Numerička analiza
24. predavanje

Saša Singer

singer@math.hr

web.math.hr/~singer

PMF – Matematički odjel, Zagreb

Sadržaj predavanja

- Algoritmi za računanje svih svojstvenih vrijednosti simetričnih matrica:
  - Jacobijev algoritam.
  - Algoritam "podijeli pa vladaj".
- Algoritam za svojstvene vrijednosti simetričnih matrica u zadanom intervalu.
  - Algoritam bisekcije.

Jacobijev algoritam

Dijagonalizacija rotacijom u ravnini

Simetričnu 2×2 matricu

$$A = \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{bmatrix}$$

možemo dijagonalizirati korištenjem jedne rotacije $R(\theta)$,

$$A' = \begin{bmatrix} a'_{ii} & 0 \\ 0 & a'_{jj} \end{bmatrix} = R(\theta) A R^*(\theta),$$

gdje je

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

Računanje kuta

Uz kraće oznake $c := \cos\theta$, $s := \sin\theta$, $t := \operatorname{tg}\theta$, množenjem dobivamo:

$$\begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$

$$= \begin{bmatrix} a_{ii}c^2 - 2a_{ij}sc + a_{jj}s^2 & (c^2 - s^2)a_{ij} + (a_{ii} - a_{jj})sc \\ (c^2 - s^2)a_{ij} + (a_{ii} - a_{jj})sc & a_{ii}s^2 + 2a_{ij}sc + a_{jj}c^2 \end{bmatrix}.$$

Da bismo poništili element na mjestu $(1,2)$ (istovremeno, i element na mjestu $(2,1)$), moramo naći sinus i kosinus kuta za koji je

$$(c^2 - s^2)a_{ij} + (a_{ii} - a_{jj})sc = 0.$$

Računanje kuta

Primijetimo da je $\cos(2\theta) = c^2 - s^2$ i $\sin(2\theta) = 2sc$, pa prethodnu jednadžbu možemo napisati kao

$$\cos(2\theta)a_{ij} + \frac{1}{2}\sin(2\theta)(a_{ii} - a_{jj}) = 0.$$

Odatle možemo izračunati $\zeta := \operatorname{ctg}(2\theta)$ (bolje od $\operatorname{tg}(2\theta) = 1/\zeta$)

$$\zeta := \operatorname{ctg}(2\theta) = \frac{a_{jj} - a_{ii}}{2a_{ij}}.$$

Sada prvo treba izračunati t, pa zatim c i s. Idemo redom:

$$\operatorname{ctg}(2\theta) = \frac{\cos(2\theta)}{\sin(2\theta)} = \frac{c^2 - s^2}{2cs} = \frac{c^2 - s^2}{2cs} : \frac{c^2}{c^2} = \frac{1 - t^2}{2t}.$$

Računanje kuta

Budući da ζ znamo, treba riješiti kvadratnu jednadžbu po t

$$t^2 + 2\zeta t - 1 = 0.$$

Njezina rješenja su

$$t_{1,2} = \frac{-2\zeta \pm \sqrt{4\zeta^2 + 4}}{2} = -\zeta \pm \sqrt{\zeta^2 + 1}.$$

Po apsolutnoj vrijednosti manji od dva (tangensa) kuta, tj. $|t| \leq 1$ (to će nam poslije trebati kod dokaza konvergencije), je

$$t = \begin{cases} -\zeta + \sqrt{\zeta^2 + 1}, & \text{za } \zeta \geq 0, \\ -\zeta - \sqrt{\zeta^2 + 1}, & \text{za } \zeta < 0, \end{cases}$$

Računanje kuta

što možemo pisati kao

$$t = -\zeta + \operatorname{sign}(\zeta)\sqrt{\zeta^2+1} = \operatorname{sign}(\zeta)\bigl(-|\zeta| + \sqrt{\zeta^2+1}\bigr).$$

Jedino, ne smijemo tako računati,

- doći će do katastrofalnog kraćenja!

Katastrofalno kraćenje ćemo izbjeći deracionalizacijom izraza

$$t = \operatorname{sign}(\zeta)\bigl(-|\zeta| + \sqrt{\zeta^2+1}\bigr) \cdot \frac{|\zeta| + \sqrt{\zeta^2+1}}{|\zeta| + \sqrt{\zeta^2+1}} = \frac{\operatorname{sign}(\zeta)}{|\zeta| + \sqrt{\zeta^2+1}}.$$

Posljednji izraz se stabilno računa.

