# Metode redukcije dimenzije visokodimenzionalnih podataka

**Prusina, Tomislav**

Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni preddiplomski studij matematike i računarstva

Tomislav Prusina

# Metode redukcije dimenzije visokodimenzionalnih podataka

Završni rad

Osijek, 2020.

Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni preddiplomski studij matematike i računarstva

Tomislav Prusina

# Metode redukcije dimenzije visokodimenzionalnih podataka

Završni rad

Mentor: izv. prof. dr. sc. Domagoj Matijević

Osijek, 2020.

# Sažetak

U ovome radu ćemo promatrati podatke u visokodimenzionalnim prostorima. Analiza podataka u takvim prostorima može biti računski i vremenski zahtjevna. Stoga uvodimo pojam redukcije (smanjenja) dimenzije kako bismo mogli lakše i brže provesti analizu te ćemo promatrati sljedeće metode za redukcije dimenzije:

- slučajna projekcija,
- analiza glavnih komponenti,
- skaliranje najmanjih kvadrata.

Za svaku metodu ćemo navesti prednosti i nedostatke te ćemo za metodu skaliranja najmanjih kvadrata dodatno proučiti algoritam pod nazivom "skaliranje majorizacijom kompliciranе funkcije".



# Ključne riječi

Redukcija dimezije, slučajna projekcija, skaliranje najmanjih kvadrata, Sammon-ovo mapiranje, metoda glavnih komponenti, skaliranje majorizacijom kompliciranе funkcije.



# Abstract

In this paper, we will observe data in high-dimensional spaces. Data analysis in such spaces can be a computational and time demanding. Therefore, we introduce the concept of dimensionality reduction so that we can more easily and quickly perform data analysis. We will observe the following three methods for reducing dimensions:

- random projection,
- principal component analysis,
- least-squares scaling.

For each method we will list the advantages. Additionaly, for the least-squares scaling method we will study the algorithm: scaling by majorizing a complicated function.



# Key words

Dimensionality reduction, randomized projection, least-squares scaling, Sammon mapping, principal component analisys, scaling by majorizing a complicated function.

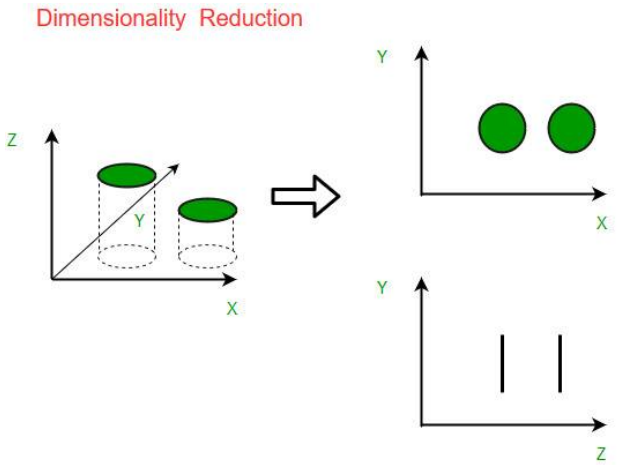
# Sadržaj

# Uvod

U problemima strojnog učenja ponekad imamo previše faktora na temelju kojih se vrši klasifikacija podataka. Ti su faktori u osnovi poznati kao **varijable** (ili, **značajke**). Što je veći broj varijabli, to je teže vizualizirati skup trening podataka i raditi na njemu. Ponekad je većina ovih varijabli povezana, a samim tim i suvišna. Također, računanje s velikim brojem varijabli može biti zahtjevno te u tu svrhu uvodimo pojam **smanjenja** ili **redukcije dimenzije**. Pod pojmom redukcije dimenzije podrazumijevamo transformaciju podataka iz visokodimenzionalnog prostora u prostor niže dimenzije tako da ostane sačuvan što veći broj informacija o podacima. Reducirati dimenziju podataka možemo učiniti na dva načina. Prvi način je odabrati podskup varijabli, iz skupa varijabli originalnih podataka, pomoću kojih ćemo prikazati originalne podatke, dok je drugi način koristiti neku od metoda za redukciju dimenzije. Od metoda za redukciju dimenzije podataka najpoznatije su:

- analiza glavnih komponenti (eng. principal component analysis),
- slučajna projekcija (eng. random projection),
- Sammon-ovo mapiranje (eng. Sammon mapping),
- Skaliranje majorizacijom komplicirane funkcije (eng. scaling by majorizing a complicated funciton).

Sljedeća slika prikazuje primjer redukcije dimenzije, odnosno kako podatke iz trodimenzionalnog prostora možemo prikazati u dvodimenzionalnom prostoru:




Slika 1: Primjer redukcije dimenzije.

# 1 Klasični MDS

## 1.1 Analiza glavnih komponenti

Analiza glavnih komponenti (ili, kraće, PCA) je tehnika koja se često koristi u praksi za smanjenje dimenzije, kompresiju podataka, prepoznavanje značajki i prikazivanje podataka. PCA je još poznata i pod nazivom *Karhunen-Loève* transformacija.

Najčešće su dvije korištene definicije PCA tehnike iz kojih proizlazi isti algoritam. Jedna definicija je da je PCA pronalazak ortogonalnog projektora koji će projicirati podatke u niže-dimenzionalni prostor, poznat kao principijalni potprostor, i očuvati što više varijance među projiciranim podacima. Druga definicija PCA je pronalazak linearnog projektora koji minimizira grešku projiciranih podataka, definiranu kao suma kvadrata udaljenosti originalnih i projiciranih podataka. Ova tehnika je bazirana na sljedećem vrlo važnom teoremu.

