Znanstveno računanje 1

3. dio vježbi
Problemi svojstvenih vrijednosti i SVD

Nela Bosner

Problemi svojstvenih vrijednosti i SVD

- Problemi svojstvenih vrijednosti i dekompozicija singularnih vrijednosti (SVD) su srodni problemi.

Definicija

Neka je $A \in \mathbb{C}^{n \times n}$. Skalar $\lambda \in \mathbb{C}$ zove se **svojstvena vrijednost** *matrice A, ako postoji vektor $x \in \mathbb{C}^n$, $x \neq 0$ takav da je*

$$Ax = \lambda x.$$

Takav vektor x zove se **svojstveni vektor** *od A, koji pripada svojstvenoj vrijednosti λ.*

Definicija

Ukoliko za matricu $A = [a_1 \ \ldots \ a_n]$ možemo napisati da je $A = SDS^{-1}$, za neku regularnu matricu $S = [s_1 \ \ldots \ s_n]$, i $D = \mathrm{diag}(d_1, \ldots, d_n)$ dijagonalnu matricu tada vrijedi:

$$AS = SD \quad \Rightarrow \quad As_i = d_i s_i \ i = 1, \ldots, n.$$

*U tom slučaju dijagonalni elementi matrice D predstavljaju svojstvene vrijednosti matrice A, a stupci matrice S predstavljaju svojstvene vektore matrice A. Za rastav $A = SDS^{-1}$ kažemo da je **spektralna dekompozicija matrice** A.*

Napomena

Ako je $A \in \mathbb{R}^{n \times n}$ simetrična matrica, onda postoji ortogonalna matrica $U \in \mathbb{R}^{n \times n}$ i dijagonalna matrica $\Lambda = diag(\lambda_1, \ldots, \lambda_n)$, pri čemu su $\lambda_i \in \mathbb{R}$ za $i = 1, \ldots, n$, takve da je

$$A = U \Lambda U^T.$$

Teorem (Dekompozicija singularnih vrijednosti (SVD))

Neka je $A \in \mathbb{C}^{m\times n}$ matrica ranga r. Tada postoje unitarne matrice $U \in \mathbb{C}^{m\times m}$ i $V \in \mathbb{C}^{n\times n}$ takve da je

$$U^*AV = \begin{array}{cc} \begin{bmatrix} \Sigma_+ & 0 \\ 0 & 0 \end{bmatrix} & \begin{array}{l} r \\ m-r \end{array} \\ \begin{array}{cc} r & n-r \end{array} & \end{array}$$

gdje je $\Sigma_+ = diag(\sigma_1, \ldots, \sigma_r)$, uz $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$. (Tada je $A = U\Sigma V^*$.)

Definicija

Pozitivni skalari $\sigma_1, \ldots, \sigma_r$ zovu se **singularne vrijednosti**, *a stupci matrica U i V zovu se* **lijevi i desni singularni vektori** *matrice A.*

Napomena

Za matricu $A \in \mathbb{C}^{m\times n}$ ranga $r \leq \min(m, n)$, matrice $A^*A \in \mathbb{C}^{n\times n}$ i $AA^* \in \mathbb{C}^{m\times m}$ su simetrične i pozitivno semidefinitne. Vrijedi:

- $$V^*A^*AV = diag(\sigma_1^2, \ldots, \sigma_r^2, \underbrace{0, \ldots, 0}_{n-r}), \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$$

  tj, kvadrati singularnih vrijednosti matrice A su svojstvene vrijednosti matrice A^*A, samo što se među njima nalazi $n-r$ nula, a stupci matrice V su njeni svojstveni vektori.

Napomena (nastavak)

- $$U^* AA^* U = diag(\sigma_1^2, \ldots, \sigma_r^2, \underbrace{0, \ldots, 0}_{m-r}), \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$$

  tj, kvadrati singularnih vrijednosti matrice A su svojstvene vrijednosti matrice AA^*, samo što se među njima nalazi $m - r$ nula, a stupci matrice U su njeni svojstveni vektori.

Metode numeričkog računanja spektra

- Ako je $A \in \mathbb{R}^{n\times n}$ simetrična matrica, onda tražimo niz ortogonalnih matrica U_1, U_2,... takvih da
$$U_k^T \cdots U_2^T U_1^T A U_1 U_2 \cdots U_k \to \Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_n),$$
$$\text{kad } k \to \infty.$$
- Ovakvim transformacijama čuva se simetričnost matrice.

Jacobijeva metoda

- Ideja koja stoji iza Jacobijeve metode je sistematsko smanjivanje veličine
$$S(A) = \sqrt{\sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij}^2},$$
koju još nazivamo **normom vandijagonalnih elemenata.**
- Najprikladniji izbor za matrice U_i su **Jacobijeve rotacije** $R_i = R_i(p_i, q_i; \phi_i)$.
- Definirajmo niz simetričnih matrica $A^{(i)} \in \mathbb{R}^{n \times n}$, takav da je,
$$A^{(0)} = A, \qquad A^{(i)} = R_i^T A^{(i-1)} R_i.$$
- Za svaki i, biraju se
  - pivotni indeksi p_i i q_i, ovisno o **pivotnoj strategiji**,
  - kut ϕ_i, takav da poništi elemente sa indeksima p_i i q_i, tj
$$a_{p_i,q_i}^{(i)} = a_{q_i,p_i}^{(i)} = 0.$$

- Kada se raspiše djelovanje Jacobijevih rotacija u i-tom koraku, onda se vidi da se mijenjaju samo p_i-ti i q_i-ti stupac i redak. Odnosno

$$\begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & c_i & & -s_i & \\ & & & 1 & & \\ & & s_i & & c_i & \\ & & & & & 1 \end{bmatrix} \cdot \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix} \cdot \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & c_i & & s_i & \\ & & & 1 & & \\ & & -s_i & & c_i & \\ & & & & & 1 \end{bmatrix} = \begin{bmatrix} * & * & \bullet & * & \bullet & * \\ * & * & \bullet & * & \bullet & * \\ \bullet & \bullet & \bullet & \bullet & 0 & \bullet \\ * & * & \bullet & * & \bullet & * \\ \bullet & \bullet & 0 & \bullet & \bullet & \bullet \\ * & * & \bullet & * & \bullet & * \end{bmatrix}$$

pri čemu su sa $*$ označene stare vrijednosti, sa $\bullet$ nove, $c_i = \cos\phi_i$, i $s_i = \sin\phi_i$.

