Valori Proprii (2)

Valentin-Ioan VINTILĂ

Facultatea de Automatică și Calculatoare - CTI
Universitatea POLITEHNICA București

04 aprilie 2023 (Lab. 6)

Cuprins

Coeficientul Rayleigh

Definiție (coeficientul Rayleigh)

Fie o matrice pătratică $A \in \mathbb{R}^{n \times n}$, $n \in \mathbb{N}^*$. Dacă $\boldsymbol{x} \in \mathbb{R}^n$ este un vector asociat matricei A, atunci **coeficientul Rayleigh** se definește drept **scalarul** (funcția $r : \mathbb{R}^n \to \mathbb{R}$):

$$\boxed{r(\boldsymbol{x}) = \frac{\boldsymbol{x}^T A \boldsymbol{x}}{||\boldsymbol{x}||^2} = \left(\frac{\boldsymbol{x}}{||\boldsymbol{x}||}\right)^T A \left(\frac{\boldsymbol{x}}{||\boldsymbol{x}||}\right)}$$

Metoda puterii directe – sinteză

Trecem la algoritm – facem notația: $\boldsymbol{y}^{(k)} \to \boldsymbol{t}$

Algoritmic, MPD se poate scrie astfel:

1. Se aproximează $\boldsymbol{y}'^{(0)}$ (primește o valoare de început);
2. Pentru fiecare pas $k \in \mathbb{N}^*$, procedăm astfel:
   - Se calculează un *vector auxiliar*: $\boldsymbol{t} = A\boldsymbol{y}'^{(k-1)}$;
   - Acest vector normalizat va fi chiar $\boldsymbol{y}'^{(k)}$, adică $\boldsymbol{y}'^{(k)} = \frac{\boldsymbol{t}}{||\boldsymbol{t}||}$;
   - Folosind coeficientul Rayleigh, se poate calcula o nouă aproximare a valorii proprii predominante:

$$\lambda_1^{(k)} = r\left(\boldsymbol{y}'^{(k)}\right) = \frac{\left(\boldsymbol{y}'^{(k)}\right)^T A \left(\boldsymbol{y}'^{(k)}\right)}{\left|\left|\boldsymbol{y}'^{(k)}\right|\right|^2} = \left(\boldsymbol{y}'^{(k)}\right)^T A \left(\boldsymbol{y}'^{(k)}\right)$$

Metoda puterii directe - concluzii

O folosim în practică? NU! *Vom vedea de ce preferăm să utilizăm metoda puterii inverse...*

Metoda puterii inverse (1)

Metoda puterii inverse se referă la trei algoritmi distincți:

- MPI nedeplasată;
- MPI deplasată;
- **MPI deplasată cu coeficient Rayleigh**.

Ne vom referi exclusiv la ultimele două!

Metoda puterii inverse (2)

Teoremă (deplasarea spectrului)

Fie matricea pătratică $A \in \mathbb{R}^{n \times n}$, $n \in \mathbb{N}^*$. Dacă $\mu \in \mathbb{R}$, $\lambda_i \in \lambda(A)$ și $\boldsymbol{x} \in \mathbb{R}^n$ este vectorul propriu asociat, atunci corespondentul perechii $(\lambda_i, \boldsymbol{x})$ pentru A va fi $(\lambda_i - \mu, \boldsymbol{x})$ pentru matricea $(A - \mu I_n)$.

Puteți schița acasă o demonstrație ca temă...

Se poate demonstra matematic faptul că cea mai bună alegere a coeficientului μ este chiar λ_1.

Evident, vom folosi aproximarea sa, $\lambda_1^{(k)}$.

Metoda puterii inverse – sinteză

Algoritmul provine din MPD, deci MPI se poate scrie astfel:

1. Se aproximează $\boldsymbol{y}'^{(0)}$ (primește o valoare de început);
2. Pentru fiecare pas $k \in \mathbb{N}^*$, procedăm astfel:
   - Se calculează un *vector auxiliar*: $\boldsymbol{t} = (A - \lambda_1^{(k-1)} I_n) \cdot \boldsymbol{y}'^{(k-1)}$;
   - Acest vector normalizat va fi chiar $\boldsymbol{y}'^{(k)}$, adică $\boldsymbol{y}'^{(k)} = \frac{\boldsymbol{t}}{||\boldsymbol{t}||}$;
   - Folosind coeficientul Rayleigh, se poate calcula o nouă aproximare a valorii proprii predominante:

$$\lambda_1^{(k)} = \left(\boldsymbol{y}'^{(k)}\right)^T A \left(\boldsymbol{y}'^{(k)}\right)$$