**Teorem 1.1** (Eckart-Young-Mirsky teorem)
*Neka je $X \in \mathbb{R}^{m\times n}$ centrirana matrica čiji je rang $r(X)$ te je pripadna SVD dekompozicija (vidi [10]) matrice*

$$X = U\Sigma V^T,$$

*pri čemu je $\Sigma$ dijagonalna matrica*

$$\Sigma = diag(\sigma_1, \ldots, \sigma_r, 0, \ldots, 0) = \begin{bmatrix} \sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_r & & \\ & & & 0 & \\ & & & & \ddots \end{bmatrix}$$

*sa singularnim vrijednostima na dijagonali $\sigma_1 \geq \cdots \geq \sigma_r \geq 0$, a matrice $U$ i $V$ ortogonalne matrice. Tada je*

$$\hat{X} = US_kV^T$$

*rješenje optimizacijskog problema*

$$\min_{r(Y)=k} \|X - Y\|_F, \tag{1}$$

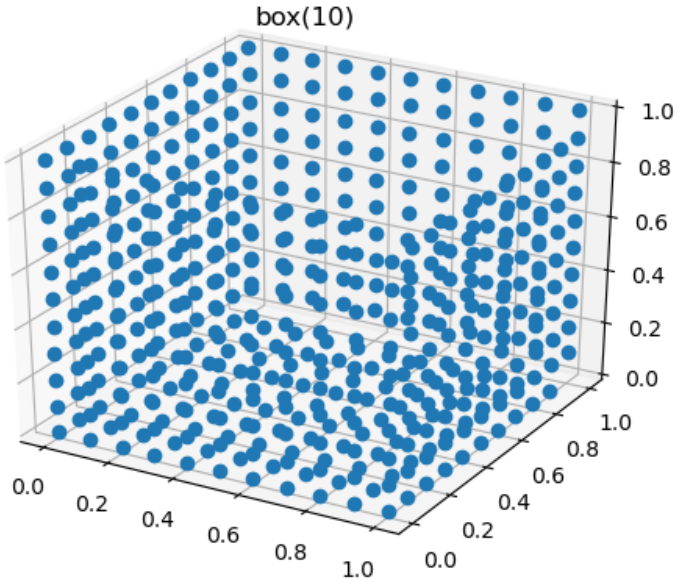
*gdje je matrica $S_k$ dijagonalna matrica sa prvih $k$ najvećih singularnih vrijednosti matrice $\Sigma$ na dijagonali, odnosno*

$$S_k = diag(\sigma_1, \ldots, \sigma_k, 0, \ldots, 0) = \begin{bmatrix} \sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_k & & \\ & & & 0 & \\ & & & & \ddots \end{bmatrix}.$$

Teorem navodimo bez dokaza, no ukoliko čitatelj želi vidjeti dokaz može ga pronaći u [5]. Uočimo da ako uzmemo matricu $V_k \in \mathbb{R}^{n\times k}$, čiji su stupci prvih $k$ stupaca matrice $V$, onda rješenje $US_kV^T$ minimizacijskog problema iz teorema možemo zapisat kao $US_kV^T = \hat{X}V_kV_k^T$. Tim zapisom dobija se linearno mapiranje $V_k^T$ koje će podatak $x_i$ iz originalne dimenzije $n$ mapirati u podatak $V_k^Tx_i$ u prostor niže dimenzije $k$.

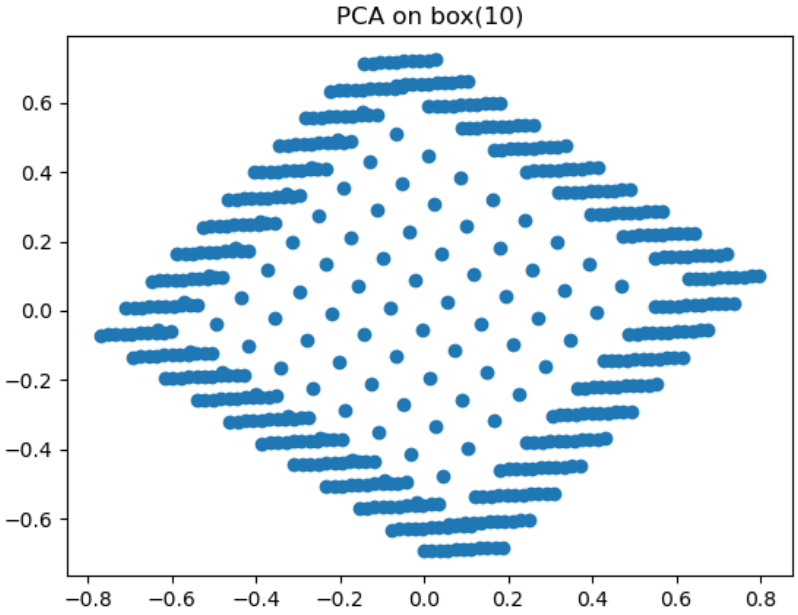
Ako pak želimo vratiti podatak u originalnu dimenziju, možemo to učinit opet matricom $V_k$ kao $V_k V_k^T x_i$.

U svrhu lakšeg razumijevanja redukcije dimenzije, koristit ćemo umjetno generirani skup podataka u prostoru $\mathbb{R}^3$. Sljedeća slika prikazuje umjetno generirani skup točaka koji predstavlja otvorenu kutiju.




Slika 2: Skup podataka: otvorena kutija.

Iz slike 2 možemo uočiti da su podaci u obliku otvorene kutije, odnosno imamo 5 stranica, svaka stranica otvorene kutije sadržava 10×10 podjednako raspoređenih točaka, što znači da taj skup ukupno sadrži 500 različitih podataka. Primjenom PCA tehnike na dani skup podataka, odnosno kao rješenje optimizacijskog problema (1) dobivamo skup podataka $Y$ oblika:




Slika 3: PCA na otvorenoj kutiji.

Slika 3 predočava što to zapravo znači reducirati dimenziju tehnikom PCA. Dakle, pronalazimo svojstvene vektore smjera u kojem je raspršenost (ili, varijanca) podataka najmanja te brisanjem tih vektora "spljošti" podatke u nižu dimenziju. U slučaju slike 2, podatke smo "spljoštili" u smjeru od otvora prema dnu kutije i time dobili prikaz koji predstavlja slika 3.