- Iz uvjeta $a^{(i)}_{p_i,q_i} = a^{(i)}_{q_i,p_i} = 0$ slijedi da su

$$c_i = \frac{1}{\sqrt{1+t_i^2}}, \qquad s_i = t_i c_i,$$

pri čemu su

$$t_i = \tan\phi_i = \frac{\operatorname{sign}(\tau_i)}{|\tau_i| + \sqrt{\tau_i^2+1}}$$

$$\tau_i = \frac{a^{(i-1)}_{q_i,q_i} - a^{(i-1)}_{p_i,p_i}}{2a^{(i-1)}_{p_i,q_i}}$$

- Još treba odabrati pivotnu strategiju. Postoje razne pivotne strategije, od kojih ćemo mi obraditi jednu.
- **Jacobijeva metoda sa cikličkom strategijom po recima** — ciklusi se sastoje od:

$$
\begin{aligned}
(p_i, q_i) = & (1,2),(1,3),\ldots,(1,n),(2,3),(2,4),\ldots,(2,n),\ldots \\
& \ldots,(n-2,n-1),(n-2,n),(n-1,n),
\end{aligned}
$$

odnosno u jednom ciklusu se poništavaju vandijagonalni elementi u sljedećem poretku:

$$
\begin{array}{|cccccc|}
\hline
* & 1 & 2 & 3 & 4 & 5 \\
1 & * & 6 & 7 & 8 & 9 \\
2 & 6 & * & 10 & 11 & 12 \\
3 & 7 & 10 & * & 13 & 14 \\
4 & 8 & 11 & 13 & * & 15 \\
5 & 9 & 12 & 14 & 15 & * \\
\hline
\end{array}.
$$

- Na kraju treba odrediti i uvjet zaustavljanja iteracija. Iteracije se zaustavljaju kada

$$S(A^{(i)}) \leq tol\|A\|_F,$$

za neku veličinu tolerancije $tol > 0$.

Algoritam (Jacobijeva metoda sa cikličkom pivotnom strategijom po recima)

tol zadan;
while $(S(A) > tol\|A\|_F)\{$
  for $(p = 0; p < n - 1; p + +)\{$
    for $(q = p + 1; q < n; q + +)\{$
      if $(A[p][q]! = 0)\{$
        $tau = (A[q][q] - A[p][p])/(2 * A[p][q]);$
        $t = sign(tau)/(fabs(tau) + sqrt(1 + pow(tau, 2)));$
        $c = 1/(sqrt(1 + pow(t, 2)));$
        $s = t * c;$
        $app = A[p][p]; apq = A[p][q]; aqq = A[q][q];$
        $app = app - t * apq;$
        $aqq = aqq + t * apq;$

Algoritam (nastavak)

$$
\begin{array}{l}
\quad\quad\quad\quad \textbf{for}\ (k = 0; k < n; k++)\{ \\
\quad\quad\quad\quad\quad pom = A[k][p]; \\
\quad\quad\quad\quad\quad A[k][p] = c * pom - s * A[k][q]; \\
\quad\quad\quad\quad\quad A[k][q] = s * pom + c * A[k][q]; \\
\quad\quad\quad\quad\quad A[p][k] = A[k][p];\ A[q][k] = A[k][q]; \\
\quad\quad\quad\quad \} \\
\quad\quad\quad\quad A[p][q] = 0;\ A[q][p] = 0; \\
\quad\quad\quad\quad A[p][p] = app;\ A[q][q] = aqq; \\
\quad\quad\quad \} \\
\quad\quad \} \\
\quad \} \\
\}
\end{array}
$$

Zadaci

Zadatak

Napišite potprogram `jacobi_sd()` *koji implementira Jacobijevu metodu za spektralnu dekompoziciju simetrične matrice. Ulazni parametri neka su*

- *dimenzija problema* n
- *matrica* A
- *tolerancija* tol

Kriterij zaustavljanja je $S(A^{(i)}) \leq tol\|A\|_F$. *Potprogram treba vratiti niz izračunatih svojstvenih vrijednosti.*

Zadatak

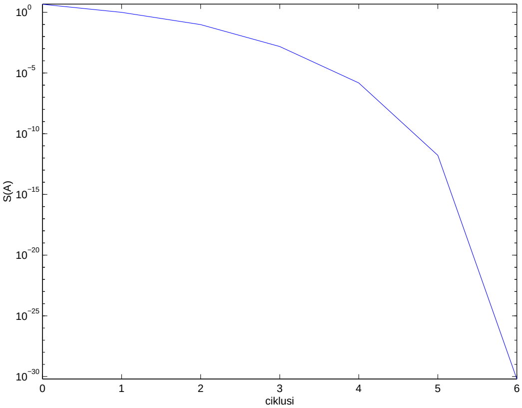
Svoj potprogram testirajte na 4×4 *simetričnoj matrici sa poznatim svojstvenim vrijednostima. Generirajte slučajnu ortogonalnu matricu* U*, dijagonalnu matricu* $D = diag(-10, -5, 0.1, 0.2)$*, a matricu* A *izračunajte kao* $A = UDU^T$*. Uzmite* $tol = 4 \cdot 10^{-16}$*.*

Zadatak

Testirajmo Jacobijevu metodu na primjeru Risove matrice. Risova matrica $A \in \mathbb{R}^{10\times 10}$ *je simetrična, i definira se kao*

$$A[i][j] = \frac{1}{2(8 - i - j + 1.5)}, \qquad i, j = 0, \ldots, 9.$$

Poznato je da svojstvene vrijednosti tvore nakupine oko $-\pi/2$ *i* $\pi/2$*. Neka je* $tol = 10^{-15}$*.*



Slika: Norme vandijagonalnih elemenata po ciklusima Jacobijeve metode sa cikličkom strategijom po recima, za matricu iz prethodnog zadatka.

Primjeri iz primjene

Za sljedeće primjere koristit ćemo CLAPACK-ove potprograme

- `dsyev_()` za računanje svih svojstvenih vrijednosti i svojstvenih vektora simetrične matrice.
  - Potprogram najprije svede matricu A na tridijagonalni oblik T, tako da je $A = QTQ^T$ pri čemu je Q ortogonalna.
  - Zatim se koristi brz algoritam za računanje spektralne dekompozicije tridijagonalne matrice T, tj. $T = P\Lambda P^T$.
  - Konačna spektralna dekompozicija od A je $A = (QP)\Lambda(QP)^T$.
  - Poziv potprograma je

```
int dsyev_(char *jobz, char *uplo,
integer *n, doublereal *a, integer *lda,
doublereal *w, doublereal *work, integer
*lwork, integer *info);
```

`jobz`	(ulaz) = ’N’: računa samo svojstvene vrijednosti = ’V’: računa svojstvene vrijednosti i svojstvene vektore
`uplo`	(ulaz) određuje dio matrice A koji je spremljen: = ’U’: gornji trokut = ’L’: donji trokut
`n`	(ulaz) red matrice A
`a`	(ulaz) matrica A; spremljen je samo njen gornji, ili donji trokut (ovisno o `uplo`) (izlaz) ako je `jobz` = ’V’ — matrica svojstvenih vektora ako je `jobz` = ’N’ — ulazni trokut je uništen

`lda`	(ulaz) vodeća dimenzija polja `a` ($lda \geq n$)
`w`	(izlaz) n polje koje sadrži svojstvene vrijednosti u rastućem poretku
`work`	pomoćno polje dimenzije ldw
`ldw`	(ulaz) vodeća dimenzija polja `work`, $ldw \geq 3n–1$
`info`	(izlaz) informacija o izvršavanju potprograma (0=OK)

- `dsyevx_()` za računanje nekih svojstvenih vrijednosti i odgovarajućih svojstvenih vektora simetrične matrice.
  - Potprogram najprije svede matricu A na tridijagonalni oblik T, tako da je $A = QTQ^T$ pri čemu je Q ortogonalna.
  - Zatim se koristi algoritam za računanje svojstvenih vrijednosti i svojstvenih vektora tridijagonalne matrice T.
  - Poziv potprograma je

```
int dsyevx_(char *jobz, char *range, char
*uplo, integer *n, doublereal *a, integer
*lda, doublereal *vl, doublereal *vu,
integer * il, integer *iu, doublereal
*abstol, integer *m, doublereal *w,
doublereal *z, integer *ldz, doublereal
*work, integer *lwork, integer *iwork,
integer *ifail, integer *info);
```

