Neke linearne metode redukcije dimenzije visokodimenzionalnih podataka

Vinković, Ivana

Sveučilište J. J. Strossmayera u Osijeku
Fakultet primjenjene matematike i informatike
Sveučilišni prijediplomski studij matematike i računarstva

Ivana Vinković

Neke linearne metode redukcije dimenzije visokodimenzionalnih podataka

Završni rad

Osijek, 2024.

Sveučilište J. J. Strossmayera u Osijeku
Fakultet primjenjene matematike i informatike
Sveučilišni prijediplomski studij matematike i računarstva

Ivana Vinković

Neke linearne metode redukcije dimenzije visokodimenzionalnih podataka

Završni rad

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

Osijek, 2024.

Sažetak

U ovom radu fokusirat ćemo se na metode za redukciju dimenzije visokodimenzionalnih podataka s posebnim naglaskom na analizu glavnih komponenti (PCA) i dekompoziciju na singularne vrijednosti (SVD). Istražit ćemo optimalan odabir broja dimenzija u nižedimenzionalnom prostoru kako bi se očuvala ključna struktura i značajke podataka. Navedene metode koriste se za smanjenje dimenzije i očuvanje što većeg broja informacija, što omogućava njihovu primjenu u kompresiji podataka, vizualizaciji, strojnom učenju i prepoznavanju obrazaca.

Ključne riječi

redukcija dimenzije, analiza glavnih komponenti, dekompozicija na singularne vrijednosti, visokodimenzionalni podaci, odabir dimenzije

Abstract

In this paper, we focus on methods for dimensionality reduction of high-dimensional data, with a particular emphasis on Principal Component Analysis (PCA) and Singular Value Decomposition (SVD). We explore the optimal selection of the number of dimensions in the lower-dimensional space to preserve the key structure and features of the data. These methods are used to reduce dimensionality while retaining as much information as possible, enabling their application in data compression, visualization, machine learning, and pattern recognition.

Keywords

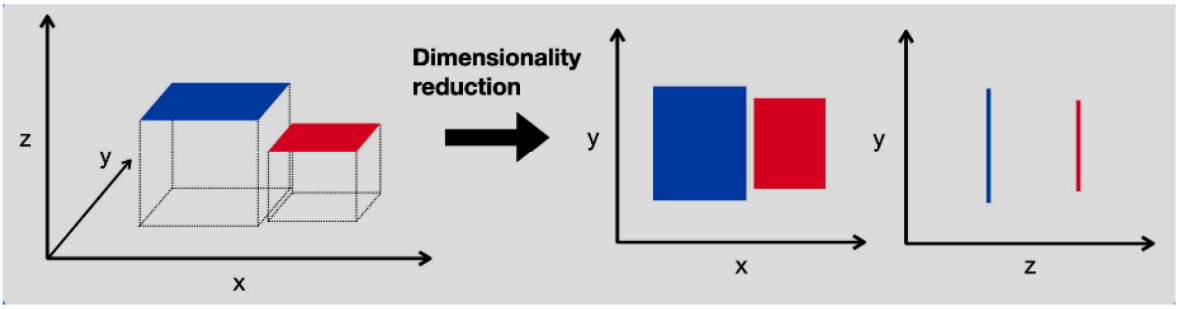
dimensionality reduction, principal component analysis, singular value decomposition, high-dimensional data, dimension selection

Sadržaj

1 Uvod

U problemima strojnog učenja i analize podataka često se susrećemo s visokodimenzionalnim skupovima podataka, gdje broj značajki može biti izuzetno velik. Takvi skupovi podataka mogu uzrokovati nekoliko izazova. Visoka dimenzionalnost može dovesti do povećane računske složenosti, što zahtijeva više vremena i resursa za obradu podataka, dok u nekim situacijama visoka dimenzionalnost može pogoršati sposobnost generalizacije algoritama strojnog učenja.

Redukcija dimenzionalnosti omogućava transformaciju podataka iz visokodimenzionalnog prostora u prostor niže dimenzije, pri čemu se nastoji očuvati što veći broj relevantnih informacija. Ovaj proces je usko povezan s pojmom kompresije podataka u teoriji informacija, gdje se podaci smanjuju uz minimalni gubitak korisnih informacija. Redukcija dimenzionalnosti se koristi iz nekoliko razloga: olakšava računsku obradu, može poboljšati generalizacijske sposobnosti algoritama te omogućuje bolju interpretaciju podataka, pronalaženje smislenih struktura i vizualizaciju podataka.



Slika 1: Primjer redukcije dimenzionalnosti: 3D podaci (lijevo) projicirani su na nižedimenzionalne prostore. Nakon redukcije, podaci su prikazani u 2D.

