simetrice

Propoziţie

Fie $A \in \mathbb{R}^{n\times n}$, $A^T = A$. Atunci toate valorile proprii ale matricei A sunt numere reale.

Demonstraţie: Fie $\lambda \in \mathbb{C}$ și $\mathbf{u} \in \mathbb{C}^n$, $\mathbf{u} \neq 0$, astfel încât $A\mathbf{u} = \lambda\mathbf{u}$.

Considerăm produsul scalar:

$$(A\mathbf{u}, \mathbf{u})_{\mathbb{C}^n} = \lambda(\mathbf{u}, \mathbf{u})_{\mathbb{C}^n} = \lambda\|\mathbf{u}\|_2^2.$$

Dar deoarece $A^T = A$, avem:

$$(A\mathbf{u}, \mathbf{u})_{\mathbb{C}^n} = (\mathbf{u}, A^T\mathbf{u})_{\mathbb{C}^n} = (\mathbf{u}, A\mathbf{u})_{\mathbb{C}^n} = \overline{(A\mathbf{u}, \mathbf{u})_{\mathbb{C}^n}} \Rightarrow (A\mathbf{u}, \mathbf{u})_{\mathbb{C}^n} \in \mathbb{R}.$$

Rezultă:

$$\lambda = \frac{(A\mathbf{u}, \mathbf{u})_{\mathbb{C}^n}}{\|\mathbf{u}\|_2^2} \in \mathbb{R}.$$

Metoda puterii pentru matrice simetrice

Propoziţie

Fie $A \in \mathbb{R}^{n\times n}$, $A = A^T$. *Atunci există o bază ortonormată de vectori proprii ai matricei* A, $\{\mathbf{u}^1, \mathbf{u}^2, \dots, \mathbf{u}^n\}$ *astfel încât*

$$(\mathbf{u}^i, \mathbf{u}^j) = \delta_{ij} = \begin{cases} 1, & \text{dacă } i = j \\ 0, & \text{dacă } i \neq j \end{cases}$$

Echivalent, putem scrie că există vectori proprii $\{\mathbf{u}^1, \mathbf{u}^2, \dots, \mathbf{u}^n\}$ asociaţi valorilor proprii reale $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ astfel ca:

$$AU = U\Lambda \quad \Leftrightarrow \quad U^T AU = \Lambda$$

cu $\Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$, şi $U = [\mathbf{u}_1\ \mathbf{u}_2\ \dots\ \mathbf{u}_n]$ matrice ortogonală.

Metoda puterii pentru matrice simetrice

Definiţie

Se numește *coeficient Rayleigh* al vectorului $\mathbf{u} \in \mathbb{R}^n$ pentru matricea A următoarea mărime scalară:

$$r(\mathbf{u}) = \frac{\mathbf{u}^T A \mathbf{u}}{\mathbf{u}^T \mathbf{u}} = \frac{(A\mathbf{u}, \mathbf{u})_{\mathbb{R}^n}}{(\mathbf{u}, \mathbf{u})_{\mathbb{R}^n}} = \frac{(A\mathbf{u}, \mathbf{u})_{\mathbb{R}^n}}{\|\mathbf{u}\|_2^2}.$$

Se verifică ușor că dacă $\mathbf{u} \in \mathbb{R}^n$ este vector propriu al matricei A asociat valorii proprii λ, atunci $r(\mathbf{u}) = \lambda$.

Fie $A \in \mathbb{R}^{n\times n}$, $A = A^T$. Matricea are valori proprii reale $\lambda_1, \lambda_2, \ldots, \lambda_n$. Presupunem în plus că:

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n| \geq 0.$$

Vectorii proprii se notează $\mathbf{u}^1, \mathbf{u}^2, \ldots, \mathbf{u}^n$.

Metoda puterii pentru matrice simetrice

Metoda puterii este un algoritm care aproximează valoarea proprie de modul maxim λ_1 și un vector propriu asociat.
Se pornește de la un vector nenul de normă euclidiană 1, $\mathbf{u}^{(0)} \in \mathbb{R}^n, \quad \|\mathbf{u}^{(0)}\|_2 = 1$, și se construiește următorul șir de vectori de normă euclidiană 1:

$$\mathbf{u}^{(0)}, \quad \mathbf{u}^{(1)} = \frac{1}{\|A\mathbf{u}^{(0)}\|_2} A\mathbf{u}^{(0)}, \quad \mathbf{u}^{(2)} = \frac{1}{\|A\mathbf{u}^{(1)}\|_2} A\mathbf{u}^{(1)} \quad \ldots$$

$$\mathbf{u}^{(k)} = \frac{1}{\|A\mathbf{u}^{(k-1)}\|_2} A\mathbf{u}^{(k-1)},$$

În anumite condiții, acest șir converge la un vector propriu asociat valorii proprii λ_1, iar coeficienții Rayleigh corespunzători converg către λ_1.

