12. Računanje singularnega razcepa

Bor Plestenjak

NLA

18. maj 2010

Uvod

Za matriko $A \in \mathbb{R}^{m \times n}$ bi radi izračunali singularni razcep $A = U\Sigma V^T$.

Ker velja

$$A^T A = V\Sigma^T \Sigma V^T,$$

dobimo preprosto idejo:

Uvod

Za matriko $A \in \mathbb{R}^{m \times n}$ bi radi izračunali singularni razcep $A = U\Sigma V^T$.

Ker velja

$$A^T A = V\Sigma^T \Sigma V^T,$$

dobimo preprosto idejo:

1) izračunamo $A^T A$,

Uvod

Za matriko $A \in \mathbb{R}^{m \times n}$ bi radi izračunali singularni razcep $A = U\Sigma V^T$.

Ker velja

$$A^T A = V\Sigma^T \Sigma V^T,$$

dobimo preprosto idejo:

1) izračunamo $A^T A$,
2) rešimo lastni problem $A^T A = V \Lambda V^T$, odtod dobimo V,

Uvod

Za matriko $A \in \mathbb{R}^{m\times n}$ bi radi izračunali singularni razcep $A = U\Sigma V^T$.

Ker velja

$$A^T A = V\Sigma^T\Sigma V^T,$$

dobimo preprosto idejo:

1) izračunamo $A^T A$,
2) rešimo lastni problem $A^T A = V\Lambda V^T$, odtod dobimo V,
3) za Σ vzamemo $m \times n$ matriko, ki ima v zgornjem bloku kvadratni koren Λ,

Uvod

Za matriko $A \in \mathbb{R}^{m \times n}$ bi radi izračunali singularni razcep $A = U\Sigma V^T$.

Ker velja

$$A^T A = V\Sigma^T \Sigma V^T,$$

dobimo preprosto idejo:

1) izračunamo $A^T A$,
2) rešimo lastni problem $A^T A = V\Lambda V^T$, odtod dobimo V,
3) za Σ vzamemo $m \times n$ matriko, ki ima v zgornjem bloku kvadratni koren Λ,
4) rešimo sistem $U\Sigma = AV$ za matriko U.

Računanje preko $A^T A$ ni numerično stabilno

Če vzamemo $A = \begin{bmatrix} 1 & 1 \\ 0 & 10^{-8} \end{bmatrix}$, potem je $A^T A = \begin{bmatrix} 1 & 1 \\ 1 & 1 + 10^{-16} \end{bmatrix}$, kar se v dvojni natančnosti zaokroži v $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$.

Računanje preko $A^T A$ ni numerično stabilno

Če vzamemo $A = \begin{bmatrix} 1 & 1 \\ 0 & 10^{-8} \end{bmatrix}$, potem je $A^T A = \begin{bmatrix} 1 & 1 \\ 1 & 1+10^{-16} \end{bmatrix}$, kar se v dvojni natančnosti zaokroži v $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$. Numerično izračunani singularni vrednosti sta

$$\begin{aligned} \widehat{\sigma}_1 &= \sqrt{2} \\ \widehat{\sigma}_2 &= 0, \end{aligned}$$

točni pa

$$\begin{aligned} \sigma_1 &= \sqrt{2} - 2.22 \cdot 10^{-16} \\ \sigma_2 &= 7.07 \cdot 10^{-9}. \end{aligned}$$

Pri majhnih singularnih vrednostih lahko torej pride do velike relativne napake.

Računanje preko $A^T A$ ni numerično stabilno

Če vzamemo $A = \begin{bmatrix} 1 & 1 \\ 0 & 10^{-8} \end{bmatrix}$, potem je $A^T A = \begin{bmatrix} 1 & 1 \\ 1 & 1 + 10^{-16} \end{bmatrix}$, kar se v dvojni natančnosti zaokroži v $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$. Numerično izračunani singularni vrednosti sta

$$\begin{aligned} \widehat{\sigma}_1 &= \sqrt{2} \\ \widehat{\sigma}_2 &= 0, \end{aligned}$$

točni pa

$$\begin{aligned} \sigma_1 &= \sqrt{2} - 2.22 \cdot 10^{-16} \\ \sigma_2 &= 7.07 \cdot 10^{-9}. \end{aligned}$$

Pri majhnih singularnih vrednostih lahko torej pride do velike relativne napake.

Računanje preko eksplicitno izračunane matrike $A^T A$ ni numerično stabilno.

Povezave z lastnimi vrednostmi

Vemo, da so lastne vrednosti $A^T A$ enake $\sigma_1^2, \ldots, \sigma_n^2$.