U ovom radu ćemo istražiti dvije popularne linearne metode za redukciju dimenzionalnosti odnosno analizu glavnih komponenti (PCA) i dekompoziciju na singularne vrijednosti (SVD). Obje metode se oslanjaju na primjenu linearnih transformacija koje projiciraju podatke iz visokodimenzionalnog prostora u prostor niže dimenzije

2 Analiza glavnih komponenti (PCA)

Analiza glavnih komponenti (PCA) je tehnika koja se široko koristi za primjene poput smanjenja dimenzionalnosti, ekstrakcije značajki te kompresije i vizualizacije podataka.

PCA omogućuje računalno učinkovito smanjenje složenosti podataka očuvanjem što više relevantnih informacija. Visokodimenzionalni skupovi podataka često uzrokuju povećanje računalnog vremena, složenosti modela i pretreniranost algoritama. Smanjenjem dimenzionalnosti, PCA olakšava daljnju analizu i poboljšava performanse algoritama.

2.1 Metode za računanje PCA

PCA se može definirati na dva načina:

- **Ortogonalna projekcija** podataka na linearni prostor niže dimenzije, poznat kao glavni potprostor, pri čemu se maksimizira varijanca projiciranih podataka.
- **Linearna projekcija koja minimizira prosječnu pogrešku**, tj. prosječnu kvadratnu udaljenost između podataka i njihovih projekcija.

Iz prethodnih definicija proizlazi isti algoritam, za kojeg osim naziva analiza glavnih komponenti koristimo i naziv Karhunen-Loèveova transformacija.

Osvrnimo se na prvi pristup, gdje je cilj maksimizirati varijancu projiciranih podataka, uz pretpostavku da se podaci projiciraju na jednodimenzionalni prostor ($M = 1$).

Razmotrimo skup podataka $\{x_n\}$ dimenzije D, pri čemu je $n = 1, \ldots, N$. Smjer ove projekcije definiramo pomoću D-dimenzionalnog jediničnog vektora u_1, pri čemu za jednostavnost i bez smanjenja općenitosti pretpostavljamo da:

$$u_1^T u_1 = 1.$$

Time nas zanima samo smjer koji definira u_1, a ne njegova veličina.

Svaka točka x_n projicira se na skalar kao $u_1^T x_n$. Srednja vrijednost skupa podataka izračunava se kao:

$$\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n.$$

Na temelju toga, kovarijacijska matrica S skupa podataka definira se formulom:

$$S = \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})(x_n - \bar{x})^T.$$

Cilj je maksimizirati varijancu projiciranih podataka, koja je dana izrazom:

$$\mathrm{Var}(u_1^T x) = u_1^T S u_1.$$

Da bismo maksimizirali varijancu pod uvjetom da u_1 ostaje jedinični vektor ($u_1^T u_1 = 1$), koristimo metodu Lagrangeovih multiplikatora[2]. Definiramo funkciju:

$$\mathcal{L}(u_1, \lambda_1) = u_1^T S u_1 + \lambda_1 (1 - u_1^T u_1).$$

Deriviranjem po u_1 i postavljanjem derivacije na nulu dobivamo:

$$Su_1 = \lambda_1 u_1,$$

što znači da je u_1 svojstveni vektor matrice S koji odgovara najvećoj svojstvenoj vrijednosti λ_1.

Kada je u_1 jednak ovom svojstvenom vektoru, maksimalna varijanca iznosi:

$$u_1^T S u_1 = \lambda_1.$$

Dodatne glavne komponente mogu se definirati iterativno, pri čemu svaka nova komponenta maksimizira varijancu uz uvjet ortogonalnosti na prethodne komponente. Ako razmotrimo opći slučaj projekcije na M-dimenzionalni prostor, optimalna linearna projekcija, za koju je varijanca projiciranih podataka maksimizirana, definirana je pomoću M svojstvenih vektora $u_1, \ldots, u_M$ kovarijacijske matrice podataka S koji odgovaraju M najvećim svojstvenim vrijednostima $\lambda_1, \ldots, \lambda_M$. Ovo se vrlo jednostavno može pokazati pomoću metode matematičke indukcije.

Ako pogledamo složenost ovakvog pristupa pronalazak svih svojstvenih vektora matrice S dimenzija $D \times D$ zahtijeva $O(D^3)$. Međutim, za veće skupove podataka i efikasnije računanje koriste se metode potencija[4] ili alternativno EM algoritam[9].

Drugi pristup računanja PCA temelji se na minimiziranju ukupne pogreške projekcije, odnosno minimizaciji srednje kvadratne udaljenosti između originalnih podataka x_n i njihovih aproksimacija $\hat{x}_n$. Razmotrimo skup podataka $\{x_n\}$ dimenzije D, pri čemu $n = 1, \ldots, N$. Svaki podatak može se izraziti kao linearna kombinacija vektora baze $\{u_i\}$, gdje je $i = 1, \ldots, D$, uz uvjet ortogonalnosti:

$$u_i^T u_j = \delta_{ij}.$$

Znajući tu činjenicu aproksimacija vektora x_n u prostoru manje dimenzije $M < D$ postiže se prikazivanjem svakog podatka kao kombinacije ortogonalnih vektora baze. Najprije definirajmo aproksimaciju vektora x_n u obliku:

$$\hat{x}_n = \sum_{i=1}^{M} z_{ni} u_i + \sum_{i=M+1}^{D} b_i u_i,$$

gdje z_{ni} predstavaljaju koeficijente projekcije koji ovise o svakom podatku x_n a b_i predstavljaju konstante jednake za sve podatke i ne ovise o uzorku n.