Metoda puterii pentru matrice simetrice

Teoremă

Fie $A \in \mathbb{R}^{n\times n}$ o matrice simetrică pentru care valorile proprii îndeplinesc condiția:

$$|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n| \geq 0.$$

Dacă $\mathbf{u}^{(0)} \in \mathbb{R}^n$, $\|\mathbf{u}^{(0)}\|_2 = 1$ și $(\mathbf{u}^{(0)}, \mathbf{u}^1) \neq 0$, unde $\mathbf{u}^1$ este vector propriu asociat lui λ_1, atunci:

$$\mathbf{u}^{(k)} = \frac{1}{\|A^k\mathbf{u}^{(0)}\|_2} A^k\mathbf{u}^{(0)} \to \mathbf{u}^1, \textit{(vector propriu asociat lui } \lambda_1\textit{)}$$

iar coeficienții Rayleigh satisfac:

$$r(\mathbf{u}^{(k)}) \to \lambda_1.$$

Metoda puterii pentru matrice simetrice

Demonstrație. Fie $\{\mathbf{u}^1, \mathbf{u}^2, \ldots, \mathbf{u}^n\}$ vectori proprii asociați valorilor proprii $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$, care formează o bază ortonormată în $\mathbb{R}^n$.
Atunci avem

$$\mathbf{u}^{(0)} = a_1\mathbf{u}^1 + a_2\mathbf{u}^2 + \cdots + a_n\mathbf{u}^n,\ a_i \in \mathbb{R}.$$

Deoarece $(\mathbf{u}^{(0)}, \mathbf{u}^1)_{\mathbb{R}^n} \neq 0$, rezultă că $a_1 \neq 0$.
Din construcția șirului $\mathbf{u}^{(k)}$, deducem că există o constantă c_k astfel încât:

$$\begin{aligned}
\mathbf{u}^{(k)} = c_k A^k \mathbf{u}^{(0)} &= c_k A^k (a_1\mathbf{u}^1 + a_2\mathbf{u}^2 + \ldots + a_n\mathbf{u}^n) \\
&= c_k (a_1\lambda_1^k\mathbf{u}^1 + a_2\lambda_2^k\mathbf{u}^2 + \ldots + a_n\lambda_n^k\mathbf{u}^n) \\
&= c_k\lambda_1^k \left[a_1\mathbf{u}^1 + a_2\left(\frac{\lambda_2}{\lambda_1}\right)^k \mathbf{u}^2 + \ldots + +a_n\left(\frac{\lambda_n}{\lambda_1}\right)^k \mathbf{u}^n \right]
\end{aligned}$$

Metoda puterii pentru matrice simetrice

Demonstrație.(continuare)

Din această ultimă relaţie, din faptul că λ_1 este valoare proprie dominantă şi $a_1 \neq 0$ deducem că pentru k suficient de mare vectorul $\mathbf{u}^{(k)}$ se aliniază după vectorul propriu $\mathbf{u}^1$:

$$\mathbf{u}^{(k)} \approx c_k \lambda_1^k a_1 \mathbf{u}^1.$$

Metoda puterii

$\mathbf{u}^{(0)} \in \mathbb{R}^n, \; \|\mathbf{u}^{(0)}\|_2 = 1;$

k=0;

do

- $k++;$
- $\mathbf{w} = A\mathbf{u}^{k-1};$
- $\mathbf{u}^{(k)} = \dfrac{1}{\|\mathbf{w}\|_2}\mathbf{w};$
- $\lambda_k = r(\mathbf{u}^{(k)}) = (A\mathbf{u}^k, \mathbf{u}^k)_{\mathbb{R}^n};$

while ($\|A\mathbf{u}^k - \lambda_k \mathbf{u}^k\| > \varepsilon$ şi $k \leq k_{max}$);

Metoda iteraţiei inverse

Metoda iterației inverse

Considerăm o matrice simetrică $A \in \mathbb{R}^{n\times n}$, cu $A = A^T$, și fie $\mu \in \mathbb{R}$ un număr real care nu este valoare proprie a lui A.

Vom folosi metoda puterii pentru a aproxima valoarea proprie a matricei A cea mai apropiată de μ și un vector propriu asociat.

$$\mu \neq \text{ valoare proprie } \rightarrow \det(A - \mu I_n) \neq 0 \;\Rightarrow\; \exists (A - \mu I_n)^{-1}.$$

Metoda iterației inverse

Fie $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ valorile proprii reale ale matricei A.
Valorile proprii ale matricei $(A - \mu I_n)^{-1}$ sunt:

$$\left\{ \frac{1}{\lambda_1 - \mu}, \frac{1}{\lambda_2 - \mu}, \dots, \frac{1}{\lambda_n - \mu} \right\}$$

Matricele A și $(A - \mu I_n)^{-1}$ au aceiași vectori proprii.