Autor algoritma: H. Rutishauser.

Računanje kuta

Sad možemo izračunati c i s. Iz $c^2 + s^2 = 1$, dijeljenjem s c^2, dobivamo $1 + t^2 = 1/c^2$, odnosno,

$$c^2 = \frac{1}{1 + t^2}.$$

Budući da smo izabrali manji od dva kuta, onda je njegov kosinus pozitivan, pa je

$$c = \frac{1}{\sqrt{1 + t^2}}.$$

Konačno, s ćemo izračunati kao

$$s = ct = \frac{t}{\sqrt{1 + t^2}}.$$

Dijagonalni elementi

I formulu za dijagonalne elemente možemo "poljepšati":

$$
\begin{aligned}
a'_{ii} &= a_{ii}c^2 - 2a_{ij}sc + a_{jj}s^2 \\
&= a_{ii}c^2 + a_{ii}s^2 - a_{ii}s^2 - 2a_{ij}sc + a_{jj}s^2 \\
&= a_{ii} + s^2(a_{jj} - a_{ii}) - 2a_{ij}sc = a_{ii} + 2a_{ij}s^2 \operatorname{ctg}(2\theta) - 2a_{ij}sc \\
&= a_{ii} + 2a_{ij}s^2 \frac{c^2 - s^2}{2sc} - 2a_{ij}sc = a_{ii} + a_{ij}t(c^2 - s^2) - 2a_{ij}sc \\
&= a_{ii} - a_{ij}t(s^2 - c^2 + 2c^2) = a_{ii} - a_{ij}t,
\end{aligned}
$$

$$
a'_{jj} = a_{ii}s^2 + 2a_{ij}sc + a_{jj}c^2 = a_{jj} + a_{ij}t.
$$

Drugu relaciju dobijemo, ili na isti način kao i prvu, ili korištenjem svojstva da sličnosti ne mijenjaju trag matrice.

Ravninske rotacije

Tehniku koju smo primijenili za dijagonalizaciju simetričnih matrica reda 2, želimo primijeniti i na simetrične matrice reda $n > 2$.

- Problem: Konstrukcija n-dimenzionalnih rotacija?
- Postoje 3-dimenzionalne rotacije (Euleovi kutovi). Bojanczyk i Lutoborski su ih iskoristili za svoju modifikaciju Jacobijevog algoritma.
- Uobičajeno — umjesto rotacija u n dimenzija, koristiti "puno" ravninskih rotacija.

Definicija. Ravninska rotacija $R(i, j, \theta)$ u ravnini (i, j) je jedinična matrica, osim na presjecima i-tog i j-tog retka i stupca, gdje je jednaka $R(\theta)$.

Ravninske rotacije — Jacobijev algoritam

Ideja Jacobijevog algoritma: Nekim redom prolaziti po gornjem (ili donjem trokutu) simetrične matrice i

- poništavati elemente na mjestima (i, j) korištenjem ravninskih rotacija.

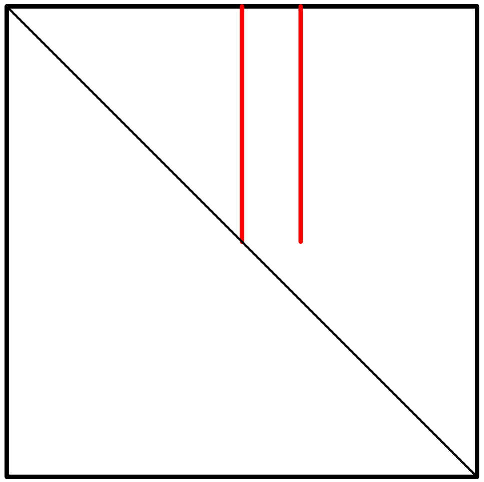
Što će se promijeniti obzirom na 2×2 algoritam?

