6. Simetrični problem lastnih vrednosti Relativno robustne reprezentacije

Bor Plestenjak

NLA

6. april 2010

Motivacija

Imamo simetrično matriko A, za katero bi radi izračunali vse lastne pare.

Računanje običajno poteka v treh fazah:

1) Redukcija A na tridiagonalno matriko T.
2) Računanje lastnih parov T.
3) Transformacija lastnih vektorjev T v lastne vektorje A.

Časovne zahtevnosti:

1) $\mathcal{O}(n^3)$
2) Odvisno od metode:
    - Vse preko QR: $\mathcal{O}(n^3)$
    - Lastne vrednosti preko QR, lastni vektorji preko inverzne iteracije: $\mathcal{O}(n^2)$, kadar je potrebna naknadna ortogonalizacija $\mathcal{O}(n^3)$.
    - Deli in vladaj: $\mathcal{O}(n^3)$, če so lastne vrednosti enakomerno razporejene
3) $\mathcal{O}(n^3)$

Izračun lastnega vektorja

$$T = \begin{bmatrix} a_1 & b_1 & & \\ b_1 & \ddots & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ & & b_{n-1} & a_n \end{bmatrix}.$$

Naj bo λ lastna vrednost ireducibilne simetrične tridiagonalne matrike T. Za lastni vektor z velja $z_1 \neq 0$ in $z_n \neq 0$.

$z_1 = 1$
$z_2 = -(a_1 - \lambda)/b_1$
$k = 2, \ldots, n-1$
$\quad z_{k+1} = -(b_{k-1} z_{k-1} + (a_k - \lambda) z_k)/b_k$

Če fiksiramo $z_1 = 1$, potem iz prve do predzadnje enačbe sistema $(T - \lambda I)z = 0$ dobimo lastni vektor z. Pri tem smo opustili zadnjo enačbo.

Če je $\widetilde{\lambda}$ približek za λ, potem za z velja $(T - \widetilde{\lambda} I)z = \delta_n e_n$ za nek δ_n.

Numerično računanje lastnega vektorja

Podobno, če bi opustili k-to enačbo, bi za ostanek veljalo $(T-\widetilde{\lambda}I)z=\delta_k e_k$.

Opustitev k-te enačbe je ekvivalentna temu, da kot približek za lastni vektor vzamemo $z=(T-\widetilde{\lambda}I)^{-1}e_k$.

Naj bo $\widetilde{\lambda}$ približek za λ_j. Katero enačbo naj opustimo?

Iz $e_k=\sum_{i=1}^n (x_i)_k x_i$ sledi

$$z^{(k)}=(T-\widetilde{\lambda}I)^{-1}e_k=\frac{(x_j)_k}{\lambda_j-\widetilde{\lambda}}\left(x_j+\sum_{i\neq j}\frac{(x_i)_k}{(x_j)_k}\cdot\frac{\lambda_j-\widetilde{\lambda}}{\lambda_i-\widetilde{\lambda}}x_i\right)$$

Vidimo, da bo z dober približek za lastni vektor, če

- opustimo k-to enačbo, kjer ima lastni vektor x_j maksimalno komponento,
- velja $|\lambda_j-\widetilde{\lambda}|\ll|\lambda_i-\widetilde{\lambda}|$ za $i\neq j$.

LDL^T in UDU^T faktorizaciji

$$T = \begin{bmatrix} a_1 & b_1 & & \\ b_1 & \ddots & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ & & b_{n-1} & a_n \end{bmatrix} = LD^+L^T = UD^-U^T$$

$$L = \begin{bmatrix} 1 & & & \\ l_1 & 1 & & \\ & \ddots & \ddots & \\ & & l_{n-1} & 1 \end{bmatrix}, \quad D^+ = \begin{bmatrix} d_1^+ & & & \\ & d_2^+ & & \\ & & \ddots & \\ & & & d_n^+ \end{bmatrix}$$

LD^+L^T

$d_1^+ = a_1$
$k = 1, \ldots, n-1$
$\quad l_k = b_k / d_k^+$
$\quad d_{k+1}^+ = a_{k+1} - b_k^2 / d_k^+$

$$U = \begin{bmatrix} 1 & u_1 & & \\ & \ddots & \ddots & \\ & & 1 & u_{n-1} \\ & & & 1 \end{bmatrix}, \quad D^- = \begin{bmatrix} d_1^- & & & \\ & d_2^- & & \\ & & \ddots & \\ & & & d_n^- \end{bmatrix}$$