Metoda iterației inverse

Presupunem că λ_I este valoarea proprie cea mai apropiată de μ (și singura). Atunci avem

$$\frac{1}{|\lambda_I - \mu|} > \frac{1}{|\lambda_j - \mu|}, \quad \forall j \neq I. \tag{*}$$

Relaţia (*) sugerează aplicarea metodei puterii pentru matricea $(A - \mu I_n)^{-1}$, pentru a aproxima valoarea proprie $(\lambda_I - \mu)^{-1}$ și un vector propriu asociat.

Algoritmul conduce la aproximarea valorii proprii cea mai apropiată de μ, λ_I, și a unui vector propriu asociat acesteia, u^I.

Metoda iteraţiei inverse

$\mathbf{u}^{(0)} \in \mathbb{R}^n, \ \|\mathbf{u}^{(0)}\|_2 = 1;$

k=0;

do

- $k++;$
- Se rezolvă sistemul $(A - \mu I_n)\mathbf{w} = \mathbf{u}^{(k-1)};$
- $\mathbf{u}^{(k)} = \dfrac{1}{\|\mathbf{w}\|_2}\mathbf{w};$
- $\lambda_k = r(\mathbf{u}^{(k)}) = (A\mathbf{u}^k, \mathbf{u}^k)_{\mathbb{R}^n};$

while ($\|A\mathbf{u}^k - \lambda_k \mathbf{u}^k\| > \varepsilon$ şi $k \leq k_{max}$);

Forma superioară Hessenberg

Forma superioară Hessenberg

Spunem că o matrice $H \in \mathbb{R}^{n\times n}$ este în *formă Hessenberg superioară* dacă:

$$h_{ij} = 0, \quad \text{pentru } i = 1, \ldots, n,\ j = 1, \ldots, i-2,$$

O matrice în formă Hessenberg superioară are forma:

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} & \cdots & h_{1\ n-1} & h_{1n} \\ h_{21} & h_{22} & h_{23} & \cdots & h_{2\ n-1} & h_{2n} \\ 0 & h_{32} & h_{33} & \cdots & h_{3\ n-1} & h_{3n} \\ 0 & 0 & h_{43} & \cdots & h_{4\ n-1} & h_{4n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & h_{n\ n-1} & h_{nn} \end{pmatrix}$$

Forma superioară Hessenberg

Ne interesează un algoritm care să transforme o matrice pătratică oarecare A într-o matrice Hessenberg superioară H, care să aibă aceleași valori proprii.

Dorim o transformare de forma:

$$A \longrightarrow H \text{ a.î. } H \sim A, \; H = \tilde{P} A \tilde{P}^{-1},$$

cu $\tilde{P}$ o matrice nesingulară.

Algoritmul este o adaptare a metodei lui Householder și se desfășoară în $n-2$ pași, folosind matrici de reflexie pentru a transforma matricea.

Reducerea la forma Hessenberg superioară

Algoritmul de reducere la forma Hessenberg superioară constă în:

Pas 1: Se calculează

$$A^{(1)} = P_1 A P_1$$

Matricea P_1 se alege astfel încât coloana 1 să fie adusă în formă Hessenberg superioară.

Pas 2:

$$A^{(2)} = P_2 A^{(1)} P_2 = P_2 (P_1 A^{init} P_1) P_2$$

Matricea P_2 transformă coloana 2 în formă Hessenberg superioară fără să modifice coloana 1.

Reducerea la forma Hessenberg superioară

Algoritmul de reducere la forma Hessenberg superioară constă în:

Pasul r:

$$A^{(r)} = P_r A^{(r-1)} P_r = P_r (P_{r-1} \cdots (P_1 A^{init} P_1) \cdots P_{r-1}) P_r$$

Se transformă coloana r în formă Hessenberg superioară fără a modifica primele $r-1$ coloane.

Reducerea la forma Hessenberg superioară

Pasul general r**, unde** $r = 1, 2, \ldots, n-2$**:**

La intrarea în pasul r, matricea A are primele $r-1$ coloane în formă Hessenberg superioară. La ieșire, va avea primele r coloane în această formă:

$$A_{\text{ies}} = P_r A_{\text{intr}} P_r, \quad A_{ies} \sim A_{intr}.$$

$$P_r = I - 2\mathbf{v}^r (\mathbf{v}^r)^T, \quad \mathbf{v}^r \in \mathbb{R}^n, \quad \|\mathbf{v}^r\|_2 = 1$$

Vectorul $\mathbf{v}^r$ se alege astfel încât matricea A_{ies} să aibă coloana r în formă Hessenberg superioară, şi să nu schimbe primele $r-1$ coloane ale matricii A_{intr}.