`jobz`	(ulaz) = ’N’: računa samo svojstvene vrijednosti = ’V’: računa svojstvene vrijednosti i svojstvene vektore
`range`	(ulaz) = ’A’: računa sve svojstvene vrijednosti = ’V’: računa svojstvene vrijednosti u $\langle vl, vu]$ = ’I’: računa od *il*-te do *iu*-te svojstvene vrijednosti u rastućem poretku
`uplo`	(ulaz) određuje dio matrice A koji je spremljen: = ’U’: gornji trokut = ’L’: donji trokut
`n`	(ulaz) red matrice A

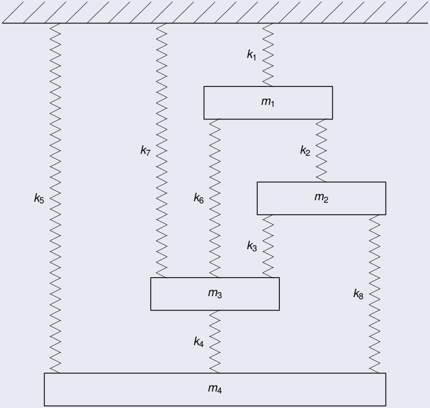
`a`	(ulaz) matrica A; spremljen je samo njen gornji ili donji trokut (ovisno o `uplo`) (izlaz) ulazni trokut je uništen
`lda`	(ulaz) vodeća dimenzija polja `a` ($lda \geq n$)
`vl`	(ulaz)
`vu`	(ulaz) $vl < vu$
`il`	(ulaz)
`iu`	(ulaz) $1 \leq il \leq iu \leq n$
`abstol`	(ulaz) Tolerancija na apsolutnu grešku u svojstvenim vrijednostima. Najbolje uzeti vrijednost `2*dlamch_('S')`
`m`	(izlaz) ukupan broj izračunatih svojstvenih vrijednosti, $0 \leq m \leq n$. ako je `range` = 'A' — $m = n$ ako je `range` = 'I' — $m = iu - il + 1$

`w`	(izlaz) n polje čijih prvih m elemenata sadrži tražene svojstvene vrijednosti u rastućem poretku
`z`	(izlaz) $ldz \times k$ polje ($k \geq m$) ako je `jobz` = 'V' i `info` = 0 — prvih m stupaca od Z sadrži ortonormalne svojstvene vektore od A, tako da je i-ti stupac jednak svojstvenom vektoru svojstvene vrijednosti $w[i]$
`ldz`	(ulaz) vodeća dimenzija polja `z` ($ldz \geq n$)
`work`	pomoćno polje dimenzije ldw
`ldw`	(ulaz) vodeća dimenzija polja `work`, $ldw \geq 8n$
`iwork`	pomoćno cjelobrojno polje dimenzije $5n$
`ifail`	(izlaz) ako je `jobz` = 'V' i `info` $>$ 0 — `ifail` sadrži indekse svojstvenih vektora koji nisu izkonvergirali
`info`	(izlaz) informacija o izvršavanju potprograma (0=OK)

Sistem masa s elastičnim oprugama

Primjer

Promatramo fizikalni sistem koji se sastoji od tijela različitih masa, povezanih elastičnim oprugama.



Primjer (nastavak)

Problem je pronaći slobodne oscilacije ovog sistema.

- *U ovom konkretnom primjeru imamo četiri tijela masa m_i $i = 1, 2, 3, 4$, i osam opruga krutosti k_l $l = 1, \ldots, 8$.*
- *Definirat ćemo sljedeće matrice:*

$$M = \begin{bmatrix} m_1 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 \\ 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & m_4 \end{bmatrix},$$

$$K = \begin{bmatrix} k_1 + k_2 + k_6 & -k_2 & -k_6 & 0 \\ -k_2 & k_2 + k_3 + k_8 & -k_3 & -k_8 \\ -k_6 & -k_3 & k_3 + k_4 + k_6 + k_7 & -k_4 \\ 0 & -k_8 & -k_4 & k_4 + k_5 + k_8 \end{bmatrix},$$

pri čemu je u matrici K prikazana interakcija među masama.

Primjer (nastavak)

- *i-ti redak odgovara i-toj masi, a njegov j-ti stupac odgovara odnosu i-te mase sa j-tom.*
- *Na svakom dijagonalnom elementu na poziciji (i, i) nalazi se zbroj krutosti svih opruga vezanih za i-tu masu.*
- *Na (i, j)-toj poziciji nalazi se $-k_l$ ukoliko l-ta opruga povezuje i-tu i j-tu masu, ili 0 ako te dvije mase nisu povezane oprugom.*

- *Na kraju, definirat ćemo vektor*

$$x = \begin{bmatrix} x[1] \\ x[2] \\ x[3] \\ x[4] \end{bmatrix},$$

kod kojeg $x[i]$ predstavlja vertikalni položaj i-te mase.

Primjer (nastavak)

- *Iz fizikalnih zakona, položaj masa može se opisati sistemom diferencijalnih jednadžbi*

$$\ddot{x} = -M^{-1}Kx.$$

- *Ako pretpostavimo da je rješenje oblika*

$$x = x_0 e^{i\phi t},$$

tada za drugu vremensku derivaciju imamo

$$\ddot{x} = -\phi^2 x_0 e^{i\phi t} = -M^{-1}Kx_0 e^{i\phi t}.$$

Sređivanjem dobivamo

$$M^{-1}Kx_0 = \phi^2 x_0,$$

Primjer (nastavak)

što se svodi na rješavanje problema traženja svojstvenih vrijednosti i svojstvenih vektora matrice $M^{-1}K$.

- *U ovom slučaju svojstvena vrijednost je oblika* ϕ^2.
- *Nadalje, možemo uočiti da je matrica* K *simetrična, ali produkt* $M^{-1}K$ *gubi to svojstvo.*
- *Matrica* M *je dijagonalna sa pozitivnom dijagonalom, zato je dobro definirana matrica* $M^{\frac{1}{2}} = diag(m_1^{\frac{1}{2}}, \ldots, m_4^{\frac{1}{2}})$.
- *Množenjem produkta* $M^{-1}K$ *matricom* $M^{\frac{1}{2}}$ *slijeva i matricom* $M^{-\frac{1}{2}}$ *zdesna, dobit ćemo matricu* A *koja je slična matrici* $M^{-1}K$.

Primjer (nastavak)

- *Matrica A je oblika*

$$A = M^{\frac{1}{2}}(M^{-1}K)M^{-\frac{1}{2}} = M^{-\frac{1}{2}}KM^{-\frac{1}{2}},$$

  što pokazuje da je ta matrica i simetrična, i imat će iste svojstvene vrijednosti kao i polazna matrica.