- Algoritam je nužno iterativan.
- Formule za transformaciju elemenata a_{ii} i a_{jj} ostaju iste.
- Mijenjaju se i-ti i j-ti redak i stupac. Za $k \neq i, j$ imamo

$$\begin{aligned} a'_{ik} &= c \cdot a_{ik} - s \cdot a_{jk}, & \qquad a'_{ki} &= c \cdot a_{ki} + s \cdot a_{kj}, \\ a'_{jk} &= s \cdot a_{ik} + c \cdot a_{jk}, & \qquad a'_{kj} &= s \cdot a_{ki} + c \cdot a_{kj}. \end{aligned}$$

Ravninske rotacije — transformirani elementi

Ipak, treba pamtiti samo polovinu tih promjena (simetrija). Obično se pamte samo one u gornjem trokutu — kao na slici.

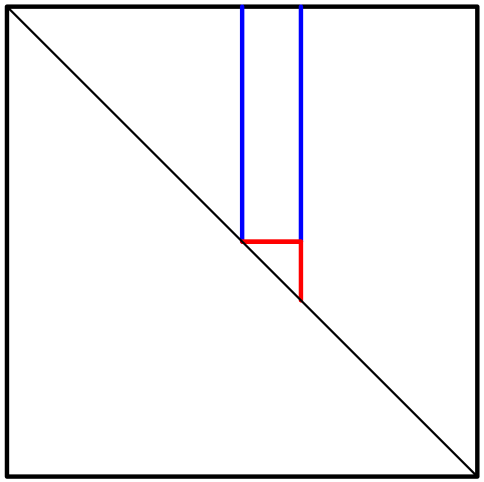


Konačno, primijetimo da prije i poslije transformacije vrijedi

$$(a'_{ik})^2 + (a'_{jk})^2 = a_{ik}^2 + a_{jk}^2, \quad i, j \neq k.$$

Ravninske rotacije — transformirani elementi

Ipak, treba pamtiti samo polovinu tih promjena (simetrija). Obično se pamte samo one u gornjem trokutu — kao na slici.

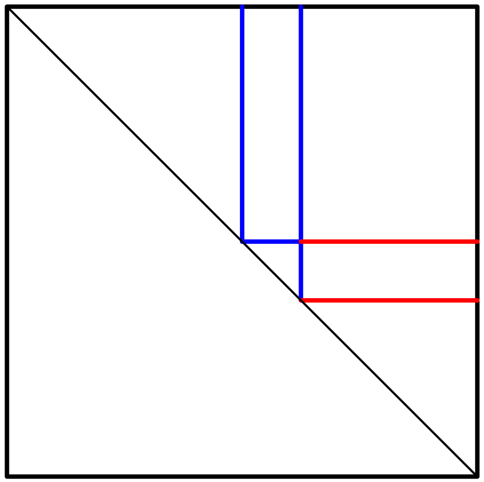


Konačno, primijetimo da prije i poslije transformacije vrijedi

$$(a'_{ik})^2 + (a'_{jk})^2 = a_{ik}^2 + a_{jk}^2, \quad i, j \neq k.$$

Ravninske rotacije — transformirani elementi

Ipak, treba pamtiti samo polovinu tih promjena (simetrija). Obično se pamte samo one u gornjem trokutu — kao na slici.



Konačno, primijetimo da prije i poslije transformacije vrijedi

$$(a'_{ik})^2 + (a'_{jk})^2 = a_{ik}^2 + a_{jk}^2, \quad i, j \neq k.$$

Redoslijed obilaska i konvergencija algoritma

Možemo li elemente gornjeg trokuta obilaziti bilo kojim redom da bismo postigli konvergenciju prema dijagonalnoj formi?

Na žalost, odgovor je ne možemo. Za neke strategije može se dokazati konvergencija:

- strategija koja poništava najveći vandijagonalni element,
- strategija koja ciklički poništava sve elemente po retcima

  $$(1,2),\ \ldots\ ,\ (1,n),\ (2,3),\ \ldots\ ,\ (2,n),\ \ldots\ ,\ (n-1,n),$$

  ili po stupcima

  $$(1,2),\ (1,3),\ (2,3),\ \ldots\ ,\ (1,n),\ \ldots\ ,\ (n-1,n),$$

- strategije koje su ekvivalentne prethodnim dvjema, tzv. "wavefront" strategije, ...