Calculul matricii P_r:

$$P_r = I - \frac{1}{\beta}\mathbf{u}\mathbf{u}^T$$

$$\beta = \sigma - k \cdot a_{r+1\ r},$$

$$k^2 = \sigma = a_{r+1\ r}^2 + \cdots + a_{i\ r}^2 + \cdots + a_{nr}^2 = \sum_{i=r+1}^{n} a_{ir}^2 \Rightarrow k = \pm\sqrt{\sigma}$$

$$\text{semn}\, k = -\ \text{semn}\, a_{r+1r}$$

$$\mathbf{u} := \begin{pmatrix} 0 \\ \vdots \\ a_{r+1\ r} - k \\ \vdots \\ a_{ir} \\ \vdots \\ a_{nr} \end{pmatrix}, \qquad \beta = 0 \rightarrow r = r + 1\ (P_r = I_n).$$

Aplicarea transformării $P_r A$

Algoritmul de trecere de la matricea A la matricea $P_r A$ este următorul:

$$(P_r A)\mathbf{e}_j = \begin{cases} A\mathbf{e}_j, & \text{pentru } j = 1, \dots, r-1 \\ (a_{1r}, a_{2r}, \dots, a_{rr}, k, 0, \dots, 0)^T & \text{pentru } j = r \\ A\mathbf{e}_j - \dfrac{\gamma_j}{\beta}\,\mathbf{u}, & \text{pentru } j = r+1, \dots, n \end{cases}$$

unde

$$\gamma_j = (A\mathbf{e}_j, \mathbf{u})_{\mathbb{R}^n} = \sum_{i=r+1}^{n} a_{ij} u_i$$

și vectorul **u** are componentele:
$u_i = 0\,,\ i = 1, \dots, r,\ u_{r+1} = a_{r+1\ r} - k,\ u_i = a_{ir},\ i = r+2, \dots, n.$

Aplicarea transformării $P_r A$

Vom descrie în continuare cum se efectuează operația

$$A := AP_r$$

fără a face înmulțire matricială, unde A este matricea obținută anterior, având primele r coloane în formă Hessenberg superioară.

Vom arăta că această operație nu schimbă forma Hessenberg superioară deja obținută. Vom pune în evidenţă transformările liniilor matricii A.

Pentru $i = 1, \ldots, n$, avem:

$$\mathbf{e}_i^T(AP) = \text{ noua linia } i \text{ a matricii } AP = (\mathbf{e}_i^T A)\left(I_n - \frac{1}{\beta}\mathbf{u}\mathbf{u}^T\right)$$

$$= \mathbf{e}_i^T A - \frac{1}{\beta}(\mathbf{e}_i^T A)\mathbf{u}\mathbf{u}^T = \mathbf{e}_i^T A - \frac{\gamma_i}{\beta}\mathbf{u}^T.$$

Aplicarea transformării $P_r A$

unde $\gamma_i = (\mathbf{e}_i^T A)\mathbf{u} = a_{i\ r+1}\mathbf{u}_{r+1} + \ldots + a_{in}\mathbf{u}_n$.

Elementele liniei i se schimbă astfel:

$$a_{ij} = a_{ij} - \frac{\gamma_i}{\beta}\mathbf{u}_j,\ j = r+1, \ldots, n,\ i = 1, \ldots, n.$$

Operaţia $A := AP_r$ nu modifică primele r coloane ale matricii A, ele rămânând în formă superior Hessenberg.

Algoritmul de obţinere a formei superior Hessenberg (1/2)

for $r = 1, \ldots, n-2$

/ / construcţia matricii P_r - constanta β şi vectorul $\mathbf{u}$

- $\sigma = \sum_{i=r+1}^{n} a_{ir}^2;$
- if $(\sigma \leq \varepsilon)$ break; / / $r = r+1 \Leftrightarrow P_r = I_n$
- $k = \sqrt{\sigma};$
- if $(a_{r+1\ r} > 0)$ $k = -k;$
- $\beta = \sigma - k \cdot a_{r+1\ r};$
- $\mathbf{u}_{r+1} = a_{r+1\ r} - k;$ $\mathbf{u}_r = a_{ir},\ i = r+2, \ldots, n;$

Algoritmul de obţinere a formei superior Hessenberg (2/3)

/ / $A = P_r * A$

/ / transformarea coloanelor $j = r + 1, \ldots, n$

- for $j = r + 1, \ldots, n$

$$\begin{cases} \gamma = (\gamma_j/\beta) = (A\mathbf{e}_j, \mathbf{u})/\beta = \left(\sum_{i=r+1}^{n} \mathbf{u}_i a_{ij}\right)/\beta; \\ \text{for } i = r+1, \ldots, n \\ \qquad a_{ij} = a_{ij} - \gamma * \mathbf{u}_i; \end{cases}$$