Lema

Naj bo A matrika $m \times n$, $m \geq n$, s singularnimi vrednostmi in vektorji $Av_i = \sigma_i u_i$, $i = 1, \ldots, n$. Lastne vrednosti matrike

$$C = \begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix}$$

so $\pm\sigma_i$ in $m - n$ ničel, ustrezni lastni vektorji pa so $\frac{1}{\sqrt{2}} \begin{bmatrix} v_i \\ \pm u_i \end{bmatrix}$ za $i = 1, \ldots, n$ in $\begin{bmatrix} 0 \\ u_k \end{bmatrix}$ za $k = n+1, \ldots, m$.

Posplošitev minimaks izreka in teorija motenj

Iz povezav med singularnim razcepom A in lastnim razcepom $A^T A$ oz. $\begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix}$ sledijo naslednje lastnosti.

Izrek

Naj bo A realna matrika $m \times n$, kjer je $m \geq n$. Za $k = 1, \ldots, n$ velja

$$\sigma_k = \max_{\substack{\dim(S)=k \\ \dim(T)=k}} \min_{\substack{0 \neq x \in S \\ 0 \neq y \in T}} \frac{y^T A x}{\|x\|_2 \|y\|_2} = \max_{\dim(S)=k} \min_{0 \neq x \in S} \frac{\|Ax\|_2}{\|x\|_2}.$$

Posledica

Za singularne vrednosti $A + E$ za $k = 1, \ldots, n$ velja

$$|\sigma_k(A + E) - \sigma_k(A)| \leq \|E\|_2.$$

Računanje preko $A^T A$ ni obratno stabilno

Od obratno stabilnega algoritma za singularne vrednosti A pričakujemo, da bo za izračunane približke veljalo $\widetilde{\sigma}_k = \sigma_k(A+E)$, kjer je $\|E\| = \mathcal{O}(u\|A\|)$, saj potem sledi $|\widetilde{\sigma}_k - \sigma_k| = \mathcal{O}(u\|A\|)$.

Računanje preko $A^T A$ ni obratno stabilno

Od obratno stabilnega algoritma za singularne vrednosti A pričakujemo, da bo za izračunane približke veljalo $\widetilde{\sigma}_k = \sigma_k(A + E)$, kjer je $\|E\| = \mathcal{O}(u\|A\|)$, saj potem sledi $|\widetilde{\sigma}_k - \sigma_k| = \mathcal{O}(u\|A\|)$.

Če računamo preko $A^T A$ z obratno stabilnim algoritmom za računanje lastnih vrednosti simetrične matrike, potem velja

$$|\widetilde{\sigma}_k^2 - \sigma_k^2| = \mathcal{O}(u\|A^T A\|) = \mathcal{O}(u\|A\|^2).$$

Računanje preko $A^T A$ ni obratno stabilno

Od obratno stabilnega algoritma za singularne vrednosti A pričakujemo, da bo za izračunane približke veljalo $\widetilde{\sigma}_k = \sigma_k(A+E)$, kjer je $\|E\| = \mathcal{O}(u\|A\|)$, saj potem sledi $|\widetilde{\sigma}_k - \sigma_k| = \mathcal{O}(u\|A\|)$.

Če računamo preko $A^T A$ z obratno stabilnim algoritmom za računanje lastnih vrednosti simetrične matrike, potem velja

$$|\widetilde{\sigma}_k^2 - \sigma_k^2| = \mathcal{O}(u\|A^T A\|) = \mathcal{O}(u\|A\|^2).$$

Po korenjenju dobimo

$$|\widetilde{\sigma}_k - \sigma_k| = \mathcal{O}\left(u\|A\|^2 \frac{1}{\sigma_k}\right).$$

Računanje preko $A^T A$ ni obratno stabilno

Od obratno stabilnega algoritma za singularne vrednosti A pričakujemo, da bo za izračunane približke veljalo $\widetilde{\sigma}_k = \sigma_k(A + E)$, kjer je $\|E\| = \mathcal{O}(u\|A\|)$, saj potem sledi $|\widetilde{\sigma}_k - \sigma_k| = \mathcal{O}(u\|A\|)$.

Če računamo preko $A^T A$ z obratno stabilnim algoritmom za računanje lastnih vrednosti simetrične matrike, potem velja

$$|\widetilde{\sigma}_k^2 - \sigma_k^2| = \mathcal{O}(u\|A^T A\|) = \mathcal{O}(u\|A\|^2).$$

Po korenjenju dobimo

$$|\widetilde{\sigma}_k - \sigma_k| = \mathcal{O}\left(u\|A\|^2 \frac{1}{\sigma_k}\right).$$

Za večje singularne vrednosti je to v redu, za manjše pa ne. Tako lahko pri σ_n pričakujemo napako velikosti $\mathcal{O}(u\|A\|\kappa_2(A))$.