Konvergencija — poništi najveći element

Prvo, prisjetimo se da je Frobeniusova norma unitarno ekvivalentna, tj. za unitarne matrice U i V vrijedi

$$\|UAV^*\|_F = \|A\|_F.$$

Posebno, to vrijedi i za svaku ortogonalnu matricu Q, tj.

$$\|QAQ^*\|_F = \|A\|_F.$$

Umjesto same norme $\|A\|_F$, možemo promatrati kvadrat norme $\|A\|_F^2$. Njega možemo rascijepiti u dva dijela:

$$d(A) = \sum_{i=1}^{n} a_{ii}^2, \qquad 2\,\mathrm{off}(A) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} a_{ij}^2,$$

tzv. dijagonalni i vandijagonalni dio.

Konvergencija — poništi najveći element

Za bilo koju ravninsku rotaciju $R(p,q,\theta)$ u ravnini (p,q) vrijedi

$$d(R(p,q,\theta)\,A\,R^*(p,q,\theta)) = d(A) + 2\left(a_{pq}^2 - (\text{novi } a_{pq})^2\right).$$

Ako se radi o Jacobijevoj rotaciji, tj. onoj koja djeluje i poništava u (i,j) ravnini, onda je

$$\text{off}(R(i,j,\theta)\,A\,R^*(i,j,\theta)) = \text{off}(A) - a_{ij}^2.$$

To pokazuje da vandijagonalna norma monotono pada, tj. sigurno konvergira, samo je pitanje

- je li njezin limes jednak 0 ili ne.

Konvergencija — poništi najveći element

Ako se tzv. pivotni elementi a_{ij} (to su oni za poništavanje) u svakom koraku uzimaju tako da je

$$a_{ij}^2 \geq \text{prosjek} \left\{ a_{pq}^2 \mid p < q \right\} = \frac{2 \,\mathrm{off}(A)}{n(n-1)},$$

onda izlazi

$$\mathrm{off}(R(i,j,\theta)\, A\, R^*(i,j,\theta)) \leq \left(1 - \frac{2}{n(n-1)} \right) \mathrm{off}(A),$$

pa $\mathrm{off}(A)$ sigurno konvergira u nulu.

Prethodna relacija osigurava konvergenciju Jacobijevog algoritma prema dijagonalnoj formi.

Treba još pokazati da ta dijagonalna forma predstavlja baš svojstvene vrijednosti matrice A.

Teorem Wielandt–Hoffman

Teorem (Wielandt–Hoffman). Neka su D i E kvadratne matrice reda n i neka su matrice D i $D+E$ obje normalne.

Neka su $\lambda_1, \ldots, \lambda_n$ svojstvene vrijednosti od D i neka su $\hat{\lambda}_1, \ldots, \hat{\lambda}_n$ svojstvene vrijednosti od $D+E$, u nekom poretku.

Tada postoji permutacija indeksa π, takva da vrijedi

$$\left(\sum_{i=1}^{n} |\hat{\lambda}_{\pi(i)} - \lambda_i|^2 \right)^{1/2} \leq \|E\|_F.$$

■

Primjena teorema Wielandt–Hoffman

Ako za D uzmemo dijagonalu od A, a za E izvandijagonalne elemente od A, tako da je $D + E = A$, prethodni teorem daje

$$|\hat{\lambda}_{\pi(i)} - a_{ii}|^2 \leq \sum_{k=1}^{n} |\hat{\lambda}_{\pi(k)} - a_{kk}|^2 \leq 2\,\mathrm{off}(A).$$

Dakle, kad $\mathrm{off}(A) \to 0$, dijagonalni elementi od A konvergiraju prema nekoj permutaciji svojstvenih vrijednosti od A.

Taj poredak π se neće mijenjati u kasnijim fazama algoritma, kada je

$$\mathrm{off}(A) \leq \frac{1}{4} \min |\lambda_p - \lambda_q|^2 =: \sigma^2,$$

pri čemu se minimum uzima po različitim svojstvenim vrijednostima. Dakle, imamo “pravu” konvergenciju.

Kvadratična konvergencija

Jacobijev algoritam (asimptotski) kvadratično konvergira za jednostruke svojstvene vrijednosti. Što to znači?

Kada je $\max_{i \neq j} |a_{ij}| \leq \eta$, za neki $\eta < \sigma$ (v. prethodnu foliju za definiciju separacije σ), onda vrijedi

$$\max_{i \neq j} |\text{novi } a_{ij}| \leq \frac{\eta^2 (n-1)}{\sigma} = \mathcal{O}(\eta^2), \quad \text{kad } \eta \to 0.$$

I za višestruke svojstvene vrijednosti može se postići kvadratična konvergencija. Uvjet za to je da

- iste svojstvene vrijednosti moraju biti poredane jedna za drugom na dijagonali (tj. "u bloku").