### 1.1.1 Metoda glavnih koordinata

Kao ekvivalent metodi PCA, metoda glavnih koordinata, ili, kraće, PCO (eng. principal coordinate analisys), pronalazi linearno mapiranje podataka u nižu dimenziju koristeći samo matricu međusobnih euklidskih udaljenost. Ukoliko nam je zadana matrica udaljenosti $D$ nepoznatih podataka $X$, ovom metodom uz malu transformaciju $D$ možemo dobiti matricu $\hat{X}$ čije međusobne udaljenosti podataka odgovaraju udaljenostima iz dane matrice i kao posljedicu dobiti jednu metodu za sniženje dimenzije. U sljedećoj propoziciji ćemo pokazakati koja je to transformacija koja daje željene podatke.

**Propozicija 1.1**
*Neka je dana matrica udaljenosti $D \in \mathbb{R}^{m\times m}$ nepoznatih podataka. Tada postoji matrica podataka $X \in \mathbb{R}^{m\times q}$, $q < m$, čija je matrica udaljenosti jednaka $D$ i vrijedi*

$$XX^T = -\frac{1}{2}HD^{[2]}H,$$

*gdje je $H$ matrica centriranja i $D^{[2]}$ matrica kojoj je svaki element dodatno kvadriran. Dodatno spektralnom dekompozicijom dobivamo jedno od rješenja*

$$X = U\Sigma.$$

*Dokaz.*

Radi jednostavnosti, tražit ćemo centriranu matricu podataka $X$. Za tu matricu vrijedi

$$\sum_i x_i = 0.$$

Kako bismo pronašli odgovarajuće točke, potrebna nam je transformacija udaljenosti

$$d_{ij}^2(X) = x_i^T x_i - 2x_i^T x_j + x_j^T x_j$$

te sumacijom po $i$, po $j$ te po $ij$ dobivamo

$$\begin{aligned}
\frac{1}{m}\sum_{i=1}^{m} d_{ij}(X) &= \frac{1}{m}\sum_{i=1}^{m} x_i^T x_i + x_j^T x_j, \\
\frac{1}{m}\sum_{j=1}^{m} d_{ij}(X) &= x_i^T x_i + \frac{1}{m}\sum_{j=1}^{m} x_j^T x_j, \\
\frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij}^2(X) &= \frac{2}{m}\sum_{i=1}^{m} x_i^T x_i.
\end{aligned}$$

Kako je matrica $X$ centrirana, dio $-2x_i^T(\sum x_j)$ iščezava. Nadalje, iz toga slijedi

$$x_i^T x_j = -\frac{1}{2}\left(d_{ij}^2(X) - \frac{1}{m}\sum_{i=1}^{m} d_{ij}^2(X) - \frac{1}{2}\sum_{j=1}^{m} d_{ij}^2(X) + \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij}^2(X)\right)$$

te uz odgovarajuće oznake

$$x_i^T x_j = a_{ij} - a_{i.} - a_{.j} + a_{..},$$

gdje je

$$\begin{aligned}
a_{ij} &= -\frac{1}{2} d_{ij}^2(X), \\
a_{i.} &= \frac{1}{m}\sum_{j=1}^{m} a_{ij}, \\
a_{.j} &= \frac{1}{m}\sum_{i=1}^{m} a_{ij}, \\
a_{..} &= \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} a_{ij}.
\end{aligned}$$

Ukoliko definiramo matricu $A$ kao $A = [a_{ij}]$, dobivamo da je

$$XX^T = HAH.$$

Kako je $D$ realna simetrična matrica, tako je $HAH$ realna pozitivno semidefinitna matrica, slijedi da je i $XX^T$ pozitivno semidefinitna. Spektralnom dekompozicijom dobivamo

$$XX^T = U\Sigma^2 U^T.$$

Ako uzmemo korjen od $\Sigma^2$, tj. korjenujemo svaki element dijagonale, dobivamo

$$XX^T = U\Sigma\,(U\Sigma)^T$$

$$X := U\Sigma.$$

■

Primijetimo kako udaljenost neka dva podatka novodobivene matrice $\hat{X}$ iznosi

$$d_{ij}^2(X) = \sum_{r=0}^{n} \sigma_r (x_{ir} - x_{jr})^2. \tag{2}$$

Ukoliko želimo sniziti dimenziju podataka, a da što bolje očuvamo kvadrate originalnih udaljenosti podataka, treba sačuvati samo prvih $k$ najvećih desnih singularnih vektora, što vrijedi za sve centrirane matrice.

Primjenom Eckart-Young-Mirsky teorema na danu propoziciju, tj. na $X = U\Sigma$, dobivamo da je $V_k = I_k$, jedinična matrica $I_k \in \mathbb{R}^{n\times k}$, i sniženje dimenzije postižemo čuvanjem samo prvih $k$ singularnih vrijednosti, što odgovara primjedbi (2). Dakle, ukoliko je matrica podataka centrirana, tehnika PCA će nastojati očuvati kvadrate međusobne udaljenosti podataka.

Dodatno, ako je matrica podataka centrirana i zahtijevamo da projicirani podaci isto budu centrirani, sniženje dimenzije tehnikom PCO i tehnikom PCA je ekvivalentno minimizaciji funkcije

$$\min_{r(Y)=k} \sum_i \sum_j \left( d_{ij}^2(X) - d_{ij}^2(Y) \right).$$

## 1.2 Slučajna projekcija

Metoda slučajne projekcije (ili, kraće, RP) je jednostavna, ali efikasna metoda smanjenja dimenzije iza koje stoje lijepi teorijski rezultati. Metoda se provodi jednostavnim množenjem podataka slučajnom matricom, odabranom tako da ima Johnson-Lindenstrauss (JL) svojstvo. Matrice s elementima realiziranim iz jedne velike klase distribucija, kao na primjer normalne ili diskretne uniformne distribucije, sadrže JL svojstvo.