Redukcija na bidiagonalno obliko

Pri računanju SVD za A (razen pri Jacobijevi metodi) matriko najprej reduciramo na bidiagonalno obliko. Postopek je:

1) poišči ortogonalni matriki U_1, V_1, da bo $A = U_1 B V_1^T$ in B bidiagonalna,
2) izračunaj SVD za B: $B = U_2 \Sigma V_2^T$,
3) SVD razcep za A je $A = (U_1 U_2)\Sigma(V_1 V_2)^T$.

Redukcija na bidiagonalno obliko

Pri računanju SVD za A (razen pri Jacobijevi metodi) matriko najprej reduciramo na bidiagonalno obliko. Postopek je:

1) poišči ortogonalni matriki U_1, V_1, da bo $A = U_1 B V_1^T$ in B bidiagonalna,
2) izračunaj SVD za B: $B = U_2 \Sigma V_2^T$,
3) SVD razcep za A je $A = (U_1 U_2) \Sigma (V_1 V_2)^T$.

Redukcijo na bidiagonalno obliko izvedemo s Householderjevimi zrcaljenji:

$$A \quad = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix}$$

Redukcija na bidiagonalno obliko

Pri računanju SVD za A (razen pri Jacobijevi metodi) matriko najprej reduciramo na bidiagonalno obliko. Postopek je:

1) poišči ortogonalni matriki U_1, V_1, da bo $A = U_1 B V_1^T$ in B bidiagonalna,
2) izračunaj SVD za B: $B = U_2 \Sigma V_2^T$,
3) SVD razcep za A je $A = (U_1 U_2)\Sigma(V_1 V_2)^T$.

Redukcijo na bidiagonalno obliko izvedemo s Householderjevimi zrcaljenji:

$$P_1 A = \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}$$

Redukcija na bidiagonalno obliko

Pri računanju SVD za A (razen pri Jacobijevi metodi) matriko najprej reduciramo na bidiagonalno obliko. Postopek je:

1) poišči ortogonalni matriki U_1, V_1, da bo $A = U_1 B V_1^T$ in B bidiagonalna,
2) izračunaj SVD za B: $B = U_2 \Sigma V_2^T$,
3) SVD razcep za A je $A = (U_1 U_2)\Sigma(V_1 V_2)^T$.

Redukcijo na bidiagonalno obliko izvedemo s Householderjevimi zrcaljenji:

$$P_1 A \widetilde{P}_1 \quad = \begin{bmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}$$

Redukcija na bidiagonalno obliko

Pri računanju SVD za A (razen pri Jacobijevi metodi) matriko najprej reduciramo na bidiagonalno obliko. Postopek je:

1) poišči ortogonalni matriki U_1, V_1, da bo $A = U_1 B V_1^T$ in B bidiagonalna,
2) izračunaj SVD za B: $B = U_2 \Sigma V_2^T$,
3) SVD razcep za A je $A = (U_1 U_2)\Sigma(V_1 V_2)^T$.

Redukcijo na bidiagonalno obliko izvedemo s Householderjevimi zrcaljenji:

$$P_2 P_1 A \widetilde{P}_1 \quad = \begin{bmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix}$$

Redukcija na bidiagonalno obliko

Pri računanju SVD za A (razen pri Jacobijevi metodi) matriko najprej reduciramo na bidiagonalno obliko. Postopek je:

1) poišči ortogonalni matriki U_1, V_1, da bo $A = U_1 B V_1^T$ in B bidiagonalna,
2) izračunaj SVD za B: $B = U_2 \Sigma V_2^T$,
3) SVD razcep za A je $A = (U_1 U_2)\Sigma(V_1 V_2)^T$.

Redukcijo na bidiagonalno obliko izvedemo s Householderjevimi zrcaljenji:

$$P_2 P_1 A \widetilde{P}_1 \widetilde{P}_1 = \begin{bmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & 0 \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix}$$

Redukcija na bidiagonalno obliko

Pri računanju SVD za A (razen pri Jacobijevi metodi) matriko najprej reduciramo na bidiagonalno obliko. Postopek je:

1) poišči ortogonalni matriki U_1, V_1, da bo $A = U_1 B V_1^T$ in B bidiagonalna,
2) izračunaj SVD za B: $B = U_2 \Sigma V_2^T$,
3) SVD razcep za A je $A = (U_1 U_2)\Sigma(V_1 V_2)^T$.