Drugi Jacobijevi algoritmi

Osim za simetrične (realne) matrice, Jacobijev algoritam može se napisati i za hermitske (kompleksne) matrice, samo je ključna 2×2 podmatrica tada jednaka

$$\begin{bmatrix} \cos\theta & -\sin\theta \cdot e^{i\alpha} \\ \sin\theta \cdot e^{-i\alpha} & \cos\theta \end{bmatrix}.$$

Postoji i Jacobi-nalik algoritam za parove matrica, od kojih su obje simetrične, a jedna je još i pozitivno definitna. U tom algoritmu se koriste i hiperboličke rotacije

$$\begin{bmatrix} \mathrm{ch}\,\theta & \mathrm{sh}\,\theta \\ \mathrm{sh}\,\theta & \mathrm{ch}\,\theta \end{bmatrix}.$$

Još o Jacobijevom algoritmu

Zanimljivosti:

- Algoritam je dugo vremena bio “zaboravljen”, kao prespor, obzirom na QR algoritam.
- “Revoluciju” izazvao članak J. W. Demmela i K. Veselića 1992. godine, u kojem pokazuju da svojstvene vrijednosti izračunate Jacobijevim algoritmom mogu imati visoku relativnu točnost, što QR algoritam (u načelu) nema.
- Jacobijev algoritam se lako paralelizira, svođenjem na tzv. jednostrani algoritam (bit će još govora o tome kod singularnih vrijednosti).

Algoritam
“podijeli pa vladaj”

Osnovno o algoritmu

Algoritam "podijeli pa vladaj" (engl. "divide and conquer") autora J. J. M. Cuppena, objavljen je 1981. godine.

- Algoritam rekurzivno nalazi svojstvene vrijednosti trodijagonalnih matrica.
- Nakon toga koristi korekciju dobivenog rezultata matricom ranga 1.
- Tek 1995. su M. Gu i S. Eisenstat riješili kako taj algoritam treba stabilno računati svojstvene vektore.
- Algoritam se (uz diferencijalni qd algoritam) smatra najbržim algoritmom za računanje svojstvenih vrijednosti matrica.

Konstrukcija algoritma

Neka je T trodijagonalna matrica oblika

$$T = \left[\begin{array}{cccc|cccc}
a_1 & b_1 & & & & & & \\
b_1 & \ddots & \ddots & & & & & \\
 & \ddots & a_{m-1} & b_{m-1} & & & & \\
 & & b_{m-1} & a_m & b_m & & & \\
\hline
 & & & b_m & a_{m+1} & b_{m+1} & & \\
 & & & & b_{m+1} & \ddots & \ddots & \\
 & & & & & \ddots & a_{n-1} & b_{n-1} \\
 & & & & & & b_{n-1} & a_n
\end{array}\right].$$

Konstrukcija algoritma

Matricu T možemo napisati kao

$$\left[\begin{array}{cccc|cccc}
a_1 & b_1 & & & & & & \\
b_1 & \ddots & \ddots & & & & & \\
& \ddots & a_{m-1} & b_{m-1} & & & & \\
& & b_{m-1} & a_m - b_m & & & & \\
\hline
& & & & a_{m+1} - b_m & b_{m+1} & & \\
& & & & b_{m+1} & \ddots & \ddots & \\
& & & & & \ddots & a_{n-1} & b_{n-1} \\
& & & & & & b_{n-1} & a_n
\end{array}\right] + \cdots$$

Konstrukcija algoritma

$$
\cdots + \left[\begin{array}{ccc|ccc}
 & & & & & \\
 & & & & & \\
 & & b_m & b_m & & \\
\hline
 & & b_m & b_m & & \\
 & & & & & \\
 & & & & &
\end{array}\right],
$$

Konstrukcija algoritma

odnosno, kao

$$T = \left[\begin{array}{c:c} T_1 & 0 \\ \hdashline 0 & T_2 \end{array}\right] + b_m v v^T,$$

pri čemu je

$$v^T = [\,0, \ldots, 0, 1 \mid 1, 0, \ldots, 0\,].$$

Traženje svojstvenih vrijednosti metodom podijeli pa vladaj sastoji se od

- dva tridijagonalna svojstvena problema, za T_1 i T_2, i
- njihovog efikasnog ažuriranja matricama ranga 1.