Metoda puterii inverse - concluzii

O folosim în practică? Oarecum! *Putem folosi alți algoritmi pentru a calcula toate valorile proprii!*

Metoda deflației Householder (1)

Ne amintim forma unui reflector Householder $H \in \mathbb{R}^{n \times n}$:

$$H = I_n - 2 \cdot \frac{\boldsymbol{u}\boldsymbol{u}^T}{||\boldsymbol{u}||^2}$$

Să considerăm vectorul $\boldsymbol{u}$ (aproape) ca la QR:

$$\boldsymbol{u} = \boldsymbol{x} \pm ||\boldsymbol{x}||\boldsymbol{e_1}$$

Ne amintim că, aplicând Householder pe $\boldsymbol{x}$, se obține:

$$\boxed{H\boldsymbol{x} = \mp ||\boldsymbol{x}||\boldsymbol{e_1}}$$

Metoda deflației Householder (2)

Să zicem că vrem să calculăm λ_1 pentru $A \in \mathbb{R}^{n \times n}$.

Ne definim întâi o matrice $G \in \mathbb{R}^{n \times n}$, aparent de nicăieri:

$$G = HAH^{-1} = HAH^T = HAH$$

Să considerăm $\boldsymbol{x} \in \mathbb{R}^n$ astfel încât $A\boldsymbol{x} = \lambda_1 \boldsymbol{x}$. Putem prelucra:

$$\begin{aligned} A\boldsymbol{x} = \lambda_1 \boldsymbol{x} &\Rightarrow HA\boldsymbol{x} = \lambda_1 H\boldsymbol{x} \\ &\Rightarrow HAI_n\boldsymbol{x} = \lambda_1 H\boldsymbol{x}, \text{ dar } H^2 = HH^T = H^T H = I_n \\ &\Rightarrow (HAH^T)(H\boldsymbol{x}) = \lambda_1 H\boldsymbol{x} \\ &\Rightarrow G(H\boldsymbol{x}) = \lambda_1 (H\boldsymbol{x}) \end{aligned}$$

Metoda deflației Householder (3)

Am ajuns la relația:

$$G(H\boldsymbol{x}) = \lambda_1 (H\boldsymbol{x})$$

Știm însă că $H\boldsymbol{x} = \alpha \boldsymbol{e_1}$ (am notat $\alpha = \mp ||\boldsymbol{x}||$). Deci:

$$\alpha G\boldsymbol{e_1} = \lambda_1 \alpha \boldsymbol{e_1} \Rightarrow \boxed{G\boldsymbol{e_1} = \lambda_1 \boldsymbol{e_1}}$$

Ce înseamnă asta?

$$G = \begin{bmatrix} \lambda_1 & \boxtimes \in \mathbb{R}^{1 \times (n-1)} \\ 0 \in \mathbb{R}^{(n-1) \times 1} & A' \in \mathbb{R}^{(n-1) \times (n-1)} \end{bmatrix}$$

Metoda deflației Householder – sinteză

Algoritmic, metoda deflației Householder se poate scrie astfel:

1. Se calculează λ_1 și vectorul propriu asociat al matricei A folosind **MPD** sau **MPI**;
2. Se calculează $\boldsymbol{u} = \boldsymbol{x} \pm ||\boldsymbol{x}||\boldsymbol{e_1}$;
3. Se calculează reflectorul Householder $H = I_n - 2 \cdot \frac{\boldsymbol{u}\boldsymbol{u}^T}{||\boldsymbol{u}||^2}$;
4. Se calculează matricea $G = HAH$;
5. Se extrage λ_1 din colțul din stânga sus și **se continuă algoritmul** pe submatricea $G(2:n, 2:n)$ pentru a afla restul valorilor proprii.

Metoda deflației Householder - concluzii

O folosim în practică? Oarecum! *A început să vă placă răspunsul ăsta...*

Metoda este instabilă numeric dacă se vrea calcularea a mai mult de 3-4 valori proprii. Trebuie deci să găsim altceva...

Alte deflații

Mai există și alte metode de deflație.

Nu insistăm, dar vă amintesc:

- **Deflația Wielandt** (se discută la curs, resursele pe internet sunt aproape inexistente);
- Cazul său general, **deflația Hotelling**.

Metode QR - evoluție din MPD (1)

Fie $A \in \mathbb{R}^{n \times n}$ o matrice pătratică, $n \in \mathbb{N}^*$.