Redukcijo na bidiagonalno obliko izvedemo s Householderjevimi zrcaljenji:

$$P_3 P_2 P_1 A \widetilde{P}_1 \widetilde{P}_1 = \begin{bmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & 0 \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \\ 0 & 0 & 0 & \times \\ 0 & 0 & 0 & \times \end{bmatrix}$$

Redukcija na bidiagonalno obliko

Pri računanju SVD za A (razen pri Jacobijevi metodi) matriko najprej reduciramo na bidiagonalno obliko. Postopek je:

1) poišči ortogonalni matriki U_1, V_1, da bo $A = U_1 B V_1^T$ in B bidiagonalna,
2) izračunaj SVD za B: $B = U_2 \Sigma V_2^T$,
3) SVD razcep za A je $A = (U_1 U_2)\Sigma(V_1 V_2)^T$.

Redukcijo na bidiagonalno obliko izvedemo s Householderjevimi zrcaljenji:

$$P_4 P_3 P_2 P_1 A \widetilde{P}_1 \widetilde{P}_1 = \begin{bmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & 0 \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Redukcija na bidiagonalno obliko

Pri računanju SVD za A (razen pri Jacobijevi metodi) matriko najprej reduciramo na bidiagonalno obliko. Postopek je:

1) poišči ortogonalni matriki U_1, V_1, da bo $A = U_1 B V_1^T$ in B bidiagonalna,
2) izračunaj SVD za B: $B = U_2 \Sigma V_2^T$,
3) SVD razcep za A je $A = (U_1 U_2)\Sigma(V_1 V_2)^T$.

Redukcijo na bidiagonalno obliko izvedemo s Householderjevimi zrcaljenji:

$$P_4 P_3 P_2 P_1 A \widetilde{P}_1 \widetilde{P}_1 = \begin{bmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & 0 \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = B$$

Redukcija na bidiagonalno obliko

Pri računanju SVD za A (razen pri Jacobijevi metodi) matriko najprej reduciramo na bidiagonalno obliko. Postopek je:

1) poišči ortogonalni matriki U_1, V_1, da bo $A = U_1 B V_1^T$ in B bidiagonalna,
2) izračunaj SVD za B: $B = U_2 \Sigma V_2^T$,
3) SVD razcep za A je $A = (U_1 U_2)\Sigma(V_1 V_2)^T$.

Redukcijo na bidiagonalno obliko izvedemo s Householderjevimi zrcaljenji:

$$P_4 P_3 P_2 P_1 A \widetilde{P}_1 \widetilde{P}_1 = \begin{bmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & 0 \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = B$$

V nadaljevanju potrebujemo le zgornji kvadratni del matrike B, zato lahko predpostavimo, da je B kvadratna.

Ekonomična redukcija

Za redukcijo s Householderjevimi zrcaljenji potrebujemo $8mn^2 - 8n^3/3$ operacij.

Če je $m \gg n$, se splača najprej narediti QR razcep matrike A, kjer dobimo $m \times m$ ortogonalno matriko Q in $n \times n$ zgornjo trikotno matriko R_1, da je

$$Q^T A = \begin{bmatrix} R_1 \\ 0 \end{bmatrix},$$

Sedaj bidiagonaliziramo R_1, da dobimo

$$U_R^T R_1 V = B_1,$$

kjer je B_1 bidiagonalna $n \times n$ matrika. Na koncu dobimo

$$\begin{bmatrix} U_R^T & 0 \\ 0 & I \end{bmatrix} Q^T A V = \begin{bmatrix} B_1 \\ 0 \end{bmatrix} = B.$$

Za to redukcijo potrebujemo $4mn^2 + 4n^3$ operacij, torej je cenejša, če je $m \geq 5n/3$.

Prevedba na tridiagonalni simetrični lastni problem

Naj bo

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

Potem je

$$B^T B = \begin{bmatrix} a_1^2 & a_1 b_1 & & & \\ a_1 b_1 & a_2^2 + b_1^2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & a_{n-1} b_{n-1} \\ & & & a_{n-1} b_{n-1} & a_n^2 + b_{n-1}^2 \end{bmatrix}$$

simetrična tridiagonalna matrika. Podobno velja za BB^T.

