

UNM

Linearni sistemi

NORME: $\|A\|_1 = \max_{j \in \{1..n\}} (\sum_{i=1}^n |a_{ij}|) =$ največji stolpec, $\|A\|_\infty = \|A^T\|_1 =$ največja vrstica
 $\|A\|_2 = \sigma_1 = \sqrt{\lambda_{\max}(A^H A)} =$ največja singularna vrednost, $\|A\|_F = \sqrt{\sum_{ij} a_{ij}^2} =$ gledamo kot vektor
Operatorska norma: $\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}$. Neenakosti: $\lambda \leq \|A\|$. $\|Ax\| \leq \|A\| \|x\|$.

$$\begin{aligned} \frac{1}{\sqrt{n}} \|A\|_F &\leq \|A\|_2 \leq \|A\|_F \\ \frac{1}{\sqrt{n}} \|A\|_1 &\leq \|A\|_2 \leq \sqrt{n} \|A\|_1 \\ \frac{1}{\sqrt{n}} \|A\|_\infty &\leq \|A\|_2 \leq \sqrt{n} \|A\|_\infty \\ N_\infty(A) &\leq \|A\|_2 \leq n N_\infty(A) \\ &\leq \|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty} \\ \|a_i\|_2, \|\alpha_i\|_2 &\leq \|A\|_2 \end{aligned}$$

Rešujemo sistem $Ax = b$. Za napako x velja ocena:

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\kappa(A)}{1 - \kappa(A) \frac{\|\Delta A\|}{\|A\|}} \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right)$$

Količina $\kappa(A)$ se imenuje občutljivost matrike. $\kappa(A) = \|A\| \|A^{-1}\|$. Velja $\kappa_2(A) = \frac{\sigma_1(A)}{\sigma_n(A)} \geq 1$.

LU RAZCEP s kompletnim pivotiranjem: matriko A zapišemo kot $PAQ = UL$, L sp. trikotna z 1 na diagonali in U zg. trikotna, ter P, Q permutacijski matriki stolpcev in vrstic. Algoritem:

```
Q = I, P = I
for j = 1 to n:
  r, q taka, da a_rq največji v podmatriki A(j+1:n)
  zamenjaj vrstici r in j v A, L, P // za delno pivotiranje
  zamenjaj stolpca q in j v A, L, Q // za kompletno pivotiranje
  for i = j+1 to n:
    l_ij = a_ij / a_jj
    for k = j+1 to n:
      a_ik = a_ik - l_ij * a_jk
```

Postopek na roke:

1. * Če delamo pivotiranje zamenjamo primerne vrstice in stolpce v A, P, Q , da je a_{00} največji.
2. Prvi stolpec delimo z a_{00} , razen a_{00} , ki ga pustimo na miru.
3. Za vsak element v podmatriki $A(2:n, 2:n)$: $a_{ij} = a_{ij} - a_{i1} \cdot a_{1j}$ (odštejemo produkt \leftarrow in \uparrow).
4. Ponovimo postopek na matriki $A(2:n, 2:n)$.

Delno pivotiranje uporablja samo matriko P , za LU razcep brez pivotiranja pa preskočimo 1.

Skalarni produkt potrebuje $2n$ operacij. Reševanje s premimi substitucijami potrebuje n^2 , z obratnimi $n^2 + n$. Reševanje z LU razcepom (brez pivotiranja) potrebuje $\frac{2}{3}n^3 + \frac{3}{2}n^2 + \frac{5}{6}n$ operacij.

Za izračunani LU razcep $\hat{L}\hat{U} = A + E$ velja $|E| \leq nu|\hat{L}||\hat{U}|$.

Pivotna rast: $g = \frac{\max u_{ij}}{\max a_{ij}}$. Pri delnem pivotiranju $g < 2^n$.

RAZCEP CHOLESKEGA: Za spd matriko A obstaja razcep $A = VV^T$.

```
for k = 1 to n:
  v_kk = sqrt(a_kk - sum(v_kj^2, j=1 to k))
  for i = k+1 to n:
    v_ik = 1/v_kk * (a_ik - sum(v_ij * v_kj, j = 1 to k))
```

Postopek na roke po stolpcih:

1. Če sem diagonalen element: odštejem od sebe skalarni produkt vrstice na levo same s sabo in se korenim.
2. Če nisem diagonalni: od sebe odštejem skalarni produkt vrstice levo od sebe z vrstico levo od mojega diagonalnega. Nato se delim z diagonalnim.

Razcep stane $\frac{1}{3}n^3$ operacij. Je obratno stabilno. Je enoličen.

Nelinearni sistemi

JACOBIJEVA ITERACIJA: Posplošitev navadne iteracije. Naj velja $G(\alpha) = \alpha$. Metoda: $x^{(r+1)} = G(x^{(r)})$. Točka α je privlačna, če velja $\rho(DG(\alpha)) < 1$. Dovolj je $\|DG(\alpha)\| < 1$. Konvergenca je linearna.