Neka su $T_1 = Q_1 \Lambda_1 Q_1^T$ i $T_2 = Q_2 \Lambda_2 Q_2^T$ svojstveni rastavi matrica T_1 i T_2.

Konstrukcija algoritma

Svojstvene vrijednosti matrice T možemo izračunati ovako

$$T = \begin{bmatrix} T_1 & \\ & T_2 \end{bmatrix} + b_m v v^T = \begin{bmatrix} Q_1 \Lambda_1 Q_1^T & \\ & Q_2 \Lambda_2 Q_2^T \end{bmatrix} + b_m v v^T$$

$$= \begin{bmatrix} Q_1 & \\ & Q_2 \end{bmatrix} \left(\begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix} + b_m u u^T \right) \begin{bmatrix} Q_1^T & \\ & Q_2^T \end{bmatrix},$$

gdje je, iz oblika vektora v,

$$u = \begin{bmatrix} Q_1^T & \\ & Q_2^T \end{bmatrix} v = \begin{bmatrix} \text{posljednji stupac matrice } Q_1^T \\ \text{prvi stupac matrice } Q_2^T \end{bmatrix}.$$

Konstrukcija algoritma

Problem: nalaženje svojstvenih vrijednosti λ matrice oblika

$$\widehat{D} = D + \rho u u^T,$$

pri čemu je D dijagonalna, $\rho = b_m$, a u je poznati vektor.

Prvo pretpostavimo da je matrica $D - \lambda I$ nesingularna, pa karakteristični polinom matrice $\widehat{D}$ glasi

$$\det(D + \rho u u^T - \lambda I) = \det[\,(D - \lambda I)\,(I + \rho(D - \lambda I)^{-1} u u^T)\,].$$

Iz pretpostavke o nesingularnosti $D - \lambda I$ izlazi da mora biti

$$\det(I + \rho(D - \lambda I)^{-1} u u^T) = 0$$

kad god je λ svojstvena vrijednost matrice $\widehat{D}$.

Konstrukcija algoritma

Nadalje, matrica $I + \rho(D - \lambda I)^{-1} uu^T$ je jedinična matrica plus matrica ranga 1, za koju se lako računa determinanta.

Lema. Ako su x i y vektori iz $\mathbb{R}^n$, onda vrijedi

$$\det(I + xy^T) = 1 + y^T x.$$

Dokaz. Determinanta je produkt svih svojstvenih vrijednosti matrice, pa je

$$\det(I + xy^T) = \prod_{i=1}^{n} \lambda_i(I + xy^T).$$

S druge strane, $I + xy^T$ je polinom u matrici xy^T, pa vrijedi

$$\lambda_i(I + xy^T) = 1 + \lambda_i(xy^T), \quad i = 1, \ldots, n.$$

Konstrukcija algoritma

Dobivamo da je

$$\det(I + xy^T) = \prod_{i=1}^{n} \left(1 + \lambda_i(xy^T)\right).$$

Treba još naći svojstvene vrijednosti matrice xy^T.

Ako je $x = 0$ ili $y = 0$, onda je $xy^T = 0$ (u $\mathbb{R}^n$) i $y^T x = 0$, pa tvrdnja očito vrijedi.

Za $x, y \neq 0$, matrica xy^T ima rang jednak 1. Neka je λ_0 jedina svojstvena vrijednost matrice xy^T različita od 0. Onda imamo

$$\det(I + xy^T) = 1 + \lambda_0.$$

Svojstvenu vrijednost λ_0 nalazimo iz traga matrice.

Konstrukcija algoritma

Trag matrice xy^T jednak je

- s jedne strane, zbroju svih dijagonalnih elemenata te matrice,
- s druge strane, zbroju svih svojstvenih vrijednosti te matrice $= \lambda_0$.