Prevedba na tridiagonalni simetrični lastni problem

Naj bo

$$B = \begin{bmatrix} a_1 & b_1 & & \\ & \ddots & \ddots & \\ & & \ddots & b_{n-1} \\ & & & a_n \end{bmatrix}, \qquad C = \begin{bmatrix} 0 & B^T \\ B & 0 \end{bmatrix}.$$

Če vzamemo permutacijsko matriko

$$P = \begin{bmatrix} e_1 & e_{n+1} & e_2 & e_{n+2} & \cdots & e_n & e_{2n} \end{bmatrix},$$

ki predstavlja t.i. perfektno mešanje, potem ima $P^T CP$ obliko

$$\begin{bmatrix} 0 & a_1 & & & & & \\ a_1 & 0 & b_1 & & & & \\ & b_1 & 0 & a_2 & & & \\ & & a_2 & 0 & \ddots & & \\ & & & \ddots & \ddots & b_{n-1} & \\ & & & & b_{n-1} & 0 & a_n \\ & & & & & a_n & 0 \end{bmatrix}.$$

Reducibilnost

$$P^T \begin{bmatrix} 0 & B^T \\ B & 0 \end{bmatrix} P = \begin{bmatrix} 0 & a_1 & & & & \\ a_1 & 0 & b_1 & & & \\ & b_1 & 0 & \ddots & & \\ & & \ddots & \ddots & b_{n-1} & \\ & & & b_{n-1} & 0 & a_n \\ & & & & a_n & 0 \end{bmatrix}.$$

V primeru, ko je $a_i = 0$ ali $b_i = 0$, lahko naredimo deflacijo, zato lahko predpostavimo, da so vsi diagonalni in bidiagonalni elementi neničelni.

Posledica

Če za matriko B velja $a_i \neq 0$ za $i = 1, \ldots, n$ in $b_i \neq 0$ za $i = 1, \ldots, n-1$, potem ima B same enostavne singularne vrednosti.

Enostranska QR iteracija

Naj bo

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

Mislimo si, da računamo implicitno QR iteracijo z Wilkinsonovimi premiki za $B^T B$, a $B^T B$ nikoli ne izračunamo eksplicitno.

Enostranska QR iteracija

Naj bo

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

Mislimo si, da računamo implicitno QR iteracijo z Wilkinsonovimi premiki za $B^T B$, a $B^T B$ nikoli ne izračunamo eksplicitno.

Za Wilkinsonov premik potrebujemo lastne vrednosti desne spodnje 2×2 podmatrike $B^T B$

$$\begin{bmatrix} a_{n-1}^2 + b_{n-2}^2 & a_{n-1} b_{n-1} \\ a_{n-1} b_{n-1} & a_n^2 + b_{n-1}^2 \end{bmatrix},$$

za premik σ^2 pa vzamemo tisto lastno vrednost, ki je bližja $a_n^2 + b_{n-1}^2$.

Premikanje grbe

Določimo Givensovo rotacijo $R = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$, da bo (glede na prvi stolpec $B^T B$)
$R \begin{bmatrix} a_1^2 - \sigma^2 \\ a_1 b_2 \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix}$ in vzamemo $R_{12}' = \begin{bmatrix} R & 0 \\ 0 & I \end{bmatrix}$.

Premikanje grbe

Določimo Givensovo rotacijo $R = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$, da bo (glede na prvi stolpec $B^T B$) $R \begin{bmatrix} a_1^2 - \sigma^2 \\ a_1 b_2 \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix}$ in vzamemo $R'_{12} = \begin{bmatrix} R & 0 \\ 0 & I \end{bmatrix}$.

Poglejmo zgled premikanja grbe na primeru matrike 5×5.

$$B = \begin{bmatrix} \times & \times & & & \\ & \times & \times & & \\ & & \times & \times & \\ & & & \times & \times \\ & & & & \times \end{bmatrix}.$$

Premikanje grbe

Določimo Givensovo rotacijo $R = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$, da bo (glede na prvi stolpec $B^T B$)
$R \begin{bmatrix} a_1^2 - \sigma^2 \\ a_1 b_2 \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix}$ in vzamemo $R'_{12} = \begin{bmatrix} R & 0 \\ 0 & I \end{bmatrix}$.

Poglejmo zgled premikanja grbe na primeru matrike 5×5.

$$B R'_{12} = \begin{bmatrix} \times & \times & & & \\ + & \times & \times & & \\ & & \times & \times & \\ & & & \times & \times \\ & & & & \times \end{bmatrix}.$$

Premikanje grbe

Določimo Givensovo rotacijo $R = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$, da bo (glede na prvi stolpec $B^T B$)
$R \begin{bmatrix} a_1^2 - \sigma^2 \\ a_1 b_2 \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix}$ in vzamemo $R'_{12} = \begin{bmatrix} R & 0 \\ 0 & I \end{bmatrix}$.