NEWTONOVA METODA: Posplošitev tangentne metode. Metoda: reši sistem $DF(x^{(r)})\Delta x^{(r)} = -F(x^{(r)})$. $x^{(r+1)} = x^{(r)} + \Delta x^{(r)}$. Konvergenca je kvadratična.

Problem najmanjših kvadratov

REŠEVANJE PREDOLOČENIH SISTEMOV: Za dan predoločen sistem $Ax = b$ rešujemo normalni sistem $A^T Ax = A^T b$. Če je A polnega ranga, je x enoličen. Rešujemo z razcepom Choleskega. Število operacij: $n^2 m + \frac{1}{3} n^3$.

QR razcep je bolj stabilen. Za $A \in \mathbb{R}^{m \times n}$ obstaja enoličen razcep $A = QR$, $Q^T Q = I$ in R zg. trikotna s pozitivnimi diagonalci. Za predoločen sistem rešimo $Rx = Q^T b$.

CGS IN MGS Klasična GS ortogonalizacija. Od vsakega stolpca a_k odštejemo pravokotne projekcije $a_i, i < k$. Algoritem ni najbolj stabilen.

```
for k = 1 to n:
  q_k = a_k
  for i = 1 to k-1:
    r_ik = q_i' * a_k (CGS) ALI = q_i' * q_k (MGS)
    q_k = q_k - r_ik q_i
  r_kk = ||q_k||
  q_k = q_k / r_kk
```

Za večjo natančnost izračunamo $[Ab] = [Qq_{n+1}][Rz; 0p]$ in rešimo $Rx = z$. Porabi $2nm^2$ operacij.

Razširjeni QR razcep: $A = \tilde{Q}\tilde{R}$, $Q \in \mathbb{R}^{m \times m}$ ortogonalna, R zgornje trapezna. $\tilde{Q} = [Q \ Q_1]$, $\tilde{R} = [R; 0]$.

GIVENSOVE ROTACIJE

Elemente v A po stolpcih enega po enega ubijamo z rotacijami. Rotacija, ki ubije element a_{ki} je $R_{ik}^T([ik], [i, k]) = [c \ s; -s \ c]$, in ostalo identiteta. Parametre nastavimo: $c = x_{ii}/r$, $s = x_{ki}/r$, $r = \sqrt{x_{ii}^2 + x_{ki}^2}$. \tilde{Q} dobimo kot produkt vseh rotacij, potrebnih za genocid elementov A . Rotacija spremeni samo i -to in k -to vrstico.

Število operacij: $3mn^2 - n^3$. Če potrebujemo \tilde{Q} , potem rabimo še dodatnih $6m^2 n - 3mn^2$ operacij.

```
Q = I_m
for i = 1 to n:
  for k = i+1 to m:
    r = sqrt(a_ii^2 + a_ki^2)
    c = a_ii/r, s = a_ki/r
    A([i,k], i:n) = [c s; -s c] A([i k], i:n)
    b([i, k]) = [c s; -s c] b([i, k]) // za predoločen sistem
    Q(i, [i k]) = Q(i, [i k]) [c -s; s c] // za matriko Q
Q = Q'
```

HAUSHOLDERJEVA ZRCALJENJA Definiramo $P = I - \frac{2}{w^T w} w w^T$. P je zrcaljenje prek ravnine z normalo w . $Px = x - \frac{1}{m}(x^T w)w$, $m = \frac{1}{2} w^T w$.

Da vektor x prezrcalimo tako, da mu uničimo vse razen prve komponente, uporabimo $w = [x_1 + \text{sign}(x_1)\|x\|_2; x_2; \dots x_n]$ in $m = \|x\|_2(\|x\|_2 + |x_1|)$. Število operacij za Pz je $4nm$ za w in m pa potrebujemo $2n$ operacij.

```
Q = I_m
for i = 1 to n:
  w_i iz R^{m-i+1}, ki prezrcali A(i:m, i) v +-k e_1
  A(i:m, i:n) = P_i * A(i:m, i:n)
  b(i:m) = P_i * b(i:m) // za predoločen sistem
  Q(i:m, i:n) = P_i * Q(i:m, i:n) // za matriko Q
Q = Q'
```

Reševanje predoločenega sistema tako stane $2mn^2 - \frac{2}{3}n^3$. Za \tilde{Q} potrebujemo še $4m^2 n - 2mn^2$ operacij. Za kvadratne sisteme je stabilnejši, a rabimo $\frac{4}{3}n^3$ operacij.

Za napako pri reševanju predoločenega sistema velja: $\frac{\|\Delta x\|}{\|x\|} \leq \frac{\varepsilon \kappa_2(A)}{1 - \varepsilon \kappa_2(A)} \left(2 + (\kappa_2(A) + 1) \frac{\|r\|}{\|A\| \|x\|} \right), r = Ax - b$.