UD^-U^T

$d_n^- = a_n$
$k = n-1, \ldots, 1$
$\quad u_k = b_k / d_{k+1}^-$
$\quad d_k^- = a_k - b_k^2 / d_{k+1}^-$

Zasukan razcep

Na k zgornjih vrsticah naredimo razcep LD^+L^T, na spodnjih $n-k+1$ pa UD^-U^T. Dobimo $T = N_k D_k N_k^T$, kjer je

$$N_k = \begin{bmatrix} 1 & & & & & & \\ l_1 & \ddots & & & & & \\ & \ddots & 1 & & & & \\ & & l_{k-1} & 1 & u_k & & \\ & & & & 1 & \ddots & \\ & & & & & \ddots & u_{n-1} \\ & & & & & & 1 \end{bmatrix}$$

in $D_k = \mathrm{diag}(d_1^+, \ldots, d_{k-1}^+, \gamma_k, d_{k+1}^-, \ldots, d_n^-)$.

Izrek

1) $\gamma_k = d_k^+ + d_k^- - a_k$
2) *Če je matrika T nesingularna, potem velja* $1/\gamma_k = e_k^T T^{-1} e_k$.

Lastnosti zasukanega razcepa

Lema

Naj bo $N_k D_k N_k^T$ zasukani razcep matrike $T - \widetilde{\lambda} I$. Velja

$$\frac{1}{\gamma_j} = \frac{[(x_j)_k]^2}{\lambda_j - \widetilde{\lambda}} + \sum_{i \neq j} \frac{[(x_i)_k]^2}{\lambda_i - \widetilde{\lambda}}.$$

Naj bo $|\lambda_j - \widetilde{\lambda}| \ll |\lambda_i - \widetilde{\lambda}|$ za $i \neq j$. Potem:

$$\text{majhna vrednost } |\gamma_k| \Leftrightarrow \text{nadpovprečna vrednost } |(x_j)_k|$$

Iz zasukanega razcepa ocenimo, kateri element lastnega vektorja je maksimalen.

Izračun lastnega vektorja iz zasukanega razcepa

$$N_k = \begin{bmatrix} 1 & & & & & & \\ l_1 & \ddots & & & & & \\ & \ddots & 1 & & & & \\ & & l_{k-1} & 1 & u_k & & \\ & & & & 1 & \ddots & \\ & & & & & \ddots & u_{n-1} \\ & & & & & & 1 \end{bmatrix},$$

$D_k = \mathrm{diag}(d_1^+, \ldots, d_{k-1}^+, \gamma_k, d_{k+1}^-, \ldots, d_n^-)$

Če je $T - \widetilde{\lambda} I = N_k D_k N_k^T$, ima sistem $(T - \widetilde{\lambda} I)z = \gamma_k e_k$ enolično rešitev, ki jo dobimo z:

$z_k = 1$
$j = k-1, \ldots, 1$
$\quad z_j = -l_j z_{j+1}$ (v primeru $z_{j+1} = 0$ pa $z_j = -b_{j+1} z_{j+2} / b_j$)
$j = k+1, \ldots, n$
$\quad z_j = -u_{j-1} z_{j-1}$ (v primeru $z_{j-1} = 0$ pa $z_j = -b_{j-2} z_{j-2} / b_{j-1}$)

Ponovitev algoritma

Naj bo $\widetilde{\lambda}$ približek za lastno vrednost matrike T. Lastni vektor z izračunamo po naslednjem postopku.

Algoritem za izračun lastnega vektorja

1) Izračunaj razcepa $T - \widetilde{\lambda} I = LD^+L^T$ in $T - \widetilde{\lambda} I = UD^-U^T$
2) Izračunaj $\gamma_1, \ldots, \gamma_n$ za zasukane razcepe in poišči minimalni $|\gamma_k|$
3) Reši sistem $(T - \widetilde{\lambda} I)z = \gamma_k e_k$ in normiraj z

Zahtevnost tega algoritma je $12n + \mathcal{O}(1)$ operacij.

Če lahko vnaprej ocenimo k (npr. z Gerschgorinovimi krogi), se zahtevnost še zmanjša.

Relativno robustna reprezentacija matrike

Tridiagonalno matriko T lahko predstavimo z vektorjema diagonalnih in obdiagonalnih elementov a in b. Majhne relativne spremembe a in b lahko povzročijo velike relativne spremembe lastnih vrednosti in vektorjev.