Poglejmo zgled premikanja grbe na primeru matrike 5×5.

$$R_{12} B R'_{12} = \begin{bmatrix} \times & \times & + & & \\ & \times & \times & & \\ & & \times & \times & \\ & & & \times & \times \\ & & & & \times \end{bmatrix}.$$

Premikanje grbe

Določimo Givensovo rotacijo $R = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$, da bo (glede na prvi stolpec $B^T B$)
$R \begin{bmatrix} a_1^2 - \sigma^2 \\ a_1 b_2 \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix}$ in vzamemo $R_{12}' = \begin{bmatrix} R & 0 \\ 0 & I \end{bmatrix}$.

Poglejmo zgled premikanja grbe na primeru matrike 5×5.

$$R_{12} B R_{12}' {\color{red}R_{23}'} = \begin{bmatrix} \times & \times & & & \\ & \times & \times & & \\ & {\color{red}+} & \times & \times & \\ & & & \times & \times \\ & & & & \times \end{bmatrix}.$$

Premikanje grbe

Določimo Givensovo rotacijo $R = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$, da bo (glede na prvi stolpec $B^T B$) $R \begin{bmatrix} a_1^2 - \sigma^2 \\ a_1 b_2 \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix}$ in vzamemo $R'_{12} = \begin{bmatrix} R & 0 \\ 0 & I \end{bmatrix}$.

Poglejmo zgled premikanja grbe na primeru matrike 5×5.

$$R_{23} R_{12} B R'_{12} R'_{23} = \begin{bmatrix} \times & \times & & & \\ & \times & \times & + & \\ & & \times & \times & \\ & & & \times & \times \\ & & & & \times \end{bmatrix}.$$

Premikanje grbe

Določimo Givensovo rotacijo $R = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$, da bo (glede na prvi stolpec $B^T B$) $R \begin{bmatrix} a_1^2 - \sigma^2 \\ a_1 b_2 \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix}$ in vzamemo $R'_{12} = \begin{bmatrix} R & 0 \\ 0 & I \end{bmatrix}$.

Poglejmo zgled premikanja grbe na primeru matrike 5×5.

$$R_{23} R_{12} B R'_{12} R'_{23} {\color{red}R'_{34}} = \begin{bmatrix} \times & \times & & & \\ & \times & \times & & \\ & & \times & \times & \\ & & {\color{red}+} & \times & \times \\ & & & & \times \end{bmatrix}.$$

Premikanje grbe

Določimo Givensovo rotacijo $R = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$, da bo (glede na prvi stolpec $B^T B$) $R \begin{bmatrix} a_1^2 - \sigma^2 \\ a_1 b_2 \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix}$ in vzamemo $R'_{12} = \begin{bmatrix} R & 0 \\ 0 & I \end{bmatrix}$.

Poglejmo zgled premikanja grbe na primeru matrike 5×5.

$$R_{34} R_{23} R_{12} B R'_{12} R'_{23} R'_{34} = \begin{bmatrix} \times & \times & & & \\ & \times & \times & & \\ & & \times & \times & + \\ & & & \times & \times \\ & & & & \times \end{bmatrix}.$$

Premikanje grbe

Določimo Givensovo rotacijo $R = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$, da bo (glede na prvi stolpec $B^T B$)
$R \begin{bmatrix} a_1^2 - \sigma^2 \\ a_1 b_2 \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix}$ in vzamemo $R'_{12} = \begin{bmatrix} R & 0 \\ 0 & I \end{bmatrix}$.

Poglejmo zgled premikanja grbe na primeru matrike 5×5.

$$R_{34} R_{23} R_{12} B R'_{12} R'_{23} R'_{34} {\color{red} R'_{45}} = \begin{bmatrix} \times & \times & & & \\ & \times & \times & & \\ & & \times & \times & \\ & & & \times & \times \\ & & & {\color{red}+} & \times \end{bmatrix} .$$

Premikanje grbe

Določimo Givensovo rotacijo $R = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$, da bo (glede na prvi stolpec $B^T B$) $R \begin{bmatrix} a_1^2 - \sigma^2 \\ a_1 b_2 \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix}$ in vzamemo $R_{12}' = \begin{bmatrix} R & 0 \\ 0 & I \end{bmatrix}$.

Poglejmo zgled premikanja grbe na primeru matrike 5×5.

$$\color{red}{R_{45}}\color{black} R_{34} R_{23} R_{12} B R_{12}' R_{23}' R_{34}' R_{45}' = \begin{bmatrix} \times & \times & & & \\ & \times & \times & & \\ & & \times & \times & \\ & & & \times & \times \\ & & & & \times \end{bmatrix} = \widetilde{B}.$$

Premikanje grbe

Določimo Givensovo rotacijo $R = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$, da bo (glede na prvi stolpec $B^T B$)
$R \begin{bmatrix} a_1^2 - \sigma^2 \\ a_1 b_2 \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix}$ in vzamemo $R'_{12} = \begin{bmatrix} R & 0 \\ 0 & I \end{bmatrix}$.