Vom nota cu $\boldsymbol{y'}_i^{(k)}$ vectorul propriu (la a k-a iterație) ce este asociat valorii proprii λ_i. Suplimentar, vom considera momentan cunoscut vectorul propriu $\boldsymbol{q_1}$ asociat cu λ_1.

Vrem acum să studiem vectorul $\boldsymbol{y'}_2^{(k)}$.

Întrucât am stabilit deja că $\boldsymbol{y'}_1^{(k)}$ va fi un vector cu norma 1, dorim să îl scriem pe $\boldsymbol{y'}_2^{(k)}$ în aceeași manieră.

Metode QR - evoluție din MPD (2)

Vom simplifica și mai mult lucrurile, cerând:

$$\boxed{\boldsymbol{y}'^{(k)}_1 \perp \boldsymbol{y}'^{(k)}_2}$$

Cum facem asta? Gram-Schmidt – proiecția lui $\boldsymbol{y}'^{(k)}_2$ pe $\boldsymbol{q}_1$ poate fi scrisă sub forma $\left\langle \boldsymbol{q}_1, \boldsymbol{y}'^{(\boldsymbol{k})}_2 \right\rangle \boldsymbol{q}_1$, deci:

$$\boxed{\boldsymbol{y}'^{(k)}_2 \to \boldsymbol{y}'^{(k)}_2 - \left\langle \boldsymbol{q}_1, \boldsymbol{y}'^{(\boldsymbol{k})}_2 \right\rangle \boldsymbol{y}'_1}$$

Metode QR - evoluție din MPD (3)

Am presupus însă cunoscut $\boldsymbol{q}_1$! În practică, am putea rula în paralel MPD pentru $\boldsymbol{y}'^{(k)}_1$, adică $\boldsymbol{q}_1 \approx \boldsymbol{y}'^{(k)}_1$.

Sintetizat, am avea pentru început:

- $\boldsymbol{y}'^{(0)}_1$ și $\boldsymbol{y}'^{(0)}_2$ primesc o aproximare de început;
- Se ortogonalizează $\boldsymbol{y}'^{(0)}_2$ în raport cu $\boldsymbol{y}'^{(0)}_1$, adică:

$$\boldsymbol{y}'^{(0)}_2 \to \boldsymbol{y}'^{(0)}_2 - \left\langle \boldsymbol{q}_1, \boldsymbol{y}'^{(\boldsymbol{0})}_2 \right\rangle \boldsymbol{q}_1$$

- Se normalizează $\boldsymbol{y}'^{(0)}_1$ și $\boldsymbol{y}'^{(0)}_2$;

Metode QR - evoluție din MPD (4)

Apoi, la fiecare pas $k \geq 1$, se va obține o aproximare tot mai bună astfel:

- Se calculează $\boldsymbol{y}'^{(k)}_1$: considerăm vectorul auxiliar $\boldsymbol{t}_1 = A\boldsymbol{y}'^{(k-1)}_1$. Normalizat, ne dă: $\boldsymbol{y}'^{(k)}_1 = \frac{\boldsymbol{t}_1}{\|\boldsymbol{t}_1\|}$;
- Se calculează $\boldsymbol{y}'^{(k)}_2$: considerăm $\boldsymbol{t}_{21} = A\boldsymbol{y}'^{(k-1)}_2$ a cărui componentă de pe $\boldsymbol{y}'^{(k)}_1$ va fi eliminată, adică $\boldsymbol{t}_{22} = \boldsymbol{t}_{21} - \left\langle \boldsymbol{q}_1, \boldsymbol{y}'^{(\boldsymbol{k})}_2 \right\rangle \boldsymbol{q}_1$;
- Prin coeficientul Rayleigh, obținem:

$$\begin{cases} \lambda^{(k)}_1 = \left(\boldsymbol{y}'^{(k)}_1\right)^T A \left(\boldsymbol{y}'^{(k)}_1\right) \\ \lambda^{(k)}_2 = \left(\boldsymbol{y}'^{(k)}_2\right)^T A \left(\boldsymbol{y}'^{(k)}_2\right) \end{cases}$$

Metode QR - evoluție din MPD (5)

De ce am făcut toate astea? Toată discuția anterioară este ca să înțelegem legătura cu QR.