Lastne vrednosti

Desni in levi lastni vektorji: $y^H A = \mu y^H$ in $Ax = \lambda x$. Levi in desni vektorji za različne l. vrednosti so pravokotni. Občutljivost lastne vrednosti je $\frac{1}{y_x^H x}$, kjer sta x in y normirana levi in desni lastni vektor.

POTENČNA METODA: Pravzaprav navadna iteracija. Izberemo si začetni vektor z in tolčemo čez matriko A in normiramo, dokler ne postane lastni vektor. Ta metoda ima linearno konvergenco k lastnemu vektorju za dominantno

lastno vrednost, če je le ena sama lastna vrednost največja. Hitrost konvergence je λ_1/λ_2 , kjer sta to dve največji lastni vrednosti.

```
z = ones(n, 1) // naključen neničeln vektor
for k = 1 to m: // m je veliko število
    y = A * z
    z = y / ||y||
```

Če imamo lastni vektor v , potem želimo imeti lastno vrednost λ . Najboljši približek je Rayleighov kvocient: $\rho(A, v) = \frac{v^H A v}{v^H v}$. Kot kriterij v potenčni metodi uporabimo $|A * z - p(A, z)| < \text{eps}$.

Če imamo dober približek $\tilde{\lambda}$ za lastno vrednost vrednost λ_i uporabimo inverzno iteracijo. Iščemo največjo vrednost matrike $(A - \tilde{\lambda}I)^{-1}$, ki ima lastne vrednosti $\frac{1}{\lambda_i - \tilde{\lambda}}$.

```
z = ones(n, 1)
for k = 0 to m:
    reši (A - lambda I)y = z
    z = y / ||y||
```

SCHUROVA FORMA: Za vsako matriko A obstaja Schurova forma S , da je $A = USU^H$, kjer je U unitarna in S zgornje trikotna. Na diagonali so lastne vrednosti. V primeru kompleksnih lastnih vrednosti imamo 2×2 bloke.

OTROGONALNA ITERACIJA: Za izračun Schurove forme. Z je lahko $n \times p$ matrika z ortonormiranimi stolpci, tako dobimo prvih p stolpcev Schurove forme. Za $p = 1$ je to potenčna metoda, za $p = n$, pa dobimo celo schurovo formo.

```
Z = eye(n) // naključna matrika z ortonormiranimi stolpci
for k = 0 to m:
    Y = A * Z
    [Q, R] = qr(Y)
    Z = Q
```

QR ITERACIJA: Najboljša metoda za izračun lastnih vrednosti A .

```
for k = 0 to m:
    [Q, R] = qr(A)
    A = R * Q
```

GERSGHORINOV IZREK: Naj bo $A \in \mathbb{C}^{n \times n}$, $C_i = \overline{K}(a_{ii}, r = \sum_{j=1, j \neq i}^n |a_{ij}|), i = 1, 2, \dots, n$. Potem vsaka lastna vrednost leži v vsaj enem Gerschgorinovem krogu. Če m krogov C_i sestavlja povezano množico, ločeno od ostalih $n - m$ krogov, potem ta množica vsebuje natanko m lastnih vrednosti.

Diagonalno dominantna matrika ($|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|$) je obrnljiva.

NLA

Singularni razcep

$A \in \mathbb{R}^{m \times n}$, $m \geq n$. Singularni razcep je razcep matirke A na $A = U\Sigma V^T$, U ortogonalna $m \times m$, V ortogonalna $n \times n$,

$\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} \in \mathbb{R}^{m \times n}$. Definiciji U in V po stolpcih: $A^T A v_i = \sigma_i^2 v_i$, $A v_i = \sigma_i u_i$. Recimo da ima matrika

prvih r singularnih vrednosti neničelnih. Potem lahko V razdelimo na dva dela, V_1 sestavljen iz lastnih vektorjev za neničelne lastne vrednosti $A^T A$ in V_2 sestavljen iz lastnih vektorjev za ničelno lastno vrednost. Bločno definiramo

$$A = \begin{bmatrix} m \times r & m \times m-r \end{bmatrix} \begin{bmatrix} r \times r & r \times n-r \\ m-r \times r & m-r \times n-r \end{bmatrix} \begin{bmatrix} r \times n \\ n-r \times n \end{bmatrix}$$

Definiramo lahko tudi ozek singularni razcep, pri katerem vzamemo samo U_1, S in V_1 . Velja:

- V_1 je ONB za $\text{im}(A^T)$
- V_2 je ONB za $\text{ker}(A)$
- U_1 je ONB za $\text{im}(A)$
- U_2 je ONB za $\text{ker}(A^T)$

Pseudoinverz: $A^+ = V\Sigma^+ U^T$, $\Sigma^+ = [1/\sigma_i]_i$. Rešitev po metodi najmanjših kvadratov: $x = \sum_{i=1}^n \frac{u_i^T b}{\sigma_i} v_i = A^+ b$.

Aproksimacija matrike z matriko nižjega ranga: $A_k = \sum_{i=1}^k \sigma_i u_i v_i^T$.