Poglejmo zgled premikanja grbe na primeru matrike 5×5.

$$R_{45} R_{34} R_{23} R_{12} B R'_{12} R'_{23} R'_{34} R'_{45} = \begin{bmatrix} \times & \times & & & \\ & \times & \times & & \\ & & \times & \times & \\ & & & \times & \times \\ & & & & \times \end{bmatrix} = \widetilde{B}.$$

Za en korak enostranske QR iteracije porabimo $30n + \mathcal{O}(1)$ osnovnih operacij in $2n$ kvadratnih korenov.

Kriterija za deflacijo sta $|b_i| \le \epsilon(|a_i| + |a_{i+1}|)$ in $|a_i| \le \epsilon \|B\|$.

LR algoritem

LR algoritem (T simetrična pozitivno definitna)

$T_0 = T$
$i = 0, 1, 2, \ldots$:
  $T_i = V_i^T V_i$ (razcep Choleskega)
  $T_{i+1} = V_i V_i^T$

LR algoritem

LR algoritem (T simetrična pozitivno definitna)

$T_0 = T$
$i = 0, 1, 2, \ldots$:
  $T_i = V_i^T V_i$ (razcep Choleskega)
  $T_{i+1} = V_i V_i^T$

T_i konvergira proti diagonalni matriki lastnih vrednosti matrike T. Matriki T_i in T_{i+1} sta podobni, saj velja

$$T_{i+1} = V_i V_i^T = V_i V_i^T V_i V_i^{-1} = V_i T_i V_i^{-1}.$$

Če je T tridiagonalna, se tridiagonalnost med algoritmom ohranja.

LR algoritem

LR algoritem (T simetrična pozitivno definitna)

$T_0 = T$
$i = 0, 1, 2, \ldots$:
  $T_i = V_i^T V_i$ (razcep Choleskega)
  $T_{i+1} = V_i V_i^T$

T_i konvergira proti diagonalni matriki lastnih vrednosti matrike T. Matriki T_i in T_{i+1} sta podobni, saj velja

$$T_{i+1} = V_i V_i^T = V_i V_i^T V_i V_i^{-1} = V_i T_i V_i^{-1}.$$

Če je T tridiagonalna, se tridiagonalnost med algoritmom ohranja.

Stabilnost in konvergenco zagotavlja naslednji izrek.

Lema

Dva koraka LR algoritma ustrezata enemu koraku QR algoritma brez premikov.

dqds metoda

Namesto enega koraka QR metode lahko naredimo dva koraka LR metode. Konvergenco pospešimo s premiki, paziti pa moramo na pozitivno definitnost.

LR algoritem s premiki

$T_0 = T$
$i = 0, 1, 2, \ldots$:
  izberi premik $\tau_i^2 > 0$, tako da je $\tau_i^2 \leq \lambda_n(T)$
  $T_i - \tau_i^2 I = V_i^T V_i$ (razcep Choleskega)
  $T_{i+1} = V_i V_i^T + \tau_i^2 I$

Matriki T_{i+1} in T_i sta podobni, saj velja

$$T_{i+1} = V_i V_i^T + \tau_i^2 I = V_i^{-T} V_i^T (V_i V_i^T + \tau_i^2 I) = V_i^{-T} T_i V_i^T.$$

dqds metoda

Namesto enega koraka QR metode lahko naredimo dva koraka LR metode. Konvergenco pospešimo s premiki, paziti pa moramo na pozitivno definitnost.

LR algoritem s premiki

$T_0 = T$
$i = 0, 1, 2, \ldots$:
  izberi premik $\tau_i^2 > 0$, tako da je $\tau_i^2 \leq \lambda_n(T)$
  $T_i - \tau_i^2 I = V_i^T V_i$ (razcep Choleskega)
  $T_{i+1} = V_i V_i^T + \tau_i^2 I$

Matriki T_{i+1} in T_i sta podobni, saj velja

$$T_{i+1} = V_i V_i^T + \tau_i^2 I = V_i^{-T} V_i^T (V_i V_i^T + \tau_i^2 I) = V_i^{-T} T_i V_i^T.$$

Pri dqds metodi delamo LR metodo s premiki za $B^T B$ a te matrike ne računamo eksplicitno. Iz bidiagonalne matrike B_i dobimo bidiagonalno B_{i+1}, pri tem pa velja, da če bi naredili LR s premikom za $B_i^T B_i$, bi dobili $B_{i+1}^T B_{i+1}$.