/ / transformarea coloanei r a matricii A

- $a_{r+1\ r} = k; a_{ir} = 0, i = r + 2, \ldots, n;$

Algoritmul de obţinere a formei superior Hessenberg (3/3)

$/ \, / \; A = A * P_r$

$/ \, /$ transformarea liniilor $i = 1, \ldots, n$

- for $i = 1, \ldots, n$

$$\begin{cases} \gamma = (\gamma_i / \beta) = ((\mathbf{e}_i^T A)\mathbf{u})/\beta = \left(\displaystyle\sum_{j=r+1}^{n} \mathbf{u}_j a_{ij} \right) / \beta; \\ \text{for } j = r+1, \ldots, n \\ \qquad a_{ij} = a_{ij} - \gamma * \mathbf{u}_j. \end{cases}$$

Algoritmul QR pentru aproximarea valorilor proprii

Algoritmul QR - aproximarea valorilor proprii

Prezentăm în continuare cel mai utilizat algoritm pentru aproximarea valorilor proprii ale unei matrice pătratice oarecare.

Definiţie

Spunem că o matrice $S \in \mathbb{R}^{n\times n}$ este în *formă Schur reală* dacă este în formă Hessenberg superioară și, în plus, este bloc-diagonală:

$$S = \begin{pmatrix} S_{11} & S_{12} & \cdots & S_{1p} \\ 0 & S_{22} & \cdots & S_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & S_{pp} \end{pmatrix}.$$

Algoritmul QR - aproximarea valorilor proprii

Blocurile S_{ii} sunt de două tipuri:

- $S_{ii} \in \mathbb{R}$ - este valorea proprie reală ;
- $S_{ii} \in \mathbb{R}^{2\times 2}$ - este bloc corespunzător valorilor proprii complexe

Valorile proprii corespunzătoare blocului S_{ii}:

$$S_{ii} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

sunt rădăcinile ecuației:

$$\begin{vmatrix} \lambda - a & -b \\ -c & \lambda - d \end{vmatrix} = \lambda^2 - (a+d)\lambda + (ad - bc) = 0$$

Se presupune că această ecuație de gradul 2 are rădăcini complexe.

Algoritmul QR - aproximarea valorilor proprii

Algoritmul QR de aproximare a valorilor proprii construiește un șir de matrice $A^{(k)} \in \mathbb{R}^{n \times n}$, asemenea cu matricea A:

$$A^{(k)} \sim A, \quad \forall k$$

Acest șir converge către o matrice în formă Schur reală:

$$A^{(k)} \longrightarrow S, \quad \text{când } k \to \infty$$

Matricea limită S este asemenea cu A, iar valorile proprii ale lui S sunt ușor de determinat.

Algoritmul QR - aproximarea valorilor proprii

Construcţia şirului

Șirul $A^{(k)}$ se construiește astfel:

$A^{(0)} = A,\ A^{(0)} = Q_0 R_0$ (descompunere QR calc. pentru matricea $A^{(0)}$)

$A^{(1)} := R_0 Q_0,\ A^{(1)} = Q_1 R_1$ (descompunere QR calc. pentru matricea $A^{(1)}$)

$A^{(2)} := R_1 Q_1$

$\vdots$

$A^{(k)} = Q_k R_k$ (descompunere QR calc. pentru matricea $A^{(k)}$)

$A^{(k+1)} := R_k Q_k, \quad k = 0, 1, 2, \ldots$

Matricele Q_k sunt ortogonale ($Q_k^{-1} = Q_k^T$) iar matricele R_k sunt superior triunghiulare.

Algoritmul QR - aproximarea valorilor proprii

Matricele $A^{(k)}$ și $A^{(k+1)}$ sunt asemenea. Într-adevăr, înmulţind la stânga relaţia $A^{(k)} = Q_k R_k$ cu Q_k^T, avem

$$Q_k^T * |A^{(k)} = Q_k R_k \Rightarrow R_k = Q_k^T A.$$

$$A^{(k+1)} = R_k Q_k = Q_k^T A^{(k)} Q_k \Rightarrow A^{(k+1)} \sim A^{(k)}, \forall k.$$

Prin urmare, toate matricele din șirul construit sunt asemenea și au aceleași valori proprii, pe cele ale matricei iniţiale $A = A^{(0)}$:

$$A = A^{(0)} \sim A^{(1)} \sim \cdots \sim A^{(k)} \sim \cdots \sim S$$

Dacă matricea $A^{(k)}$ este în formă Hessenberg superioară, atunci descompunerea QR folosind rotații Givens se simplifică.