Odavde slijedi

$$\lambda_0 = \mathrm{tr}(xy^T) = \sum_{i=1}^{n} x_i y_i = y^T x.$$

Kad to uvrstimo u formulu za determinantu, izlazi

$$\det(I + xy^T) = 1 + \lambda_0 = 1 + y^T x.$$

Konstrukcija algoritma

Iz prethodne leme, jer je $D - \lambda I$ dijagonalna, izlazi

$$\det(I + \rho(D - \lambda I)^{-1} u u^T) = 1 + \rho u^T (D - \lambda I)^{-1} u$$
$$= 1 + \rho \sum_{i=1}^{n} \frac{u_i^2}{d_i - \lambda} =: f(\lambda).$$

Jednadžba $f(\lambda) = 0$ obično se zove sekularna jednadžba, a vrijednosti λ za koje je $f(\lambda) = 0$ su svojstvene vrijednosti.

U najjednostavnijem slučaju su svi d_i različiti, a svi $u_i \neq 0$. Budući da je f monotona (f' i f'' fiksnog znaka), onda

- Newtonova metoda za nalaženje nultočaka konvergira, ali
- konvergencija može biti vrlo spora, ako su d_i vrlo maleni.

Svojstveni vektori

Lema. Ako je α svojstvena vrijednost matrice $D + \rho u u^T$, onda je $(D - \alpha I)^{-1} u$ svojstveni vektor za tu svojstvenu vrijednost.

Dokaz. Provodi se direktno po definiciji svojstvenih vektora. ■

Nažalost, prethodna formula za računanje svojstvenih vektora je numerički nestabilna, pa svojstveni vektori mogu biti neortogonalni. Zato se oni računaju na drugi način.

Algoritam

Za simetričnu trodijagonalnu matricu T svojstvene vrijednosti i vektori $T = Q\Lambda Q^T$ dobivaju se rekurzivnim algoritmom.

proc dc_eig $(T,\ Q,\ \Lambda)$
if $n = 1$ **return** $Q = 1,\ \Lambda = T$
else
  form $T = \mathrm{diag}(T_1, T_2) + b_m v v^T$
  call proc dc_eig $(T_1,\ Q_1,\ \Lambda_1)$
  call proc dc_eig $(T_2,\ Q_2,\ \Lambda_2)$
  form $\widehat{D} = D + \rho u u^T$ **from** $Q_1,\ \Lambda_1,\ Q_2,\ \Lambda_2$
  find eigenvalues Λ and **eigenvectors** Q' of $\widehat{D}$
  form $Q = \mathrm{diag}(Q_1, Q_2) \cdot Q'$
  return Q and Λ
end if

Bisekcija i inverzna iteracija

Osnovno o algoritmu

Metoda bisekcije služi za nalaženje svojstvenih vrijednosti hermitskih matrica

- u nekom zadanom poluotvorenom intervalu $[a, b\rangle$.

Teorem koji se koristi je Sylvesterov teorem o inerciji.

Definicija. Inercija hermitske/simetrične matrice A je trojka brojeva (n, z, p), pri čemu je n broj negativnih, z broj nula, a p broj pozitivnih svojstvenih vrijednosti matrice A.

Teorem (Sylvesterov teorem o inerciji). Neka je A simetrična matrica, a X nesingularna. Onda matrice A i $X^T AX$ imaju istu inerciju. ■

Transformacija $A \mapsto X^T AX$ zove se kongruencija.

Konstrukcija algoritma

Na drugi način, inerciju matrice $A - wI$ možemo izraziti i ovako:

- n je broj svojstvenih vrijednosti matrice A manjih od w,
- z je broj svojstvenih vrijednosti matrice A jednakih w,
- p broj svojstvenih vrijednosti matrice A većih od w.

Neka je

- $n_b =$ broj svojstvenih vrijednosti od A manjih od b, tj. broj negativnih svojstvenih vrijednosti matrice $A - bI$,
- $n_a =$ broj svojstvenih vrijednosti od A manjih od a, tj. broj negativnih svojstvenih vrijednosti matrice $A - aI$.

Iz prethodnih razmatranja odmah slijedi da je broj svojstvenih vrijednosti matrice A u intervalu $[a, b\rangle$ jednak $n_b - n_a$.

Konstrukcija algoritma

Kako se taj broj $n_b - n_a$ može izračunati?

Napravimo li LDL^T faktorizaciju matrice $A - wI$, gdje je L donjetrokutasta matrica s jedinicama na dijagonali,

- odmah "čitamo" inerciju matrice $A - wI$,
- jer ona "piše" na dijagonali matrice D — u predznacima tih elemenata (Sylvesterov teorem).