Inițial, am ortonormat $\boldsymbol{y'}_1{}'^{(\boldsymbol{0})}$ și $\boldsymbol{y'}_2{}'^{(\boldsymbol{0})}$. **Echivalent, se poate calcula o matrice ortogonală $\tilde{Y}^{(0)}$ prin QR:**

$$\left(\tilde{Y}^{(0)}, R\right) = QR\left(\left[\boldsymbol{y}'^{(0)}_1 \quad \boldsymbol{y}'^{(0)}_2\right]\right) = QR\left(Y^{(0)}\right)$$

De altfel, în orice pas k, se poate calcula o matrice ortogonală $\tilde{Y}^{(k)}$:

$$\boxed{\left(\tilde{Y}^{(k)}, R\right) = QR\left(A\left[\boldsymbol{y'}_1{}^{(k-1)} \quad \boldsymbol{y'}_2{}^{(k-1)}\right]\right) = QR\left(A\tilde{Y}^{(k-1)}\right)}$$

Metode QR - evoluție din MPD (6)

Vom folosi matricea $\Lambda^{(k)} \in \mathbb{R}^{2\times 2}$ pentru a dispune valorile proprii calculate la a k-a iterație pe diagonala principală. Astfel, putem restrânge calcularea valorilor λ_1, respectiv λ_2, la următorul produs:

$$\begin{bmatrix} \lambda^{(k)}_1 & 0 \\ 0 & \lambda^{(k)}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{y}'^{(k)}_1 \\ \boldsymbol{y}'^{(k)}_2 \end{bmatrix} A \left[\boldsymbol{y}'^{(k)}_1 \quad \boldsymbol{y}'^{(k)}_2\right] \Rightarrow \boxed{\Lambda^{(k)} = \left(\tilde{Y}^{(k)}\right)^T A \left(\tilde{Y}^{(k)}\right)}$$

Acceptăm fără demonstrație că această metodă poate fi generalizată pentru a afla oricâte valori proprii, să zicem $p \in \mathbb{N}^*$.

Metode QR - evoluție din MPD (7)

Sintetizată, discuția anterioară devine:

- Se calculează $\left(\tilde{Y}^{(0)}, R\right) = QR\left(\left[\boldsymbol{y}^{(0)}_1 \quad \boldsymbol{y}^{(0)}_2 \quad \ldots \quad \boldsymbol{y}^{(0)}_p\right]\right)$
- Pentru fiecare pas $k \geq 1$, se va obține o aproximare tot mai bună:
  - Calculăm $\left(\tilde{Y}^{(k)}, R\right) = QR\left(A\tilde{Y}^{(k-1)}\right)$ (se ignoră valoarea R, aceasta nu este utilă);
  - Se aproximează valorile proprii, adică $\Lambda^{(k)} = \left(\tilde{Y}^{(k)}\right)^T A \left(\tilde{Y}^{(k)}\right)$.

Metoda QR nedeplasată

Anterior, am folosit doar matricea Q din factorizarea QR – o risipă totală față de matricea R!

Vom găsi un algoritm ce folosește această matrice R, **un algoritm cu adevărat** fantastic**!**

De data aceasta pornim de la algoritm și facem abia după demonstrația.

Metoda QR nedeplasată – sinteză

Algoritmic, metoda funcționează astfel:

- Pornim de la $\Lambda^{(0)} = A$, respectiv $Y^{(0)} = Q^{(0)} = R^{(0)} = I_n$;
- Pentru fiecare pas $k \geq 1$, se va obține o aproximare tot mai bună:
  - Se calculează: $\left(Q^{(k)}, R^{(k)}\right) = QR\left(\Lambda^{(k-1)}\right)$;
  - Se aproximează **valorile proprii**, adică $\Lambda^{(k)} = R^{(k)}Q^{(k)}$;
  - Se calculează **vectorii proprii**, deci $Y^{(k)} = Y^{(k-1)}Q^{(k)}$.

Metoda QR nedeplasată – explicație (1)

Deoarece $\left(Q^{(k)}, R^{(k)}\right) = QR\left(\Lambda^{(k-1)}\right)$, știm că $Q^{(k)}R^{(k)} = \Lambda^{(k-1)}$.

Echivalent, $R^{(k)} = \left(Q^{(k)}\right)^{-1}\Lambda^{(k-1)}$. Înlocuind în calculul valorilor proprii (adică în $\Lambda^{(k)} = R^{(k)}Q^{(k)}$), se obține că:

$$\Lambda^{(k)} = \left(Q^{(k)}\right)^{-1}\Lambda^{(k-1)}Q^{(k)}$$

Ce e asta? O transformare de asemănare!

$$\boxed{\Lambda^{(k)} \sim \Lambda^{(k-1)}}$$

Metoda QR nedeplasată – explicație (2)