- *Pomnožimo li jednadžbu $M^{-1}Kx_0 = \phi^2 x_0$ sa $M^{\frac{1}{2}}$, dobit ćemo novi svojstveni problem*

$$\begin{aligned}(M^{-\frac{1}{2}}KM^{-\frac{1}{2}})(M^{\frac{1}{2}}x_0) &= \phi^2(M^{\frac{1}{2}}x_0)\\ Au &= \lambda u\end{aligned}$$

  pri čemu su

$$A = M^{-\frac{1}{2}}KM^{-\frac{1}{2}}, \quad u = M^{\frac{1}{2}}x_0, \quad \lambda = \phi^2.$$

Primjer (nastavak)

- *Neka su vrijednosti masa i krutosti opruga dane u sljedećim tablicama.*

i	1	2	3	4
$\mathbf{m_i}$	2	5	3	6

i	1	2	3	4	5	6	7	8
$\mathbf{k_i}$	10	9	8	7	6	5	5	5

- *Tada su matrice M i K dane sa*

$$M = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 6 \end{bmatrix}, \qquad K = \begin{bmatrix} 24 & -9 & -5 & 0 \\ -9 & 22 & -8 & -5 \\ -5 & -8 & 25 & -7 \\ 0 & -5 & -7 & 18 \end{bmatrix}.$$

Primjer (nastavak)

- *U našem primjeru je*

$$A = \begin{bmatrix} 12.0000 & -2.8460 & -2.0412 & 0 \\ -2.8460 & 4.4000 & -2.0656 & -0.9129 \\ -2.0412 & -2.0656 & 8.3333 & -1.6499 \\ 0 & -0.9129 & -1.6499 & 3.0000 \end{bmatrix},$$

  i mi tražimo svojstvene vrijednosti i vektore matrice A.
- *Problem ćemo riješiti pomoću potprograma* `dsyev_()`.
- *Trebamo naći matrice U i Λ, takve da je*

$$A \approx U \Lambda U^T,$$

  i iz dobivenih svojstvenih vrijednosti i svojstvenih vektora, trebamo izračunati 4 slobodne oscilacije x_i.

Primjer (nastavak)

- *Izračunate matrice bi trebale biti*

$$U = \begin{bmatrix} 0.9228 & 0.2354 & 0.1835 & -0.2438 \\ -0.2301 & 0.6226 & -0.4426 & -0.6029 \\ -0.3015 & 0.3894 & 0.8626 & -0.1161 \\ 0.0682 & 0.6367 & -0.1625 & 0.7507 \end{bmatrix},$$

$$\Lambda = \begin{bmatrix} 13.3767 & 0 & 0 & 0 \\ 0 & 1.0983 & 0 & 0 \\ 0 & 0 & 9.2700 & 0 \\ 0 & 0 & 0 & 3.9883 \end{bmatrix}.$$

- *Dakle, imamo* 4 *rješenja koja predstavljaju putanje slobodnih oscilacija, oblika*

$$x_i = M^{-\frac{1}{2}} u_i e^{i\sqrt{\lambda_i}t}, \qquad i = 1, 2, 3, 4.$$

Primjer (nastavak)

- *λ_i je i-ta svojstvena vrijednost od A, a u_i je njen svojstveni vektor, odnosno $\Lambda = diag(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$, i $U = [u_1 \ u_2 \ u_3 \ u_4]$.*
- *Na kraju dobivamo*

$$x_1 = \begin{bmatrix} 0.6525 \\ -0.1029 \\ -0.1741 \\ 0.0278 \end{bmatrix} e^{3.6574it}, \quad x_2 = \begin{bmatrix} 0.1298 \\ -0.1979 \\ 0.4980 \\ -0.0664 \end{bmatrix} e^{3.0447it},$$

$$x_3 = \begin{bmatrix} -0.1724 \\ -0.2696 \\ -0.0670 \\ 0.3065 \end{bmatrix} e^{1.9971it} \quad x_4 = \begin{bmatrix} 0.1665 \\ 0.2784 \\ 0.2248 \\ 0.2599 \end{bmatrix} e^{1.0480it}.$$

Particioniranje grafa

Primjer

Želimo podijeliti skup objekata u grupe koje sadrže objekte sa sličnim svojstvima.

- *Matematički ovaj problem je ekvivalentan particiji vrhova grafa.*
- *Particija grafa se izvodi tako da se minimiziraju težine bridova koji povezuju vrhove iz različitih grupa.*
- *Ovaj problem se dalje svodi na svojstveni problem.*
- *Za početak nam trebaju neke definicije.*

Definicija

Graf je uređeni par $G = (V, E)$, gdje je

- $\emptyset \neq V = V(G)$ skup vrhova,
- $E = E(G)$ je skup bridova disjunktan s V,
- a svaki brid $e \in E$ povezuje dva vrha $u, v \in V$ koje nazivamo krajevima od e.
- Vrhovi u i v su tada incidentni, i pišemo $e = \{u, v\}$.

Definicija

- Graf G je *konačan* ako su skupovi V i E konačni.
- Brid čiji krajevi se podudaraju naziva se *petlja*.
- Dva ili više bridova sa istim parom krajeva nazivaju se *višestruki bridovi*.

Definicija (nastavak)

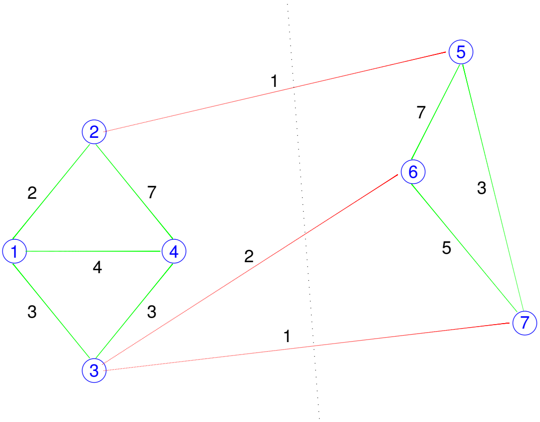
- *Graf je jednostavan ako ne sadrži petlje niti višestruke bridove.*
- *Jednostavan graf kod kojeg je svaki par vrhova povezan s jednim bridom naziva se potpuni graf.*
- *Neka je w funkcija $w : E(G) \to F$, gdje F može biti $\mathbb{R}$, $\mathbb{R}^+$, $\mathbb{Z}_m, \ldots$. Uređeni par (G, w) koji se sastoji od grafa G i težinske funkcije w naziva se težinski graf.*

Primjer (nastavak)

- *Neka je zadan jednostavan težinski graf* (G, w)*, gdje je* $G = (V, E)$*,*
- $\emptyset \neq V = \{1, 2, \ldots, n\}$ *je skup vrhova,*
- E *je skup bridova* $\{i, j\}$ $i, j \in V$*, sa težinama* $w(\{i, j\}) \in \mathbb{R}^+$*.*
- *Želimo podijeliti* V *u dva podskupa* V_1 *i* V_2*. Taj postupak nazivamo* ***biparticija****.*
- *Biparticija skupa* V *može se opisati relacijom*

$$V = V_1 \cup V_2, \qquad V_1 \cap V_2 = \emptyset.$$

- *Kako izvesti smislenu biparticiju tako da* V_1 *i* V_2 *predstavljaju grupe sa nekim zajedničkim svojstvima?*



Slika: Biparticija grafa G.

Definicija

Matrica susjedstva grafa G je $n \times n$ matrica $W = [w_{ij}]$, gdje je

$$w_{ij} = \begin{cases} w(\{i,j\}), & \text{ako } \{i,j\} \in E, \\ 0, & \text{inače.} \end{cases}$$

Napomena

- Matrica W je simetrična matrica čiji elementi su nenegativni realni brojevi.
- Budući da je graf jednostavan, dijagonalni elementi od W su jednaki 0.