Če je T pozitivno definitna, jo lahko predstavimo z vektorjema diagonalnih in obdiagonalnih elementov faktorja Choleskega. Majhne relativne spremembe teh elementov povzročijo majhne relativne spremembe lastnih parov, zato je takšna predstavitev relativno robustna.

Definicija

Množica števil $\{p_i\}$, ki določa matriko T, je relativno robustna reprezentacija (RRR), če se lastni pari matrike $T + \delta T$, ki jo določa množica zmotenih elementov $\{p_i(1+\epsilon_i)\}$, relativno malo razlikujejo od lastnih parov matrike T.

Če reprezentacija z visoko relativno natančnostjo določa le lastne vrednosti $(\lambda_j, \ldots, \lambda_k)$, pravimo da je delna RRR$(j, \ldots, k)$.

Robustnost razcepa LDL^T

Če je T pozitivno definitna, lahko namesto razcepa Choleskega uporabimo LDL^T razcep (izognemo se kvadratnim korenom), ki je prav tako RRR.

Če je T nedefinitna, lahko uporabimo razcep $T - \sigma I = LDL^T$ za izbrani premik σ.

Ob primerno izbranem σ dosežemo, da je zgornji razcep delna RRR za iskane lastne vrednosti.

Medtem ko lastne vrednosti lahko izračunamo z visoko relativno natančnostjo npr. z bisekcijo, pa za natančne lastne vektorje potrebujemo, da je lastna vrednost relativno dobro izolirana, kar lahko dosežemo s premiki.

RRR premaknjene matrike

Denimo, da poznamo RRR matrike $T + \tau I$ oblike LDL^T.

Potrebujemo $L^+D^+{L^+}^T$ in $U^-D^-{U^-}^T$ razcepa premaknjene matrike $T + \tau I - \mu I = LDL^T - \mu I$.

Z naslednjimi algoritmi ohranimo relativno robustno reprezentacijo.

dstqds algoritem

$$\begin{aligned}
&s_1 = -\mu\\
&i = 1, \dots, k-1\\
&\quad d_i^+ = s_i + d_i\\
&\quad l_i^+ = l_i d_i / d_i^+\\
&\quad s_{i+1} = l_i^+ l_i s_i - \mu\\
&d_n^+ = s_n + d_n
\end{aligned}$$

dqds algoritem

$$\begin{aligned}
&p_n = d_n - \mu\\
&i = n-1, \dots, 1\\
&\quad d_{i+1}^- = d_i l_i^2 + p_{i+1}\\
&\quad t = d_i / d_{i+1}^-\\
&\quad p_i = p_{i+1} i - \mu\\
&d_1^- = p_1
\end{aligned}$$

izračun γ

$$\begin{aligned}
&i = 1, \dots, n\\
&\quad \gamma_i = s_i + p_i + \mu
\end{aligned}$$

GetVec

Naj bo $\widetilde{\lambda}$ približek za lastno vrednost matrike LDL^T, ki je RRR.

Algoritem GetVec

1) Preko dstqds izračunaj razcep $LDL^T - \widetilde{\lambda} I = L^+ D^+ {L^+}^T$
2) Preko dqds izračunaj razcep $LDL^T - \widetilde{\lambda} I = U^- D^- {U^-}^T$
3) Izračunaj $\gamma_1, \ldots, \gamma_n$ za zasukane razcepe in poišči minimalni $|\gamma_k|$
4) Reši sistem $N_k D_k N_k^T z = \gamma_k e_k$ in normiraj z

Natančnost

Naj bo $\widetilde{\lambda}$ približek za lastno vrednost λ_j matrike $T = LDL^T$.

Za lastni vektor izračunan preko inverzne iteracije, velja

$$|\sin \angle(z, x_j)| \leq \frac{\mathcal{O}(nu\|T\|)}{\text{gap}(\widetilde{\lambda})},$$

kjer je $\text{gap}(\widetilde{\lambda}) = \min_{i \neq j} |\widetilde{\lambda} - \lambda_i|$,
Za lastni vektor, izračunan preko RRR, pa velja

$$|\sin \angle(z, x_j)| \leq \frac{\mathcal{O}(nu)}{\text{relgap}(\widetilde{\lambda})},$$

kjer je $\text{relgap}(\widetilde{\lambda}) = \dfrac{\text{gap}(\widetilde{\lambda})}{|\widetilde{\lambda}|}$.

Kadar je $|\widetilde{\lambda}|$ dosti manjša kot $\|T\|$, je RRR lahko veliko natančnejši.