Algoritmul QR - aproximarea valorilor proprii

Reamintim algoritmul lui Givens:

$$R_{n-1,n}(\theta_{n-1,n})\cdots R_{pn}(\theta_{pn})\cdots R_{p,p+1}(\theta_{p,p+1})\cdots R_{12}(\theta_{12})A = R$$

Dacă matricea A este în formă Hessenberg superioară, atunci în algoritmul lui Givens, din cele $\dfrac{n(n-1)}{2}$ înmulțiri cu matrici de rotație, rămân doar $n-1$:

$$R_{n-1,n}(\theta_{n-1,n})\cdots R_{p,p+1}(\theta_{p,p+1})\cdots R_{23}(\theta_{23})R_{12}(\theta_{12})A = R$$

Algoritmul QR - aproximarea valorilor proprii

Problema care se pune este dacă, pornind cu o matrice în formă Hessenberg, toate matricele șirului QR rămân în formă Hessenberg:

$$A^{(k)}(\text{în formă Hessenberg}) = H = QR \text{ (cu Givens)} \;\Rightarrow ?$$

Avem

$$A^{(k+1)} = \bar{H} = RQ = Q^T A^{(k)} Q = Q^T H Q$$

şi vrem să arătăm că $A^{(k+1)}$ este tot în formă Hessenberg. Avem

$$\bar{H} = Q^T H Q = R R_{12}^T(\theta_{12}) \cdots R_{r\ r+1}^T(\theta_{r\ r+1}) \cdots R_{n-1\ n}^T(\theta_{n-1\ n})$$

Algoritmul QR - aproximarea valorilor proprii

Notăm:

$$\bar{R} = RR_{12}^T(\theta_{12})$$

pentru care avem:

$$\begin{cases} \bar{r}_{i1} = cr_{i1} + sr_{i2}, \ \forall i \\ \bar{r}_{i2} = -sr_{i1} + cr_{i2}, \ \forall i \end{cases} + \begin{cases} \bar{r}_{i1} = 0, \ \forall i = 2, .., n \\ \bar{r}_{i2} = 0, \ \forall i = 3, .., n \end{cases} \Rightarrow \begin{cases} \bar{r}_{i1} = 0, \ i = 3, .., n \\ \bar{r}_{i2} = 0, \ i = 3, .., n. \end{cases}$$

deci coloana 1 se transformă în formă Hessenberg, iar coloana 2 rămâne în formă superior triunghiulară.

Algoritmul QR - aproximarea valorilor proprii

La pasul p, avem:

$$(RR_{12}^T(\theta_{12})\cdots R_{p-1\ p}^T(\theta_{p-1\ p}))R_{p\ p+1}^T(\theta_{pp+1}) = \tilde{R}R_{p\ p+1}^T(\theta_{pp+1}) = \bar{R},$$

$$\tilde{R} = R \cdot R_{12}^T(\theta_{12})\cdots R_{p-1\ p}^T(\theta_{p-1p}).$$

Matricea R are primele $p-1$ coloane în formă Hessenberg, iar restul coloanelor în formă superior triunghiulară. Se arată că, la acest pas, matricea $\bar{R}$ va avea primele p coloane în formă Hessenberg, iar restul coloanelor în formă superior triunghiulară.

Algoritmul QR - aproximarea valorilor proprii

Operația $\bar{R} := \tilde{R}\, R^T_{pp+1}(\theta_{pp+1})$ presupune doar modificarea elementelor coloanelor p și $p+1$:

$$\begin{cases} \bar{r}_{ip} = c\tilde{r}_{ip} + s\tilde{r}_{i\;p+1}, \forall i \\ \bar{r}_{i\;p+1} = -s\tilde{r}_{ip} + c\tilde{r}_{i\;p+1}, \forall i \end{cases} + \begin{cases} \tilde{r}_{ip} = 0, i = p+1, \ldots, n \\ \tilde{r}_{i\;p+1} = 0, i = p+2, \ldots, n \end{cases} \Rightarrow$$

$$\begin{cases} \bar{r}_{ip} = 0,\ i = p+2, \ldots, n \\ \bar{r}_{i\;p+1} = 0,\ i = p+2, \ldots, n. \end{cases}$$

Algoritmul QR - aproximarea valorilor proprii

Observăm din relațiile de mai sus că în matricea $\bar{R}$, coloana p are formă Hessenberg, iar coloana $p+1$ rămâne în formă superior triunghiulară. Celelalte elemente ale matricei nu se modifică.

Prin urmare, după pasul $n-1$, matricea

$$\bar{H} = A^{(k+1)}$$

este în formă Hessenberg superioară.

Algoritmul QR de aproximare a valorilor proprii, folosind descompunerea Givens, păstrează forma Hessenberg.

Algoritmul QR pentru valori proprii

Algoritm:

/ / Se aduce matricea A la forma Hessenberg

- $A = \bar{Q} A \bar{Q}^T$;
- k=0;
- while ($A \neq$ forma Schur reală)
  - $A = QR$; / / Se calculează cu algoritmul Givens
  - $A = RQ$ sau $Q^T A Q$;
  - $k = k + 1$.