Primjer (nastavak)

- *Razlika između dva skupa V_1 i V_2 može se mjeriti kao ukupna težina svih bridova koji povezuju ta dva skupa.*
- *Ta mjera se naziva* ***rez*** *particije, i definira se kao*

$$\mathrm{cut}(V_1, V_2) = \sum_{i \in V_1, j \in V_2} w_{ij} \quad \textit{za } V_1, V_2 \subset V.$$

- *Dalje, generaliziramo pojam težinske funkcije i definirajmo pojam* ***težine vrha*** $\mathbf{i} \in \mathbf{V}$*:*

$$w(i) = \sum_{j=1}^{n} w_{ij}.$$

- *Definirajmo još i pojam* ***težine skupa*** $\mathbf{V_k} \subseteq \mathbf{V}$*:*

$$w(V_k) = \sum_{i \in V_k} w(i) = \mathrm{cut}(V_k, V) = \sum_{i \in V_k, j \in V} w_{ij}.$$

Primjer (nastavak)

- *Za primjer grafa G sa slike:*
  - $V = \{1, 2, 3, 4, 5, 6, 7\}$
  - $V_1 = \{1, 2, 3, 4\}$
  - $V_2 = \{5, 6, 7\}$

  imamo

$$
\begin{aligned}
\mathrm{cut}(V_1, V_2) =& 4 \\
w(V_1) =& w(1) + w(2) + w(3) + w(4) = 42 \\
w(V_2) =& w(5) + w(6) + w(7) = 34
\end{aligned}
$$

- *Najjednostavniji način za biparticiju grafa je minimizacija reza, i za to postoje efikasni algoritmi.*
- *Međutim, kod takvih particija često se izoliraju blokovi sa malim brojem vrhova. To ponekad nije poželjno.*

Primjer (nastavak)

- *Ako želimo podskupove sa balansiranim težinama, trebamo minimizirati* ***normalizirani rez****:*

$$\mathrm{cut}_N(V_1, V_2) = \frac{\mathrm{cut}(V_1, V_2)}{w(V_1)} + \frac{\mathrm{cut}(V_2, V_1)}{w(V_2)},$$

- *Za graf G sa slike imamo*

$$\mathrm{cut}_N(V_1, V_2) = \frac{4}{42} + \frac{4}{32} = 0.2202.$$

- *Problem nalaženja egzaktnog minimalnog normaliziranog reza pripada NP klasi — vrijeme izvršavanja algoritma koji rješava taj probleme raste eksponencijalno sa veličinom ulaznih podataka.*

Primjer (nastavak)

- *Taj problem možemo riješiti aproksimativno prebacivanjem u realnu domenu.*
- *Ponovo polazimo od skupa vrhova* $V = \{1, 2, \ldots, n\}$.
- *Particija* $V = V_1 \cup V_2$ *može se reprezentirati vektorom* $x = [x_i]$, *koji je definiran sa*

$$x_i = \begin{cases} 1, & i \in V_1 \\ -1, & i \in V_2 \end{cases}, \qquad i = 1, \ldots, n.$$

- *Može se pokazati da za matricu* $D = \mathrm{diag}(w(1), \ldots, w(n))$ *vrijedi sljedeće*

$$\mathrm{cut}(V_1, V_2) = \frac{1}{4} x^T (D - W) x, ,$$

$$\mathrm{cut}_N(V_1, V_2) = \frac{z^T (D - W) z}{z^T D z},$$

Primjer (nastavak)

gdje je

$$z_i = \begin{cases} 1, & i \in V_1 \\ -q, & i \in V_2 \end{cases}, \qquad i = 1, \ldots, n, \quad q = \frac{w(V_1)}{w(V_2)}.$$

- *Matrica $L = D - W$ zove se* ***Laplaceova matrica grafa G*** *i ima mnogo korisnih svojstava.*
- *Matrica $L = [\ell_{ij}]$ je $n \times n$ matrica čiji svaki red i stupac odgovara jednom vrhu, tako da*

$$\ell_{ij} = \begin{cases} \displaystyle\sum_{k=1}^{n} w_{ik}, & i = j \\ -w_{ij}, & i \neq j,\ \{i,j\} \in E \\ 0, & \text{inače.} \end{cases}$$

Primjer (nastavak)

- *L je simetrična pozitivno semidefinitna matrica, tako da su joj sve svojstvene vrijednosti realne i nenegativne.*
- *Dalje, vrijedi*

$$Le = 0, \qquad za \qquad e = \begin{bmatrix} 1 & \ldots & 1 \end{bmatrix}^T,$$

što znači da je 0 je najmanja svojstvena vrijednost od L, a e je njen svojstveni vektor.

- *Sada možemo preformulirati problem. Započet ćemo sa problemom diskretne minimizacije*

$$\min_{\substack{V_1 \cup V_2 = V \\ V_1 \cap V_2 = \emptyset}} \mathrm{cut}(V_1, V_2) = \min_{\substack{x_i \in \{-1,1\} \\ x^T e = 0}} x^T L x$$

$$\min_{\substack{V_1 \cup V_2 = V \\ V_1 \cap V_2 = \emptyset}} \mathrm{cut}_N(V_1, V_2) = \min_{\substack{z_i \in \{-q,1\} \\ z^T D e = 0}} \frac{z^T L z}{z^T D z},$$

Primjer (nastavak)

- *Uvijet* $x^T e = 0$ *služi da se izbjegne trivijalno rješenje, i da se balansira broj vrhova u podskupovima.*
- *Za slučaj kada je* n *paran, uvijet* $x^T e = 0$ *znači da je* $\sum_{i=1}^{n} x_i = 0$*, i da skupovi* V_1 *i* V_2 *sadrže isti broj vrhova.*
- *Uvijet* $z^T De = 0$ *znači da je*

$$0 = \sum_{i=1}^{n} z_i w(i) = \sum_{i \in V_1} w(i) - q \sum_{i \in V_2} w(i) = w(V_1) - q w(V_2),$$

  što daje definiciju broja q*.*
- *Gornji diskretni problemi će se sada zamijeniti problemom kontinuirane minimizacije.*

Primjer (nastavak)

$$\min_{\substack{\|x\|_2=1\\ x^T e=0}} x^T L x$$

$$\min_{z^T D e=0} \frac{z^T L z}{z^T D z} = \min_{\substack{\|y\|_2=1\\ y^T D^{\frac{1}{2}} e=0}} y^T D^{-\frac{1}{2}} L D^{-\frac{1}{2}} y$$

gdje je $y = D^{\frac{1}{2}} z$.

- *Matrica* $L_N = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ *zove se* ***normalizirana Lapalaceova matrica grafa*** $\mathbf{G}$.
- L_N *je također simetrična pozitivno semidefinitna, čija je najmanja svojstvena vrijednost jednaka 0, sa svojstvenim vektorom* $D^{\frac{1}{2}} e$.