RP metoda se uspješno koristi u raznim primjenama kao što su procesuiranje signala, strojno učenje, podatkovno i tekstualno rudarenje, prijenos podataka i optimizacija određenih funkcija. Ta metoda zahtijeva znatno manje računanja u odnosu na ostale metode korištene u navedenim područjima primjene, ali greška i količina izgubljenih podataka ovisi o slučajnosti i vjerojatnosti. PCA tehnika sniženja dimenzije računski zna biti skupa i separirane klastere podataka, zna zbližiti dok kod RP to nije slučaj. Dodatno, ukoliko podaci dolaze iz mješavine Gaussijana, primjenom ove tehnike Gaussijani će postati više sferični nego što su bili u originalnoj dimenziji. Metoda RP se temelji na sljedećim lemama, koje navodimo bez dokaza (vidi [4, str. 4]).

**Lema 1.1** (Johnson–Lindenstrauss lema)
*Za dani* $0 < \varepsilon < 1$, $X \in \mathbb{R}^{m\times n}$ *i realan broj* $k > C\varepsilon^{-2}\log n$, *gdje je* $C > 0$ *dovoljno velika konstanta, postoji linearno mapiranje* $f : \mathbb{R}^n \to \mathbb{R}^k$ *takvo da vrijedi*

$$(1-\varepsilon)\|x_i - x_j\|_2 \le \|f(x_i) - f(x_j)\|_2 \le (1+\varepsilon)\|x_i - x_j\|_2, \quad \forall\, i, j = 1 \dots m.$$

**Lema 1.2** (Lema slučajne projekcije)
*Neka je* $\varepsilon \in \langle 0, 1\rangle$ *i slučajno normalizirano linearno mapiranje* $T : \mathbb{R}^n \to \mathbb{R}^k$. *Tada za svaki* $x \in \mathbb{R}^n$ *vrijedi*

$$P\left((1-\varepsilon)\|x\|_2 \le \|Tx\|_2 \le (1+\varepsilon)\|x\|_2\right) \ge 1 - 2e^{-c\varepsilon^2 k},$$

*gdje je* $c > 0$ *neka konstanta neovisna o* $n$, $k$ *i* $\varepsilon$.

Slučajno normalizirano linearno mapiranje $T$ iz leme 1.2 možemo izabrati kao npr.

- $T = \frac{1}{\sqrt{k}}A$ gdje je svaki element matrice $A$ slučajno odabran iz standardne normalne distribucije $N(0,1)$,
- $T = \frac{1}{\sqrt{k}}A$ gdje svaki element od $A$ poprima vrijednost 1 ili $-1$, svaku s vjerojatnošću $\frac{1}{2}$.

Leme 1.1 i 1.2 nam govore da ukoliko nam originalni podaci imaju distribuciju neke mješavine Gaussijana, tada postoji vjerojatnost da će Gaussijani u toj mješavini ostati jednako separirani. Definirajmo mješavinu Gaussijana.

**Definicija 1.1**
*$n-$dimenzionalni Gaussijan je slučajna varijabla $N(\mu, \Sigma)$, čija je funkcija gustoće*

$$f(x) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}.$$

**Definicija 1.2** (c-separiranost klastera)
*Dva sferična Gaussijana $N(\mu_1, \sigma^2 I_n)$ i $N(\mu_2, \sigma^2 I_n)$ su c-separirana ako*

$$\|\mu_1 - \mu_2\| \geq c\sigma\sqrt{n}.$$

*Općenitije, Gaussijani $N(\mu_1, \Sigma_1)$ i $N(\mu_2, \Sigma_2)$ u $\mathbb{R}^n$ su c-separirani ako*

$$\|\mu_1 - \mu_2\| \geq c\sqrt{\max\{tr(\Sigma_1), tr(\Sigma_2)\}}$$

Sljedeće dvije leme nam govore o svojstvima projiciranih podataka koja su očuvana RP tehnikom redukcije dimenzije.

**Lema 1.3** (Dasgupta [4])
*Neka je mješavina od $t$ Gaussijana iz $\mathbb{R}^n$ c-separirana i neka su $\delta, \varepsilon \in \langle 0, 1\rangle$ vjerojatnosni parametar i parametar preciznosti. Ukoliko je mješavina projicirana u podprostor dimenzije $k \geq \frac{C_1}{\varepsilon^2}\ln\frac{t}{\delta}$, gdje je $C_1$ neka univerzalna konstanta, tada s vjerojatnošću većom od $1-\delta$, projicirana mješavina Gaussijana u $\mathbb{R}^k$ će biti $c\sqrt{1-\varepsilon}$-separirana.*

**Lema 1.4** (Dasgupta [4])
*Neka su $\delta, \varepsilon \in \langle 0, 1\rangle$ vjerojatnosni parametar i parametar preciznosti. Ako je proizvoljan Gaussijan iz prostora $\mathbb{R}^n$ projiciran u nasumično odabran podprosto dimenzije $k$ takav da vrijedi*

$$n > C_2\frac{\lambda_{\max}(\Sigma)}{\lambda_{\min}(\Sigma)\varepsilon^2}\left(\log\frac{1}{\delta} + k\log\frac{k}{\varepsilon}\right),$$

*gdje je $C_2$ neka univerzalna konstanta. Tada uz vjerojatnost većom od $1-\delta$, omjer za projicirane podatke bit će*

$$\sqrt{\frac{\lambda_{\max}(\hat{\Sigma})}{\lambda_{\min}(\hat{\Sigma})}} \leq 1 + \varepsilon,$$

*gdje je $\lambda_{\max}(\Sigma)$ najveća svojstvena vrijednost matrice $\Sigma$, a $\lambda_{\min}(\Sigma)$ najmanja svojstvena vrijednost. Ako vrijedi za originalne podatke*

$$\sqrt{\frac{\lambda_{\max}(\Sigma)}{\lambda_{\min}(\Sigma)}} \leq n^{1/2}C_2^{-1/2}\left(\log\frac{1}{\delta} + k\log k\right)^{-1/2},$$

*onda s vjerojatnošću većom od* $1-\delta$*, projicirani podaci će imati*

$$\sqrt{\frac{\lambda_{\max}\hat{\Sigma}}{\lambda_{\min}\hat{\Sigma}}} \geq 2.$$

Lema 1.3 govori o očuvanju c-separiranosti klastera dok lema 1.4 govori o obliku novih klastera, odnosno, novi klasteri će biti više sferični nego što su bili klasteri u originalnoj dimeniziji. Ova dva svojstva metode RP su se pokazala jako korisnim u strojnom učenju, u problemima klasifikacije i grupiranju podataka.