Osnovna ideja dqds metode

$$B_i = \begin{bmatrix} a_1 & b_1 & & \\ & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} \\ & & & a_n \end{bmatrix} \quad \text{in} \quad B_{i+1} = \begin{bmatrix} \widetilde{a}_1 & \widetilde{b}_1 & & \\ & \ddots & \ddots & \\ & & \widetilde{a}_{n-1} & \widetilde{b}_{n-1} \\ & & & \widetilde{a}_n \end{bmatrix}.$$

S primerjavo $T_i = B_i^T B_i + \tau_i^2 I$ in $T_{i+1} = B_{i+1}^T B_{i+1} + \tau_{i+1}^2 I = B_i B_i^T + \tau_i^2 I$ pridemo do osnovnega algoritma, kjer je $\delta = \tau_{i+1}^2 - \tau_i^2 \geq 0$ in privzamemo $b_0 = \widetilde{b}_0 = 0$.

Osnovni algoritem

$$\begin{aligned} & k = 1, \ldots, n-1 \\ & \quad \widetilde{a}_k^2 = a_k^2 + b_k^2 - \widetilde{b}_{k-1}^2 - \delta \\ & \quad \widetilde{b}_k^2 = a_{k+1}^2 b_k^2 / \widetilde{a}_k^2 \\ & \widetilde{a}_n^2 = a_n^2 - \widetilde{b}_{n-1}^2 - \delta \end{aligned}$$

qds algoritem

Osnovni algoritem

$$
\begin{aligned}
&k = 1, \dots, n-1 \\
&\quad \widetilde{a}_k^2 = a_k^2 + b_k^2 - \widetilde{b}_{k-1}^2 - \delta \\
&\quad \widetilde{b}_k^2 = a_{k+1}^2 b_k^2 / \widetilde{a}_k^2 \\
&\widetilde{a}_n^2 = a_n^2 - \widetilde{b}_{n-1}^2 - \delta
\end{aligned}
$$

Ves čas lahko računamo s kvadrati in korenimo le na koncu. Če označimo $c_k = a_k^2$ in $d_k = b_k^2$, dobimo novo različico algoritma:

qds algoritem

$$
\begin{aligned}
&k = 1, \dots, n-1 \\
&\quad \widetilde{c}_k = c_k + d_k - \widetilde{d}_{k-1} - \delta \\
&\quad \widetilde{d}_k = c_{k+1} d_k / \widetilde{c}_k \\
&\widetilde{c}_n = c_n - \widetilde{d}_{n-1} - \delta
\end{aligned}
$$

Novi algoritem porabi le $5n + \mathcal{O}(1)$ operacij za en korak.

Končni dqds algoritem

qds

$$
\begin{aligned}
& k = 1, \ldots, n-1 \\
& \quad \widetilde{c}_k = c_k + d_k - \widetilde{d}_{k-1} - \delta \\
& \quad \widetilde{d}_k = c_{k+1} d_k / \widetilde{c}_k \\
& \widetilde{c}_n = c_n - \widetilde{d}_{n-1} - \delta
\end{aligned}
$$

dqds

$$
\begin{aligned}
& g = c_1 - \delta \\
& k = 1, \ldots, n-1 \\
& \quad \widetilde{c}_k = g + d_k \\
& \quad t = c_{k+1} / \widetilde{c}_k \\
& \quad \widetilde{d}_k = d_k \cdot t \\
& \quad g = g \cdot t - \delta \\
& \widetilde{c}_n = g
\end{aligned}
$$

Če definiramo $g_k = c_k - \widetilde{d}_{k-1} - \delta$, potem velja

$$
g_k = c_k - \frac{c_k d_{k-1}}{\widetilde{c}_{k-1}} - \delta = c_k \frac{\widetilde{c}_{k-1} - d_{k-1}}{\widetilde{c}_{k-1}} - \delta = c_k \frac{c_{k-1} - \widetilde{d}_{k-2} - \delta}{\widetilde{c}_{k-1}} - \delta = \frac{c_k}{\widetilde{c}_{k-1}} g_{k-1} - \delta.
$$

Nov algoritem ima eno množenje namesto seštevanja. Izkaže se, da nam ravno to zagotavlja visoko obratno stabilnost.

Stabilnost dqds algoritma brez premikov

Izrek

Če je B bidiagonalna nesingularna matrika s singularnimi vrednostmi $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$, potem za izračunane singularne vrednosti $\widetilde{\sigma}_1 \geq \cdots \geq \widetilde{\sigma}_n > 0$ po dqds algoritmu velja (če ne uporabljamo premikov)

$$\frac{|\widetilde{\sigma}_i - \sigma_i|}{\sigma_i} \leq (10n - 5)u + O(u^2).$$