Primjer (nastavak)

- *Može se pokazati da se minimum prvog problema kontinuirane minimizacije postiže za $u_2 = [u_i^{(2)}]$, što predstavlja svojstveni vektor druge najmanje svojstvene vrijednosti od L.*
- *Taj vektor se naziva **Fiedlerov vektor**.*
- *Minimum drugog problema kontinuirane minimizacije postiže za $u_{N,2} = [u_i^{(N,2)}]$, što predstavlja svojstveni vektor druge najmanje svojstvene vrijednosti od L_N.*
- *Taj vektor se naziva **normalizirani Fiedlerov vektor**.*

Primjer (nastavak)

- *Na kraju, za aproksimativno rješenje optimalne biparticije možemo uzeti*

  za minimizaciju cut

$$V_1 = \{i : u_i^{(2)} \geq 0\}, \qquad V_2 = \{i : u_i^{(2)} < 0\}$$

  za minimizaciju cut_N

$$V_1 = \{i : D^{-\frac{1}{2}} u_i^{(N,2)} \geq 0\}, \quad V_2 = \{i : D^{-\frac{1}{2}} u_i^{(N,2)} < 0\}.$$

- *Problem grafa G sa slike ćemo riješiti pomoću potprograma* `dsyevx_()`*.*
- *Nakon što izračunamo Fiedlerov vektor i normalizirani Fiedlerov vektor, trebaju se odrediti podskupovi V_1 i V_2 dobiveni za oba slučaja.*

Primjer (nastavak)

- *Izračunati Fiedlerovi vektori bi trebali biti*

$$u_2 = \begin{bmatrix} -0.38326 \\ -0.37308 \\ -0.14519 \\ -0.38009 \\ 0.42371 \\ 0.40851 \\ 0.44941 \end{bmatrix}, \quad u_{N,2} = \begin{bmatrix} -0.34096 \\ -0.36254 \\ -0.13241 \\ -0.45047 \\ 0.41428 \\ 0.46191 \\ 0.38324 \end{bmatrix}.$$

- *Odgovarajuće svojstvene vrijednosti su*

$$\lambda_2 = 1.9497, \qquad \lambda_{N,2} = 0.17559.$$

- *U oba slučaja ispada*

$$V_1 = \{5, 6, 7\}, \quad V_2 = \{1, 2, 3, 4\}.$$

SVD

Za sljedeće primjere koristit ćamo CLAPACK-ov potprogram `dgesvd_()`.

- `dgesvd_()` računa dekompoziciju singularnih vrijednosti općenite pravokutne matrice.
  - Potprogram najprije svede matricu A na bidijagonalni oblik B, tako da je $A = WBY^T$ pri čemu su W i Y ortogonalne.
  - Zatim se koristi brz algoritam za računanje SVD-a bidijagonalne matrice B, tj. $B = S\Sigma Z^T$.
  - Konačni SVD od A je $A = (WS)\Sigma(YZ)^T$.
  - Poziv potprograma je

```
int dgesvd_(char *jobu, char *jobvt,
integer *m, integer *n, doublereal *a,
integer *lda, doublereal *s, doublereal
*u, integer * ldu, doublereal *vt,
integer *ldvt, doublereal *work, integer
*lwork, integer *info);
```

`jobu` (ulaz)
= 'A': svih m stupaca od U su vraćeni u polju `u`
= 'S': prvih $\min\{m, n\}$ stupaca od U su vraćeni u polju `u`
= 'O': prvih $\min\{m, n\}$ stupaca od U su vraćeni u polju `a`
= 'N': stupci od U nisu izračunati

`jobvt` (ulaz)
= 'A': svih n redaka od V^T su vraćeni u polju `vt`
= 'S': prvih $\min\{m, n\}$ redaka od V^T su vraćeni u polju `vt`
= 'O': prvih $\min\{m, n\}$ redaka od V^T su vraćeni u polju `a`
= 'N': reci od V^T nisu izračunati

`m`	(ulaz) broj redaka matrice A
`n`	(ulaz) broj stupaca matrice A
`a`	(ulaz) matrica A; (izlaz) ako je `jobu` = 'O' — prvih $\min\{m, n\}$ stupaca od U ako je `jobvt` = 'O' — prvih $\min\{m, n\}$ redaka od V^T ako je `jobu` $\neq$ 'O' i `jobvt` $\neq$ 'O' — sadržaj od A je uništen
`lda`	(ulaz) vodeća dimenzija polja `a` ($lda \geq m$)
`s`	(izlaz) $\min\{m, n\}$ polje koje sadrži singularne vrijednosti u padajućem poretku

`u`	(izlaz) ako je `jobu` = ’A’ — $m \times m$ ortogonalna matrica U ako je `jobu` = ’S’ — prvih $\min\{m, n\}$ stupaca od U ako je `jobu` = ’N’ ili ’O’, `u` se ne referencira
`ldu`	(ulaz) vodeća dimenzija polja `u` ($ldu \geq m$)
`vt`	(izlaz) ako je `jobvt` = ’A’ — $n \times n$ ortogonalna matrica V^T ako je `jobvt` = ’S’ — prvih $\min\{m, n\}$ redaka od V^T ako je `jobvt` = ’N’ ili ’O’, `vt` se ne referencira
`ldvt`	(ulaz) vodeća dimenzija polja `vt` ($ldvt \geq n$)
`work`	pomoćno polje dimenzije ldw
`ldw`	(ulaz) vodeća dimenzija polja `work`, $ldw \geq 5\max\{m, n\}$
`info`	(izlaz) informacija o izvršavanju potprograma

Primjeri iz primjene

SVD ima široku primjenu:

- računanje inverza regularne kvadratne matrice
- računanje generaliziranog inverza pravokutne matrice
- računanje uvjetovanosti matrice
- rješavanje ortogonalnog Procrustes problema
- nalaženje presjeka jezgara dvaju linearnih operatora
- nalaženje kuteva između dva potprostora
- nalaženje presjeka potprostora
- rješavanje linearnog problema najmanjih kvadrata
- rješavanje linearnog problema totalnih najmanjih kvadrata
- rješavanje integralnih jednadžbi (geofizika)
- procesiranje slika
- modeliranje prometa na internetu
- genetika (obrnuti inženjering genske mreže)

Rješavanje ortogonalnog Procrustes problema

Primjer

- *Ovaj problem se npr. pojavljuje u psihometriji (znanost o mjerenju mentalnih sposobnosti i procesa, tj. predočavanje psihičkih pojava matematičkim sredstvima).*
- *Pretpostavimo da je $A \in \mathbb{R}^{m \times n}$ matrica podataka dobivenih izvođenjem određenog skupa eksperimenata.*
- *Ako se isti skup eksperimenata ponovo izvede, dobiva se druga matrica podataka $B \in \mathbb{R}^{m \times n}$.*
- *U ortogonalnom Procrustes problemu istražujemo mogućnost da li se podaci u A mogu zarotirati tako da dobijemo podatke u B (čuvaju se međusobni kutevi i udaljenosti).*

Primjer (nastavak)

- *Zbog grešaka u podacima taj problem vjerojatno nije moguće riješiti egzaktno.*
- *Zato tražimo rješenje problema*

$$\min_{Q \in \mathbb{R}^{n\times n},\ Q^T Q = I_n} \|AQ - B\|_F.$$

- *Primijetimo da ako je $Q \in \mathbb{R}^{n\times n}$ ortogonalan, tada je*

$$\min \|AQ - B\|_F^2 = \min(\mathrm{trag}(A^T A) + \mathrm{trag}(B^T B) - 2\,\mathrm{trag}(Q^T A^T B)).$$

- *Ortogonalni Procrustes problem je, prema tome, ekvivalentan problemu maksimiziranja izraza* $\mathrm{trag}(Q^T A^T B))$*.*
- *Maksimizirajući Q se može naći računanjem SVD-a matrice $A^T B$.*

Primjer (nastavak)

- *Ako je*

$$U^T(A^TB)V = \Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$$

  SVD matrice A^TB i ako definiramo ortogonalnu matricu $Z = V^TQ^TU$,

- *tada je*

$$\mathrm{trag}(Q^TA^TB) = \mathrm{trag}(Q^TU\Sigma V^T) = \mathrm{trag}(Z\Sigma) = \sum_{i=1}^{n} z_{ii}\sigma_i \leq \sum_{i=1}^{n} \sigma_i.$$

- *Jasno se vidi da se gornja ograda postiže za $Z = I$, pa je tada $Q = UV^T$.*

Algoritam (Rješavanje ortogonalnog Procrustes problema)

1. $C = A^T B$
2. *Izračunaj SVD* $U^T C V = \Sigma$.
3. $Q = U V^T$.