# 2 Metrički MDS

U ovom poglavlju bavit ćemo se metričkim višedimenzionalnim skaliranjme (ili, kraće, mMDS-om). mMDS je klasičan algoritam za pronalazak strukture podataka preko matrice međusobnih udaljenosti ili matrice različitosti. Glavna ideja MDS algoritma je pronalazak točaka

$$X = [x_1, x_2, \ldots, x_n]^T,$$

u željenoj dimenziji, čije međusobne euklidske udaljenosti ($\|x_i - x_j\|_2$) odgovaraju danoj matrici udaljenosti $\Delta = [\delta_{ij}]$. Za razliku od klasičnog višedimenzionalnog skaliranja, metričko višedimenzionalno skaliranje projicira podatke u nižu dimenziju na nelinearan način i eksplicitno ne daje funkciju projekcije kojom radi projiciranje. Prvu metodu za računanje mMDS-a predložio je Sammon još 1969. godine koja Newton-ovom metodom rješava minimizacijski problem oblika

$$\bar{X} = \arg\min_X \sigma(X).$$

$$\sigma(X) = \frac{1}{\sum\limits_{i<j} \delta_{ij}} \sum_{i<j} \frac{\left(\|x_i - x_j\|_2 - \delta_{ij}\right)^2}{\delta_{ij}} \tag{3}$$

i time dobiva točke $X$ u željenoj dimenziji čija je matrica euklidskih udaljenosti najsličnija traženoj matrici udaljenosti $\Delta$. Kako ju je Sammon prvi predložio, funkcija (3) dobiva naziv Sammon-ovo mapiranje. Često se u literaturi, zbog oblika funkcije (3), mMDS još naziva i skaliranje pomoću najmanjih kvadrata te se u praksi umjesto minimizacije funkcije (3) optimizira generalnija funkcija

$$\sigma_W(X) = \sum_{i<j} w_{ij} \left(\|x_i - x_j\|_2 - \delta_{ij}\right)^2, \tag{4}$$

koja rješava isti problem. Funkcije (3) i (4) nisu konveksne zbog čega, ovisno o izboru početnih parametara i veličina $w_{ij}$, možemo dobiti različita i neoptimalna rješenja. Jednako tako, matrica euklidskih udaljenosti je neovisna o rotaciji, translaciji i zrcaljenju rješenja što ne utječe na vrijednost funkcije, nego samo na podatke $X$.

Kako bismo mogli minimizirati funckiju (4) trebamo prvo poznavati gradijent te funkcije. Parcijalna derivacija funckije $d_{ij}(X) = \|x_i - x_j\|_2$ po podatku $x_t$ iznosi nula ukoliko $i \neq t$ i $j \neq t$. U slučaju kada je $i = t$, ona iznosi:

$$\frac{\partial}{\partial x_t} \left(d_{tj}(X)\right)^2 = \frac{\partial}{\partial x_t} (x_t - x_j)^T (x_t - x_j)$$

$$2 d_{tj}(X) \frac{\partial}{\partial x_t} d_{tj}(X) = 2(x_t - x_i). \tag{5}$$

Izračunajmo sada derivaciju funkcije cilja (4) po podatku $x_t$:

$$\frac{\partial}{\partial x_t} \sigma_W(X) = \frac{\partial}{\partial x_t} \sum_{i<j} w_{ij} \left(\|x_i - x_j\|_2 - \delta_{ij}\right)^2.$$

Dovoljno je gledati samo članove u sumi oblika $w_{tj} \left(\|x_t - x_j\|)_2 - \delta_{tj}\right)^2$ jer ostali članovi ne ovise o varijabli $x_t$ te stoga njihova derivacija iznosi nula.

Dakle, tada imamo:

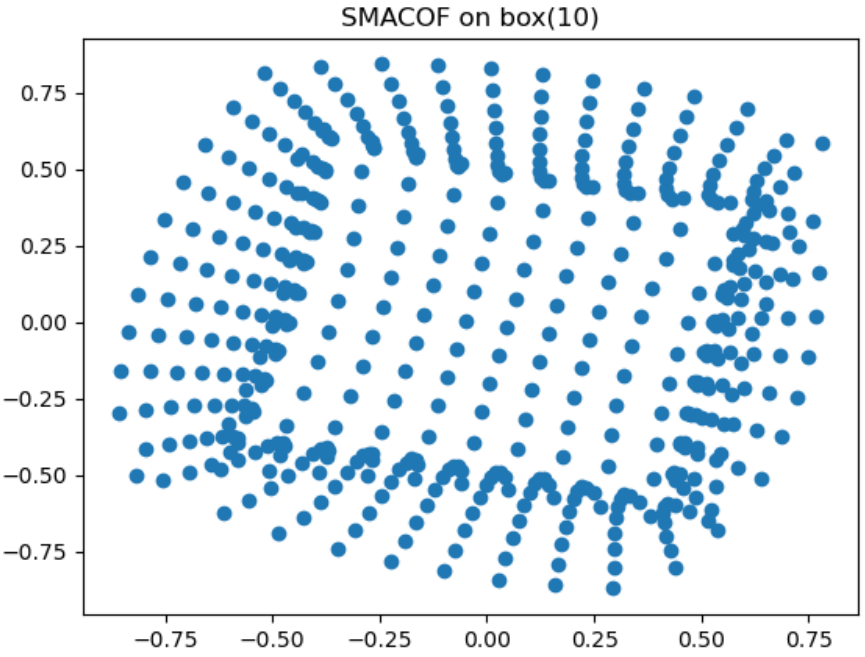
$$\frac{\partial}{\partial x_t}\sigma_W(X) = \sum_{j\neq t} w_{tj}\frac{\partial}{\partial x_t}\left(\|x_t - x_j\|_2 - \delta_{ij}\right)^2$$
$$\frac{\partial}{\partial x_t}\sigma_W(X) = \sum_{j\neq t} 2w_{tj}\left(\|x_t - x_j\|_2 - \delta_{ij}\right)\cdot\frac{\partial}{\partial x_t}\|x_t - x_j\|_2$$