Algoritmul QR - aproximarea valorilor proprii

În practică se presupune că matricea A este în **formă Hessenberg neredusă**, adică:

$$a_{i\ i-1} \neq 0, \forall i = 2, \ldots, n.$$

Dacă matricea nu este în formă neredusă, problema se decuplează

$$\begin{array}{cc} A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} & \begin{array}{l} p \\ n-p \end{array} \\ \begin{array}{cc} \quad\quad\quad p & \quad n-p \end{array} & \end{array}$$

cu $p = n-1$ sau $n-2$.

Algoritmului QR cu deplasare ("shift") simplă

Algoritmul cu deplasare simplă este următorul:

- $A = \bar{Q}A\bar{Q}^T$;/ / aducerea la forma Hessenberg neredusă
- $k = 0$;
- while ($A \neq$ forma Schur reală)
  - $A - d_k I_n = QR$;/ / se calculează cu Givens
  - $A := RQ + d_k I_n$;
  - $k = k + 1$;

  $d_k \in \mathbb{R}$ sunt constantele de deplasare.

Algoritmul QR cu deplasare simplă

Dacă $A - dI_n = QR(A^{(k)})$ și $\bar{A} = RQ + dI_n$ $(A^{(k+1)})$, se pune problema dacă cele două matrice sunt asemenea ($A \sim \bar{A}$), adică dacă matricele din șirul construit prin pasul QR cu deplasare simplă au aceleași valori proprii.

$$\bar{A} = Q^T QRQ + dQ^T Q = Q^T (QR + dI_n) Q = Q^T AQ \Rightarrow \bar{A} \sim A.$$

Varianta cu deplasare se utilizează pentru a accelera convergența algoritmului.

Dacă $\lambda_1, \lambda_2, \ldots, \lambda_n$ sunt valorile proprii ale matricei A, ordonate astfel încât:

$$|\lambda_1 - d| \geq |\lambda_2 - d| \geq \cdots \geq |\lambda_n - d|.$$

Algoritmul QR cu deplasare simplă

Rapiditatea cu care elementele subdiagonale tind la zero $a_{p+1\ p}^{(k)} \to 0, k \to \infty$ este dată de rata de convergenţă a expresiei $\left|\dfrac{\lambda_{p+1} - d}{\lambda_p - d}\right|^k$.

Dacă se alege $d \approx \lambda_n$ convergenţa $a_{n-1n}^{(k)} \to 0$ este rapidă.

Teoremă

Fie d o valoare proprie a unei matrice Hessenberg neredusă H. Dacă $\bar{H} = RQ + dI_n$, cu $H - dI_n = QR$ descompunerea QR a matricei $H - dI_n = QR$. Atunci

$$\bar{h}_{n\ n-1} = 0,\ \bar{h}_{nn} = d.$$

Algoritmul QR cu deplasare simplă

Algoritmul QR cu deplasare simplă găseşte valoarea proprie d într-un singur pas.

Euristic s-a constatat că la fiecare pas, cea mai bună aproximare a unei valori proprii este $a_{nn}^{(k)}$.

$$d_k = a_{nn}^{(k)}.$$

Algoritmul QR cu deplasare simplă

Algoritmul QR cu deplasare simplă

- $A = \bar{Q}A\bar{Q}^T$; / / aducerea la forma Hessenberg neredusă
- $k = 0$;
- while ($A \neq$ forma Schur reală)
  - $A - a_{nn}I_n = QR$; / / se calc. cu alg. Givens
  - $A := RQ + a_{nn}I_n$;
  - $k = k + 1$;

Algoritmul QR cu deplasare ("shift") dublă

În cazul când valorile proprii a_1, a_2 corespunzătoare blocului:

$$G = \begin{pmatrix} a_{pp} & a_{pn} \\ a_{np} & a_{nn} \end{pmatrix}, \ p = n - 1,$$

sunt complexe, $a_1, a_2 \in \mathbb{C}$, abordarea cu deplasare simplă nu mai asigură accelerarea convergenţei.