Primjer (nastavak)

- *Mi ćemo uzeti sljedeće konkretne podatke*

$$A = \begin{bmatrix} 1.2 & 2.1 \\ 2.9 & 4.3 \\ 5.2 & 6.1 \\ 6.8 & 8.1 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{bmatrix}.$$

- *Konkretan problem rješavat ćemo pomoću gornjeg algoritma i potprograma* `dgesvd_()`.

Primjer (nastavak)

- *Osim rješenja Q nađite i minimalnu normu za $\|AQ - B\|_F$.*
- *Izračunata matrica Q bi trebala biti*

$$Q = \begin{bmatrix} 0.9999 & -0.0126 \\ 0.0126 & 0.9999 \end{bmatrix},$$

- *a minimalna vrijednost*

$$\|AQ - B\|_F = 0.4661.$$

Nalaženje kuteva između dva potprostora

Primjer

- *Neka su $\mathcal{X}$ i $\mathcal{Y}$ potprostori od $\mathbb{R}^m$, čije dimenzije zadovoljavaju*

$$p = \dim(\mathcal{X}) \geq \dim(\mathcal{Y}) = q \geq 1.$$

- ***Glavni kutevi*** *$\theta_1, \ldots, \theta_q \in [0, \pi/2]$ između $\mathcal{X}$ i $\mathcal{Y}$ definiraju se rekurzivno sa*

$$\cos(\theta_k) = \max_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} x^T y = x_k^T y_k$$

tako da:

$$\begin{array}{ll} \|x\|_2 = \|y\|_2 = 1 & \\ x^T x_i = 0 & i = 1, \ldots, k-1 \\ y^T y_i = 0 & i = 1, \ldots, k-1 \end{array}$$

Primjer (nastavak)

- *Vektori $\{x_1, \ldots, x_q\}$ i $\{y_1, \ldots, y_q\}$ zovu se **glavni vektori** između potprostora $\mathcal{X}$ i $\mathcal{Y}$.*
- *Za glavne kuteve vrijedi*

$$0 \leq \theta_1 \leq \cdots \leq \theta_q \leq \pi/2.$$

- *Ako je $p = q$ tada je*

$$\mathrm{dist}(\mathcal{X}, \mathcal{Y}) = \|P_{\mathcal{X}} - P_{\mathcal{Y}}\|_2 = \sqrt{1 - \cos(\theta_p)^2} = \sin(\theta_p),$$

udaljenost između potprostora jednakih dimenzija, gdje su $P_{\mathcal{X}}$ i $P_{\mathcal{Y}}$ ortogonalne projekcije na $\mathcal{X}$ i $\mathcal{Y}$.

Primjer (nastavak)

- *U tom slučaju kut $\angle(\mathcal{X},\mathcal{Y})$ može se definirati kao*

$$\begin{aligned}\angle(\mathcal{X},\mathcal{Y}) &= \arcsin(\mathrm{dist}(\mathcal{X},\mathcal{Y})) = \theta_p\\ &= \max_{x\in\mathcal{X}} \angle(x,\mathcal{Y}) = \max_{x\in\mathcal{X}} \min_{y\in\mathcal{Y}} \angle(x,y)\\ &= \max_{y\in\mathcal{Y}} \angle(y,\mathcal{X}) = \max_{y\in\mathcal{Y}} \min_{x\in\mathcal{X}} \angle(x,y).\end{aligned}$$

- *Štoviše*

$$\angle(x,\mathcal{Y}) = \min_{y\in\mathcal{Y}} \angle(x,y) = \min_{y\in\mathcal{Y}} \arccos\left(\frac{x^T y}{\|x\|_2\|y\|_2}\right),$$

pri čemu svi vektori x i y moraju biti različiti od nule.

Primjer (nastavak)

- *Ako stupci od $X \in \mathbb{R}^{m\times p}$ i $Y \in \mathbb{R}^{m\times q}$ definiraju ortnormirane baze za $\mathcal{X}$ i $\mathcal{Y}$, tada*

$$\max_{\substack{x\in\mathcal{X}\\ \|x\|_2=1}} \max_{\substack{y\in\mathcal{Y}\\ \|y\|_2=1}} x^T y = \max_{\substack{u\in\mathbb{R}^p\\ \|u\|_2=1}} \max_{\substack{v\in\mathbb{R}^q\\ \|v\|_2=1}} u^T(X^T Y)v.$$

- *Slijedi, da ako je*

$$U^T(X^T Y)V = \mathrm{diag}(\sigma_1, \ldots, \sigma_q), \quad sa \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_q$$

SVD od $X^T Y$, tada možemo definirati x_k, y_k i θ_k sa

$$\begin{aligned} \begin{bmatrix} x_1 & \ldots & x_q \end{bmatrix} &= XU, \\ \begin{bmatrix} y_1 & \ldots & y_q \end{bmatrix} &= YV, \\ \cos(\theta_k) &= \sigma_k, \qquad k = 1, \ldots, q. \end{aligned}$$

Primjer (nastavak)

- *Obično potprostori $\mathcal{X}$ i $\mathcal{Y}$ su definirani kao slike matrica $A \in \mathbb{R}^{m \times p}$ i $B \in \mathbb{R}^{m \times q}$ koje ne moraju biti ortonormirane.*
- *U tom slučaju tražene ortonormirane baze mogu se dobiti QR faktorizacijom matrica A i B.*

Algoritam (Računanje glavnih kutova i glavnih vektora između dva potprostora)

1. *Izračunaj QR faktorizaciju* $A = Q_A R_A$: $Q_A \in \mathbb{R}^{m\times p}$, $Q_A^T Q_A = I_p$, $R_A \in \mathbb{R}^{p\times p}$.
2. *Izračunaj QR faktorizaciju* $B = Q_B R_B$: $Q_B \in \mathbb{R}^{m\times q}$, $Q_B^T Q_B = I_q$, $R_B \in \mathbb{R}^{q\times q}$.
3. $C = Q_A^T Q_B$.
4. *Izračunaj SVD* $U^T C V = \mathrm{diag}(\cos(\theta_k))$.
5. $[x_1, \ldots, x_q] = Q_A U(:, 1 : q)$.
6. $[y_1, \ldots, y_q] = Q_B V$.