te korištenjem (5) dobivamo

$$\frac{\partial}{\partial x_t}\sigma_W(X) = 2\sum_{j\neq t} w_{tj}\frac{\|x_t - x_j\|_2 - \delta_{tj}}{\|x_t - x_j\|_2}\left(x_t - x_j\right). \tag{6}$$

Formula (6) nam govori u kojem smjeru moramo pomaknuti točku $x_t$ želimo li smanjiti vrijednost funkcije cilja. Uočimo da funkcija nije diferencijabilna u svim točkama jer za $x_t = x_j$ dobivamo nulu u nazivniku, što stvara problem pri računanju.

Kako bismo si lakše predočili što to zapravo Sammon-ovo mapiranje radi, reducirajmo otvorenu kutiju (Slika 2) ovom nelinearnom tehnikom smanjenja dimenzije.




Slika 4: SMACOF na otvorenoj kutiji.

Slika 4 prikazuje otvorenu kutiju nakon redukcije dimenzije nelinearnom tehnikom Sammon-ovo mapiranje. Za optimizaciju funckije cilja (4) korišten je specijalizirani algoritam SMACOF, o kojem ćemo nešto više reći u sljedećem poglavlju.

Lako je vidljiva razlika između linearne tehnike PCA (Slika 3) i nelinearne tehnike Sammon-ovo mapiranje (Slika 4). Nakon što su podaci "spljošteni", Sammon-ovo mapiranje dodatno pogura podatke u smjeru u kojem će međusobne udaljenosti podataka biti najbolje očuvane.

## 2.1 SMACOF

Skaliranje majorizacijom komplicirane funkcije (ili, kraće, SMACOF) je algoritam od teorijske i praktične važnosti. Napravljen je kao rješenje problema nediferencijabilnosti originalnog algoritma za minimizaciju Sammon-ove funkcije. Sam algoritam je jednostavan za implementirati te koristi osnovne rezultate linearne algebre i diferencijalnog računa.

Guttman prvi predstavlja verziju ovog algoritma zvanu Guttman transformacija koja pomoću rješenja jednadžbe

$$X = V^{+}B(X)X$$

po $X$ pronalazi podatke u nižoj dimenziji za koje vrijedi da je (4) minimalna. Za pronalazak $X$, Guttman predlaže gradijentnu metodu te ne daje dokaz o konvergenciji. Zatim, De Leeuw predlaže trenutni majorizacijski pristup pronalaska $X$ poznat kao SMACOF. Također, nudi iterativnu metodu rješavanja mMDS problema te jednostavnim dokazom pokazuje konvergenciju novog predloženog algoritma koji rješava problem nediferencijabilnosti (6) u određenim točkama.

Radi lakšeg razumijevanja, uvest ćemo nekoliko oznaka i definicija. Počet ćemo s upravljanjem osnovnim podacima te kasnije doći do matričnog zapisa osnovnih funkcija. Ako bismo htjeli uzeti $i-$ti podatak iz naše matrice $X$, tu matricu bismo trebali pomnožiti sa $i$-tim vektorom kanonske baze, t.j. vektorom koji na $i$-toj poziciji sadrži jedinicu, a na ostalim pozicijama nule, odnosno

$$x_i = X^T e_i.$$

Koristeći takav zapis podatka, ponvno definirajmo udaljenost dva podatka.

**Definicija 2.1**
*Definirajmo matricu $A_{ij}$ na način:*

$$A_{ij} = (e_i - e_j)^T(e_i - e_j),$$

*gdje je $e_i$ $i$-ti vektor kanonske baze. Tada udaljenost $d_{ij}(X)$ $i$-tog i $j$-tog podatka možemo transformirat tako da dobijemo oblik iskazan pomoću matrica:*

$$\begin{aligned}
(d_{ij}(X))^2 &= \|x_i - x_j\|_2^2 \\
&= (x_i - x_j)^T(x_i - x_j) \\
&= (X^T e_i - X^T e_j)^T(X^T e_i - X^T e_j) \\
&= (e_i - e_j)^T X X^T (e_i - e_j) \\
&= tr\left(X^T(e_i - e_j)^T(e_i - e_j)X\right) \\
&= tr\left(X^T A_{ij} X\right).
\end{aligned}$$

Kako bismo preoblikovali funkciju cilja (4) u nama pogodniji oblik, trebaju nam sljedeće dvije definicije.

**Definicija 2.2**
*Difinirajmo matricu težina $V$ koristeći težine $w_{ij}$ iz (4) na način:*

$$V = \sum_{i<j} w_{ij} A_{ij}.$$

Iz ovakve definicije matrice $V$ slijedi elegantan zapis izraza:

$$\sum_{i<j} w_{ij} \left(d_{ij}(X)\right)^2 = tr\left(X^T V X\right).$$

**Definicija 2.3**
*Definirajmo matricu $B(X)$ po uzoru na prijašnju matricu, na način:*

$$B(X) = \sum_{\substack{i<j \\ d_{ij}(X)>0}} w_{ij} \frac{\delta_{ij}}{d_{ij}(X)} A_{ij}.$$

Jednako kao i $V$, matricu $B(X)$ smo uveli da bismo mogli elegantnije zapisati izraz oblika:

$$\sum_{i<j} w_{ij}\delta_{ij} d_{ij}(X) = tr\left(X^T B(X) X\right).$$

Sada kada smo definirali osnovne matrice, funkciju cilja (4) možemo zapisati na pogodniji način u obliku sljedeće definicije.