Označimo s $\text{neg\_ev}(A, w) = n_w$. Onda je $\text{neg\_ev}(A, w) =$ broj negativnih dijagonalnih elemenata matrice D.

Primijetite da bi računanje s punim matricama bilo skupo, pa se, umjesto toga,

- brzo računa za trodijagonalne matrice, uz uvjet da se ne pivotira u LDL^T faktorizaciji.

Algoritam bisekcije — ideja

Ideja algoritma: Algoritam se sastoji u prebrajanju svojstvenih vrijednosti na intervalu, koji se svaki puta smanjuje na polovinu svoje prethodne duljine.

- Ako na intervalu $[a, b\rangle$ ima svojstvenih vrijednosti, on se raspolovi, i prebroji se koliko je svojstvenih vrijednosti ostalo u svakoj polovini intervala.
- Ako je duljina (raspolovljenog) intervala veća od tolerancije, i on sadrži svojstvene vrijednosti, stavljamo ga u listu za daljnje raspolavljanje.
- Ako je duljina intervala manja od zadane tolerancije, a on ima svojstvenih vrijednosti, izbacujemo ga s liste, a sredinu intervala proglasimo svojstvenom vrijednošću.
- Postupak ponavljamo dok lista intervala nije prazna.

Algoritam bisekcije

Metoda bisekcije za danu simetričnu matricu A nalazi sve svojstvene vrijednosti unutar intervala $[a, b\rangle$, s tim da se

- sve svojstvene vrijednosti koje se razlikuju za manje od tol smatraju jednakima.

$$
\begin{array}{l}
\qquad\text{/* } \mathit{work\_list} \text{ je lista intervala za bisekciju */} \\
n_a = \text{neg\_ev}(A, a) \\
n_b = \text{neg\_ev}(A, b) \\
\textbf{if } n_a = n_b \textbf{ quit} \quad \text{/* nema sv. vrijednosti unutar */} \\
\textbf{else} \\
\quad \textbf{put } [a, n_a, b, n_b] \textbf{ onto } \mathit{work\_list} \\
\textbf{end if} \\
\textbf{while } \mathit{work\_list} \neq \emptyset \textbf{ do} \\
\quad \textbf{remove } [low, n_{low}, up, n_{up}] \textbf{ from } \mathit{work\_list}
\end{array}
$$

Algoritam bisekcije

if $up - low < tol$ **then**
  print "$n_{up} - n_{low}$ sv. vrijednosti u $[up, low\rangle$"
else
  $mid = (up + low)/2$
  $n_{mid} = \text{neg\_ev}(A, mid)$
  if $n_{mid} > n_{low}$
    put $[low, n_{low}, mid, n_{mid}]$ **onto** $work\_list$
  end if
  if $n_{up} > n_{mid}$
    put $[mid, n_{mid}, up, n_{up}]$ **onto** $work\_list$
  end if
end if
end while

Bisekcija za trodijagonalne matrice

Ako je $A - wI$ trodijagonalna, onda je njezina LDL^T faktorizacija bez pivotiranja

$$A - wI = \begin{bmatrix} a_1 - w & b_1 & & \\ b_1 & a_2 - w & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ & & b_{n-1} & a_n - w \end{bmatrix} = LDL^T,$$

gdje je

$$L = \begin{bmatrix} 1 & & & \\ l_1 & \ddots & & \\ & \ddots & \ddots & \\ & & l_{n-1} & 1 \end{bmatrix}, \qquad D = \begin{bmatrix} d_1 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & d_n \end{bmatrix}.$$

Bisekcija za trodijagonalne matrice

Uspoređivanjem elemenata, dobivamo početak rekurzije

$$d_1 = a_1 - w,$$

a zatim

$$l_{i-1}^2 d_{i-1} + d_i = a_i - w, \quad d_{i-1} l_{i-1} = b_{i-1},$$

za $i = 2, \ldots, n$.

Supstituiramo li l_{i-1} iz druge jednadžbe u prvu, dobivamo rekurziju

$$d_i = (a_i - w) - \frac{b_{i-1}^2}{d_{i-1}}, \quad i = 2, \ldots, n.$$

Iako se čini opasnim, ova rekurzija je, zbog trodijagonalnosti matrice, vrlo stabilna.