Primjer (nastavak)

- *Mi ćemo računati glavne kuteve i glavne vektore izmedju dvije 2D ravnine u* $\mathbb{R}^4$ *pomoću gornjeg algoritma i potprograma* `dgesvd_()`.
- *Ravnina* $\mathcal{X}$ *neka je razapeta vektorima* $(1, 0, 0, 0)$ *i* $(1, 1, 1, 1)$.
- *Ravnina* $\mathcal{Y}$ *neka je razapeta vektorima* $(1, -1, 1, -1)$ *i* $(0, 1, 0, 1)$.
- *Koliki je kut* $\angle(\mathcal{X}, \mathcal{Y})$*?*

Primjer (nastavak)

- *Kosinusi glavnih kuteva bi trebali biti*

$$\cos(\theta_1) = 1, \qquad \cos(\theta_2) = 0.5774.$$

- *Glavni vektori bi trebali biti*

$$[\ x_1 \quad x_2\ ] = \begin{bmatrix} -0.5000 & -0.8660 \\ -0.5000 & 0.2887 \\ -0.5000 & 0.2887 \\ -0.5000 & 0.2887 \end{bmatrix}, \quad [\ y_1 \quad y_2\ ] = \begin{bmatrix} -0.5000 & -0.5000 \\ -0.5000 & 0.5000 \\ -0.5000 & -0.5000 \\ -0.5000 & 0.5000 \end{bmatrix}.$$

- *Kut između ravnina iznosi*

$$\angle(\mathcal{X}, \mathcal{Y}) = 0.9553 = 54^\circ 44'$$

Nalaženje presjeka potprostora

Primjer

Isti postupak kao u prethodnom primjeru može se koristiti za računanje ortogonalne baze za $\mathrm{Im}(A) \cap \mathrm{Im}(B)$*, gdje su* $A \in \mathbb{R}^{m\times p}$ *i* $B \in \mathbb{R}^{m\times q}$.

Teorem

Neka su $\cos(\theta_k)$ *za* $k = 1, \ldots, q$, $\begin{bmatrix} x_1 & \cdots & x_q \end{bmatrix}$ *i* $\begin{bmatrix} y_1 & \cdots & y_q \end{bmatrix}$ *definirani kao u prethodnom primjeru. Ako indeks s definiramo sa* $1 = \cos(\theta_1) = \cdots = \cos(\theta_s) > \cos(\theta_{s+1})$*, tada vrijedi*

$$\mathrm{Im}(A) \cap \mathrm{Im}(B) = [\{x_1, \ldots, x_s\}] = [\{y_1, \ldots, y_s\}].$$

Primjer (nastavak)

- *Koja je baza presjeka ravnina iz prethodnog zadatka?*

Procesiranje slika

Primjer

- *Želimo nacrtati plohu u 3D grafici.*
- *Pretpostavimo da je ploha definirana kao graf sljedeće funkcije*

$$f(x,y) = \frac{1}{250}(x^2y - x^2 - y^2 + 175) \qquad (x,y) \in [-5,5] \times [-5,5].$$

- *Budući da je računalna grafika diskretna, prvo moramo definirati mrežu na kvadratu* $[-5,5] \times [-5,5]$*, i onda crtamo točke koje predstavljaju vrijednost funkcije u svakom čvoru mreže.*
- *Početni kvadrat možemo podijeliti na male* 0.5×0.5 *kvadrate, proizvevši ukupno* $21 \times 21 = 441$ *točaka mreže.*

Primjer (nastavak)

- *Točke mreže označimo sa*

$$(x_i, y_j), \quad i, j = 0, \ldots, 20, \quad x_i = -5+0.5i,\ y_j = -5+0.5j.$$

- *Te točke dalje organiziramo u* 21×21 *matricu* $A = [a_{ij}]$, *gdje su*

$$a_{ij} = f(x_i, y_j),$$

koja se dalje koristi za crtanje.

- *Prema tome moramo spremiti 441 elementa.*
- *Pomoću potprograma* `dgesvd_()` *izračunajte SVD matrice* A.
- *Koliko je netrivijalnih singularnih vrijednosti od* A, *tj. koji je* $r = r(A)$?

Teorem

Neka je dan SVD od $A \in \mathbb{R}^{m\times n}$. Ako je $k < r = r(A)$ i

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T = U(1:m,1:k)\Sigma(1:k,1:k)V(1:n,1:k)^T$$

tada, za svaku unitarno invarijantnu normu $\|\cdot\|$ vrijedi

$$\min_{r(B)\le k} \|A-B\| = \|A-A_k\|,$$

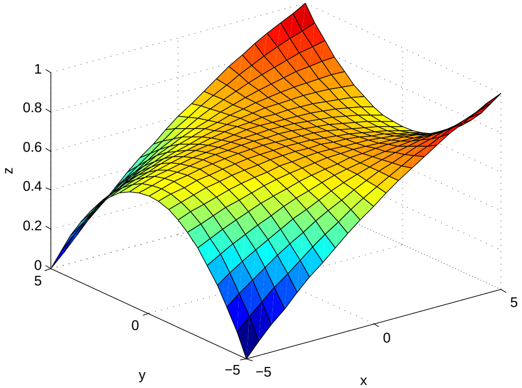
a posebno

$$\min_{r(B)\le k} \|A-B\|_2 = \|A-A_k\|_2 = \sigma_{k+1},$$

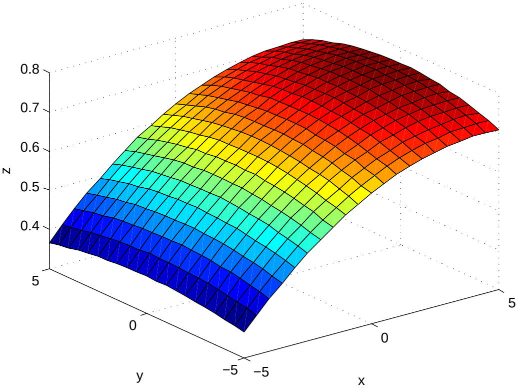
$$\min_{r(B)\le k} \|A-B\|_F = \|A-A_k\|_F = \sqrt{\sum_{i=k+1}^{r} \sigma_i^2}.$$

Primjer (nastavak)

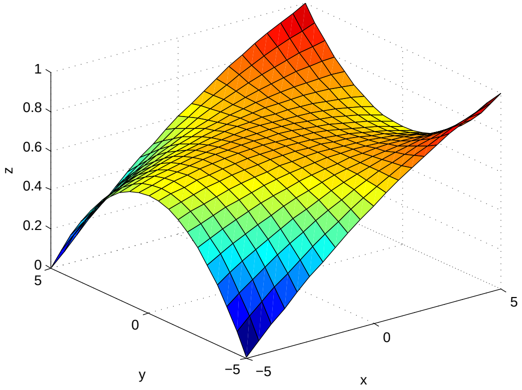
- *Izračunajte* A_k *za* $k = 1, \ldots, r$ *i usporedite* $\|A - A_k\|_F$.
- *Što možete zaključiti?*
- *Za spremanje matrice* A_k *dovoljno je umjesto* $m \cdot n$ *elemenata spremiti* $(m + n + 1) \cdot k$ *elemenata od* $U(1 : m, 1 : k)$, $V(1 : n, 1 : k)$ *i* $\Sigma(1 : k, 1 : k)$, *što može biti velika ušteda.*



Slika: 3D prikaz funkcije $f(x, y) = \frac{1}{250}(x^2 y - x^2 - y^2 + 175)$.



Slika: Za aproksimaciju ranga 1 A_1 potrebno je spremiti 43 elemenata, što predstavlja 9.75% originalne količine memorije. Prikaz nije precizan.



Slika: Za aproksimaciju ranga 2 A_2 potrebno je spremiti 86 elemenata, što predstavlja 19.50% originalne količine memorije. Prikaz je identičan originalu.