**Definicija 2.4**
*Ponovno definirajmo funkciju cilja $\sigma(X)$ za problem smanjenja dimenzije. Počevši od* (4)*, nizom transformacija dobivamo izraz oblika:*

$$\begin{aligned}\sigma(X) &= \sum_{i<j} w_{ij} \left(\delta_{ij} - d_{ij}(X)\right)^2 \\ &= \sum_{i<j} w_{ij}\delta_{ij}^2 - 2tr\left(X^T B(X) X\right) + tr\left(X^T V X\right).\end{aligned}$$

*Budući da je $\sum_{i<j} w_{ij}\delta_{ij}^2$ konstanta, uvodimo oznaku $c = \sum_{i<j} w_{ij}\delta_{ij}^2$ te time zapis postaje*

$$\sigma(X) = c - 2tr\left(X^T B(X) X\right) + tr\left(X^T V X\right). \tag{7}$$

Kako smo funkciju (4) zapisali koristeći matrice, tako možemo i njenu derivaciju (6). Sljedećom definicijom dajemo taj zapis derivacije.

**Definicija 2.5**
*Definirajmo derivaciju $\nabla\sigma(X)$, koristeći* (6) *i definicije 2.2 i 2.3, na način:*

$$\nabla\sigma(X) = -2B(X)X + 2VX.$$

Uočimo da ukoliko postavimo $\nabla\sigma(X)$ na nula dobivamo kojeg su oblika stacionarne točke $X$ funkcije (7):

$$VX = B(X)X.$$

Koristeći Moore-Penrose inverz $V^+$ (vidi [8]) dobivamo temeljnu jednadžbu za konstruckiju SMACOF algoritma:

$$X = V^+ B(X) X.$$

Računanje desne strane ove jednakosti je SMACOF algoritam. Definirajmo to kao jednu iteraciju alogritma.

**Definicija 2.6** (Guttman transformacija)
*Definirajmo Guttman transformaciju $\Gamma(X)$ na način:*

$$\Gamma(X) = V^+ B(X) X \tag{8}$$

Primjenom Guttman transformacije na projekciju podataka smanjujemo vrijednost funkcije cilja (4). Konvergenciju algoritma ćemo dokazati u teoremu korak SMACOF-a, pa su nam u tu svrhu potrebne sljedeće leme.

**Lema 2.1**
*Za sve $X$ i $Y$ iz $\mathbb{R}^{m\times n}$ vrijedi*

$$tr\left(X^T B(X)X\right) \geq tr\left(X^T B(Y)Y\right).$$

*Dokaz.*

Po Cauchy-Schwarz nejednakosti (vidi [3]) slijedi

$$tr\left(X^T A_{ij}Y\right) \leq \sqrt{tr\left(X^T A_{ij}X\right)} \cdot \sqrt{tr\left(Y^T A_{ij}Y\right)} = d_{i,j}(X) \cdot d_{ij}(Y).$$

Ukoliko pomnožimo obje strane s $w_{ij}\delta_{i,j}/d_{ij}(Y)$ i prosumiramo po svim $i < j$ dobivamo:

$$\sum_{i<j} tr\left(X^T\left(w_{ij}\frac{\delta_{ij}}{d_{ij}(Y)}A_{ij}\right)Y\right) \leq \sum_{i<j} w_{ij}\delta_{ij}d_{ij}(X)\frac{d_{ij}(Y)}{d_{ij}(Y)}$$

$$tr\left(X^T B(Y)Y\right) \leq tr\left(X^T B(X)X\right).$$

■

**Lema 2.2**
*Za sve $X$ i $Y$ iz $\mathbb{R}^{m\times n}$ vrijedi*

$$-2tr\left(X^T VY\right) + \eta^2(X) = \eta^2\left(X - Y\right) - \eta^2(Y),$$

*gdje je*

$$\eta^2(Z) = tr\left(Z^T VZ\right).$$

*Dokaz.*

Budući da je $\eta^2(Z) = tr\left(Z^T VZ\right)$, tada za $Z = X - Y$ imamo:

$$\begin{aligned}\eta^2(X-Y) &= tr\left((X-Y)^T V(X-Y)\right)\\ &= tr\left(X^T VX - X^T VY - Y^T VX + Y^T VY\right)\\ &= \eta^2(X) - tr\left(X^T VY\right) - tr\left(Y^T VX\right) + \eta^2(Y).\end{aligned}$$

Znamo da je $tr(X) = tr(X^T)$ te iz konstrukcije matrice $V$ vrijedi $V^T = V$ pa imamo:

$$\begin{aligned}tr\left(Y^T VX\right) &= tr\left(X^T V^T Y\right)\\ tr\left(X^T V^T Y\right) &= tr\left(X^T VY\right)\end{aligned}$$

iz čega trivijalno slijedi:

$$\eta^2(X-Y) = \eta^2(X) - 2tr\left(X^T VY\right) + \eta^2(Y).$$

■

**Lema 2.3**
*Za sve $X$ i $Y$ iz $\mathbb{R}^{m\times n}$ vrijedi*

$$\sigma(X) \leq c - \eta^2\left(\Gamma(Y)\right) + \eta^2\left(X - \Gamma(Y)\right),$$

*gdje je*

$$\eta^2(Z) = tr\left(Z^T V Z\right).$$

*Dokaz.*

Prema definiciji $\sigma(X)$ slijedi

$$\sigma(X) = c - 2tr\left(X^T B(X) X\right) + \eta^2(X).$$

Korištenjem leme 2.1 dobivamo

$$\sigma(X) \leq c - 2tr\left(X^T B(Y) Y\right) + \eta^2(X)$$

te uvođenjem zamjene $V\Gamma(Y) = VV^+B(Y)Y = B(Y)Y$ imamo

$$\sigma(X) \leq c - 2tr\left(X^T V\Gamma(Y)\right) + \eta^2(X).$$

Iz leme 2.2 slijedi:

$$\sigma(X) \leq c - \eta^2\left(\Gamma(Y)\right) + \eta^2\left(X - \Gamma(Y)\right).$$