Avem:

$$\begin{aligned} \det(\lambda I_2 - G) &= (\lambda - a_1)(\lambda - a_2) = (\lambda - a_{pp})(\lambda - a_{nn}) - a_{pn}a_{np} \\ &= \lambda^2 - (a_1 + a_2)\lambda + a_1 a_2 = \lambda^2 - (a_{pp} + a_{nn})\lambda + a_{pp}a_{nn} - a_{pn}a_{np} \end{aligned}$$

$$a_1 + a_2 = a_{pp} + a_{nn} = trace(G), \quad a_1 a_2 = a_{pp}a_{nn} - a_{pn}a_{np} = \det(G).$$

Algoritmul QR cu deplasare ("shift") dublă

Algoritmul QR cu deplasare dublă constă în trecerea de la matricea $A = A^{(k)}$ la matricea $A_2 = A^{(k+1)}$ realizând doi paşi cu deplasare simplă :

$A \longrightarrow A_1$(deplasare simplă a_1), $A_1 \longrightarrow A_2$(deplasare simplă a_2)

$$\begin{aligned} A - a_1 I_n &= Q_1 R_1 \\ A_1 &= R_1 Q_1 + a_1 I_n \\ A_1 - a_2 I_n &= Q_2 R_2 \\ A_2 &= R_2 Q_2 + a_2 I_n. \end{aligned}$$

Algoritmul QR cu deplasare ("shift") dublă

Fie matricea:

$$\begin{aligned}
M := (Q_1Q_2)(R_2R_1) &= Q_1(Q_2R_2)R_1 = Q_1(A_1 - a_2I_n)R_1 \\
&= Q_1(Q_1^TAQ_1 - a_2I_n)R_1 = Q_1Q_1^TAQ_1R_1 - a_2Q_1R_1 \\
&= (A - a_2I_n)Q_1R_1 = (A - a_2I_n)(A - a_1I_n) \\
&= A^2 - (a_1 + a_2)A + a_1a_2I_n.
\end{aligned}$$

Avem următoarele relaţii de asemănare:

$$A \sim A_1 = Q_1^TAQ_1 \sim A_2 = Q_2^TA_1Q_2 = Q_2^TQ_1^TAQ_1Q_2$$

$$A_2 = (Q_1Q_2)^TA(Q_1Q_2) = Q^TAQ,\ Q := Q_1Q_2.$$

Algoritmul QR cu deplasare ("shift") dublă

Matricea Q care asigură trecerea de la matricea A la matricea A_2 este matricea ortogonală din descompunerea QR a matricii $M = (A - a_2 I_n)(A - a_1 I_n)$.

Pasul QR cu deplasare dublă se face urmând etapele:

1. se calculează matricea $M = A^2 - sA + qI_n$ cu
   $s = a_1 + a_2 = a_{pp} + a_{nn}, \quad q = a_1 a_2 = a_{pp}a_{nn} - a_{pn}a_{np}$;
2. se calculează descompunerea QR a matricei M;
3. $A_2 := Q^T A Q$.

Vectori proprii

Considerăm două matrice asemenea A şi B:

$$A \sim B \iff A = PBP^{-1}, P \text{ - matrice nesingulară.}$$

Ştim că cele două matrice au acelaşi polinom caracteristic, $p_A(\lambda) \equiv p_B(\lambda)$, deci au aceleaşi valori proprii.

Ne interesează care este legătura între vectorii proprii asociaţi aceleiași valori proprii.

Fie $\mathbf{u}$ vector propriu asociat valorii proprii λ pentru matricea A şi $\mathbf{w}$ un vector propriu asociat valorii proprii λ pentru matricea B.

Care este relaţia între $\mathbf{u}$ şi $\mathbf{w}$?

Vectori proprii

$$A\mathbf{u} = \lambda\mathbf{u}, B\mathbf{w} = \lambda\mathbf{w}, A = PBP^{-1} \Rightarrow PBP^{-1}\mathbf{u} = \lambda\mathbf{u}$$

$$\Rightarrow BP^{-1}\mathbf{u} = \lambda P^{-1}\mathbf{u} \Rightarrow \mathbf{w} = P^{-1}\mathbf{u},\ \mathbf{u} = P\mathbf{w}.$$

Dacă se aplică algoritmul QR unei matrice simetrice, forma Schur reală la care se ajunge este o matrice diagonală:

$$S = \Lambda = \mathrm{diag}[\lambda_1, \lambda_2, \ldots, \lambda_n].$$

Legătura dintre matricea simetrică iniţială A şi matricea diagonală este de forma:

$$S = \Lambda = \mathrm{diag}[\lambda_1, \lambda_2, \ldots, \lambda_n] = U^T A U$$

cu U o matrice ortogonală, coloanele matricei U fiind vectori proprii asociaţi valorilor proprii $\lambda_1, \lambda_2, \ldots, \lambda_n$ reale.

Algoritmul QR pentru matrice simetrice (valori+vectori proprii)

Matricea U se poate calcula astfel:

/ / se aduce matricea A la forma Hessenberg

- $A = \bar{Q}A\bar{Q}^T$;
- $U = \bar{Q}^T$;
- $k = 0$;
- while ($A \neq$ matrice diagonală)
  - $A = QR$;/ / se calc. cu alg. Givens
  - $A = RQ$ sau Q^TAQ;
  - $U = UQ$;
  - $k = k + 1$.