Dacă știm:

$$\Lambda^{(k)} \sim \Lambda^{(k-1)}$$

...este evident că putem ajunge la:

$$\Lambda^{(k)} \sim \Lambda^{(k-1)} \sim \ldots \sim \Lambda^{(0)}$$

Dat fiind că $\Lambda^{(0)} = A$, obținem că:

$$\Lambda^{(k)} \sim A \Rightarrow \boxed{\lambda\left(\Lambda^{(k)}\right) = \lambda(A)}$$

Metoda QR nedeplasată – explicație (3)

Amintim încă o dată formula:

$$\Lambda^{(k)} = \left(Q^{(k)}\right)^{T}\Lambda^{(k-1)}Q^{(k)}$$

Putem obține treptat ceva interesant:

$$\begin{aligned}
\Lambda^{(k)} &= \left(Q^{(k)}\right)^{T}\Lambda^{(k-1)}Q^{(k)} \\
&= \left(Q^{(k)}\right)^{T}\left(Q^{(k-1)}\right)^{T}\Lambda^{(k-2)}Q^{(k-1)}Q^{(k)} \\
&= \ldots \\
&= \left[\left(Q^{(k)}\right)^{T}\ldots\left(Q^{(0)}\right)^{T}\right]\cdot A\cdot\left[Q^{(0)}\ldots Q^{(k)}\right] \\
&= \left(Y^{(k)}\right)^{T}AY^{(k)}
\end{aligned}$$

Metoda QR nedeplasată – explicație (4)

Am mai întâlnit această formulă odată!

Deși incompletă, demonstrația prezentată ar trebui să fie suficientă cât să vă demonstreze că **algoritmul funcționează!**

Metoda QR nedeplasată - concluzii

O folosim în practică? NU! *Este utilă din punct de vedere teoretic, dar are câteva probleme...*

De exemplu, nu converge pe:

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

Puteți demonstra acest lucru ca temă...

Metoda QR deplasată

Cum am îmbunătățit MPD?

Similar, vom folosi niște coeficienți $\mu^{(k)} \in \mathbb{R}$ pentru deplasare!

Metoda QR deplasată – sinteză

Algoritmic, metoda funcționează astfel:

- Pornim de la $\Lambda^{(0)} = A$, respectiv $Y^{(0)} = Q^{(0)} = R^{(0)} = I_n$;
- Pentru fiecare pas $k \geq 1$, se va obține o aproximare tot mai bună:
  - Se calculează cumva $\mu^{(k)}$ – există mai multe strategii, vom vedea imediat câteva dintre ele;
  - Se calculează: $\left(Q^{(k)}, R^{(k)}\right) = QR\left(\Lambda^{(k-1)} - \mu^{(k)}I_n\right)$;
  - Se aproximează **valorile proprii**, adică $\Lambda^{(k)} = R^{(k)}Q^{(k)} + \mu^{(k)}I_n$;
  - Se calculează **vectorii proprii**, deci $Y^{(k)} = Y^{(k-1)}Q^{(k)}$.

Metoda QR deplasată – alegerea lui μ

Putem calcula $\mu \in \mathbb{R}$ în următoarele moduri:

- **Coeficient Rayleigh:**

$$\mu^{(k)} = r\left(\boldsymbol{A}_n^{(k)}\right) = \ldots = A_{nn}^{(k)}$$

- **Shiftare Wilkinson:** să notăm cu B matricea 2×2 aflată în colțul din dreapta jos al matricei $A^{(k)}$:

$$B = \begin{bmatrix} a_{m-1} & b_{m-1} \\ b_{m-1} & a_m \end{bmatrix}$$

  Atunci $\mu^{(k)} = a_m - \operatorname{sign}(\delta)\cdot\frac{b_{m-1}^2}{|\delta| + \sqrt{\delta^2 + b_{m-1}^2}}$, unde $\delta = \frac{a_{m-1} - a_m}{2}$

Metoda QR deplasată - concluzii

O folosim în practică? DA! *Este o bijuterie numerică!*

Bibliografie

Toate resursele bibliografice de care aveți nevoie se găsesc în descrierea cu care a venit atașată această prezentare

Metode Numerice Matriceale – concluzii

Am încheiat ultimul capitol din metodele numerice matriceale!

Propun un al doilea Kahoot care să treacă prin toate noțiunile!

...dar de data aceasta cu premii!

Metode Numerice Matriceale – feedback

Acum, o să vă rog pe toți cei prezenți să completați un formular de feedback **diferit**, dedicat metodelor numerice matriceale!

Sfârșit

Mulțumesc frumos pentru atenție!

Vă rog frumos să **completați formularul de feedback**!