■

**Teorem 2.1** (korak SMACOF-a)
*Za sve $X \in \mathbb{R}^{m\times n}$ vrijedi*

$$\sigma\left(\Gamma(X)\right) \leq \sigma(X).$$

*Dokaz.*

Uvrštavanjem $\Gamma(X)$ u lemu 2.3 dobivamo

$$\sigma\left(\Gamma(X)\right) \leq c - \eta^2\left(\Gamma(Y)\right) + \eta^2\left(\Gamma(X) - \Gamma(Y)\right), \forall Y \in \mathbb{R}^{m\times n}.$$

Posebno, za $Y = X$ imamo

$$\sigma(\Gamma(X)) \leq c - \eta^2\left(\Gamma(X)\right).$$

Kako je za svaki $Z \in \mathbb{R}^{m\times n}$, $\eta^2(Z) \geq 0$, jer je $\eta^2(Z) = \sum\limits_{i<j} w_{ij} d_{ij}^2(Z)$, vrijedi

$$\begin{aligned}\sigma(\Gamma(X)) &\leq c - \eta^2\left(\Gamma(X)\right)\\ &\leq c - \eta^2\left(\Gamma(X)\right) + \eta^2\left(X - \Gamma(X)\right)\\ &= \sigma(X).\end{aligned}$$

■

Teorem 2.1 nam govori da iterativnom primjenom funkcije (8) na naše podatke funkcija cilja (4) monotono pada. Znamo da je funkcija cilja nenegatvina iz čega, zajedno sa monotonim padom, slijedi da funkcija cilja konvergira.

**Napomena 2.1**
*Važno je napomenuti da kako* $n \to \infty$ *i* $\sigma(X_n) \to \sigma$*, ne znači da vrijedi* $X_n \to X$*. Razlog tome je to što za matricu rotacije* $K$ *vrijedi* $D_{ij}(X) = D_{ij}(KX)$ *pa onda i* $\sigma(X) = \sigma(KX)$*.*

Odnosno, ako funkcija cilja $\sigma(X_n)$ konvergira prema $\sigma$, ne znači da će i niz novih podataka $X_n$ konvergirati prema rješenju $X$. Rješenje $X$ nije jedinstveno i za proizovljnu matricu rotacije ili matricu zrcaljenja $K$, matrica $KX$ je još jedno rješenje optimizacijskog problema (4) s funkcijom cilja jednakom rješenju $X$. Naime, to znači $\sigma(X) = \sigma(KX)$.

Sada kada imamo dokazan teorem o konvergenciji SMACOF-a, možemo dati i sam algoritam. Radi jednostavnosti, algoritam dajemo napisan u programskom jeziku python.

```
import numpy as np
from scipy.spatial.distance import squareform, pdist

def D(X : np.array):
  return squareform(pdist(X))

def SMACOF(Delta : np.array, dim : int, W : np.array, max_iter = 300):
  V = -W
  np.fill_diagonal(V, -V.sum(axis = 0))

  Vp = np.linalg.pinv(V) # Moore-Penrose

  X = np.random.rand(len(Delta), dim)
  for _ in range(max_iter):
    Dx = D(X)
    Bx = -W * Delta * np.reciprocal(Dx, where = (Dx!=0))
    np.fill_diagonal(Bx, -Bx.sum(axis = 0))
    X = Vp.dot(Bx).dot(X)

  return X
```

# Literatura


[1] C. M. Bishop, *Pattern recognition and machine learning*, Springer, New York, 2006.

[2] I. Borg, P. J. F. Groenen, *Modern Multidimensional Scaling Theory and Applications*, Springer, New York, 2005.

[3] Chauchy-Schwarz inequality, Wikipedia. Dostupno na: `https://en.wikipedia.org/wiki/Cauchy%E2%80%93Schwarz_inequality` (19. rujan 2020.)

[4] S. Dasgupta, *Experiments with random projection*, AT&T Labs – Research.

[5] Eckart–Young–Mirsky theorem, Wikipedia. Dostupno na: `https://en.wikipedia.org/wiki/Low-rank_approximation` (22. rujan 2020.)

[6] J. A. Lee, M. Verleysen, *Nonlinear Dimensionality Reduction*, Springer – Verlag, New York, 2007.

[7] J. Matoušek, *Lecture notes on metric embeddings*, 2013. Dostupno na: `https://kam.mff.cuni.cz/~matousek/ba-a4.pdf` (27. kolovoz 2020.)

[8] Moore-Penrose inverse, Wikipedia. Dostupno na: `https://en.wikipedia.org/wiki/Moore%E2%80%93Penrose_inverse` (29. rujan 2020.)

[9] S. Shalev–Shwartz, S. Ben–David, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, 2014.

[10] Singular Value Decomposition, Wikipedia. Dostupno na: `https://en.wikipedia.org/wiki/Singular_value_decomposition` (29. rujan 2020.)

[11] C. van Wezel, N. Kok, A. Kosters, *Two Neural Network Methods for Multidimensional Scaling*, Leiden University, Dept. of Mathematics and Computer Science, Leiden, 1996.

[12] Jan de Leeuw, Convergence of the Majorization Method for Multidimensional Scaling. Dostupno na: `deleeuwpdx.net/janspubs/1988/articles/deleeuw_A_88b.pdf` (27. kolovoz 2020.)

[13] Jan de Leeuw, Convergence of SMACOF. Dostupno na: `http://deleeuwpdx.net/pubfolders/converge/converge.html` (27. kolovoz 2020.)

[14] Introduction To Probability, Statistics and Random Processes, Probability Course. Dostupno na: `https://www.probabilitycourse.com/` (27. kolovoz 2020.)

[15] Multidimensional Scaling, SpringerLink. Dostupno na: `https://link.springer.com/chapter/10.1007%2F978-3-540-33037-0_14` (27. kolovoz 2020.)