Končni algoritem (MRRR)

1) Poišči τ da bo LDL^T razcep matrike $T + \tau I$ RRR.
2) Izračunaj $T + \tau I = L_0 D_0 L_0^T$
3) Relativno natančno izračunaj lastne vrednosti $\widetilde{\lambda}_1, \ldots, \widetilde{\lambda}_n$ matrike $L_0 D_0 L_0^T$.
4) $l = 1$, $m = n$
5) Razdeli lastne vrednosti $\widetilde{\lambda}_l, \ldots, \widetilde{\lambda}_m$ na izolirane in gruče.
6) Za vsako izolirano lastno vrednost $\widetilde{\lambda}_j$ izračunaj z_j preko GetVec.
7) Za vsako gručo $\widetilde{\lambda}_j, \ldots, \widetilde{\lambda}_{j+k-1}$
   - Preko dstqds izračunaj $L_0 D_0 L_0^T - \tau_s I = L_S D_S L_S^T$, kjer τ_s izbereš tako, da
     - bo v $L_S D_S L_S^T$ vsaj ena lastna vrednost izolirana,
     - bo $L_S D_S L_S^T$ delna $RRR(j, \ldots, j + k - 1)$.
   - Relativno natančno izračunaj lastne vrednosti $\mu_j, \ldots, \mu_{j+k-1}$ matrike $L_s D_s L_s^T$.
   - Določi $\widetilde{\lambda}_i = \mu_i$ za $i = j, \ldots, j + k - 1$
   - $l = j$, $m = j + k - 1$, $L_0 = L_s$, $D_0 = D_s$, rekurzivno se vrni na korak 5).

Kriterij za izolirano lastno vrednost oz. gručo

Za lastno vrednost $\widetilde{\lambda}_j$ pravimo, da je relativno izolirana, če velja

$$\mathrm{relgap}(\widetilde{\lambda}_j) \geq \delta.$$

Lastne vrednosti $\widetilde{\lambda}_j, \ldots, \widetilde{\lambda}_{j+k-1}$ tvorijo gručo, če velja

$$\begin{aligned} \mathrm{reldis}(\widetilde{\lambda}_i, \widetilde{\lambda}_{i+1}) &< \delta \quad \text{za} \quad i = j, \ldots, j+k-2 \\ \mathrm{reldis}(\widetilde{\lambda}_{j-1}, \widetilde{\lambda}_j) &\geq \delta \\ \mathrm{reldis}(\widetilde{\lambda}_{j+k-1}, \widetilde{\lambda}_{j+k}) &\geq \delta, \end{aligned}$$

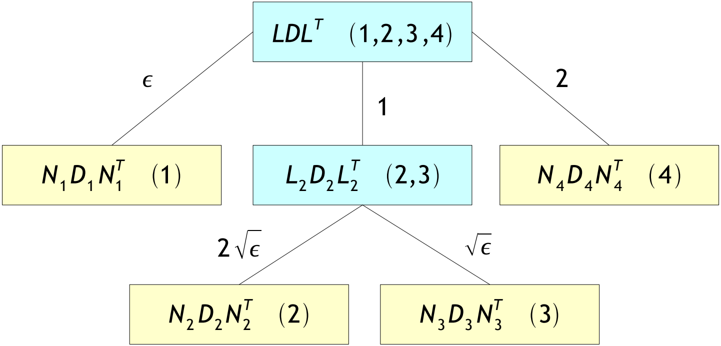
kjer je

$$\mathrm{reldis}(\lambda, \mu) = \frac{|\lambda - \mu|}{|\lambda|}.$$

Za δ vzamemo npr. 10^{-3}.

Predstavitveno drevo

Matrika ima lastne vrednosti $\epsilon, 1+\sqrt{\epsilon}, 1+2\sqrt{\epsilon}, 2$.



- Listi predstavljajo izračune lastnih vektorjev (GetVec).
- Povezave predstavljajo prehod na delno RRR premaknjene matrike (dstqds).

Ostalo

- Avtor RRR algoritma je Inderjit S. Dhillon (1997)
- Podrobnosti sta v člankih objavila Dhillon in Parlett (2000, 2003, 2004)
- Zahtevnost metode za izračun k lastnih parov je $\mathcal{O}(kn)$.
- Razvili so tudi varianto za singularni razcep bidiagonalne matrike
- V LAPACKU je prisoten od verzije 3.0 dalje (1999)
- Predvideva se, da bo sčasoma postal osnovna metoda za reševanje simetričnega problema lastnih vrednosti
- Metoda je večinoma hitrejša od ostalih in potrebuje najmanj dodatnega spomina