Prisjetimo se da je cilj ove metode je minimizirati ukupnu pogrešku projekcije, definiranu kao srednja kvadratna udaljenost između originalnih podataka x_n i njihovih aproksimacija $\hat{x}_n$:

$$J = \frac{1}{N} \sum_{n=1}^{N} \|x_n - \hat{x}_n\|^2.$$

Sada koristeći ortogonalnost vektora baze u_i, koeficijenti projekcije mogu se izračunati kao:

$$z_{nj} = x_n^T u_j, \quad j = 1, \ldots, M,$$

a vrijednosti konstanti kao:

$$b_j = \bar{x}^T u_j, \quad j = M+1, \ldots, D,$$

gdje je $\bar{x}$ srednja vrijednost skupa podataka, definirana kao:

$$\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n.$$

Zamjenom izraza za z_{nj} i b_j u funkciju pogreške J, dobivamo:

$$J = \frac{1}{N} \sum_{n=1}^{N} \|x_n - \hat{x}_n\|^2 = \sum_{i=M+1}^{D} u_i^T S u_i.$$

gdje S predstavlja kovarijacijska matrica definirana kao:

$$S = \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})(x_n - \bar{x})^T,$$

Kovarijacijska matrica S mjeri varijabilnost podataka i omogućuje izračunavanje svojstvenih vektora i svojstvenih vrijednosti potrebnih za daljnju analizu.

Ukupna pogreška projekcije može se izraziti kao suma svojstvenih vrijednosti za one vektore koji su ortogonalni na glavni potprostor:

$$J = \sum_{i=M+1}^{D} \lambda_i,$$

gdje λ_i predstavljaju svojstvene vrijednosti kovarijacijske matrice S.

Minimizacija pogreške postiže se odabirom vektora u_i kao svojstvenih vektora kovarijacijske matrice S koji odgovaraju M najvećim svojstvenim vrijednostima $\lambda_1, \lambda_2, \ldots, \lambda_M$. Ovi vektori definiraju smjerove glavnih komponenti, omogućujući smanjenje dimenzionalnosti uz očuvanje maksimalne količine varijance podataka.

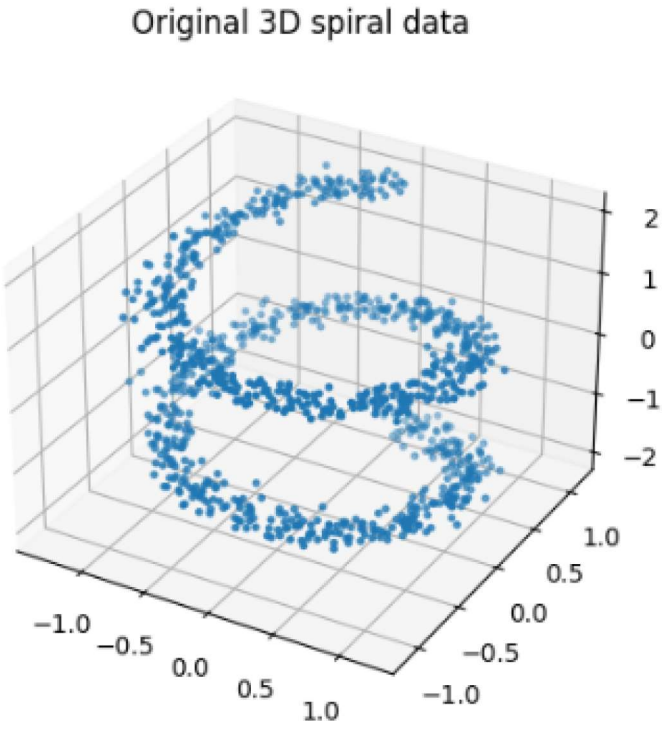
Ako su svojstvene vrijednosti jednake, bilo koji smjer može se koristiti kao glavna komponenta jer svi jednako minimiziraju pogrešku. U tom slučaju, smjerovi glavnih komponenti mogu se odabrati slobodno, jer će ukupna pogreška J ostati ista za sve moguće smjerove unutar tog potprostora.

Prethodno opisani pristupi, iako različiti jer se onaj prvi koncentrira na očuvanje varijance a drugi na minimiziranje pogreške, dovode do istih rješenja. Svojstveni vektori koji maksimiziraju varijancu istodobno minimiziraju pogrešku projekcije. Stoga su komplementarni i koriste se ovisno o perspektivi problema koji se analizira.

2.2 PCA u praksi

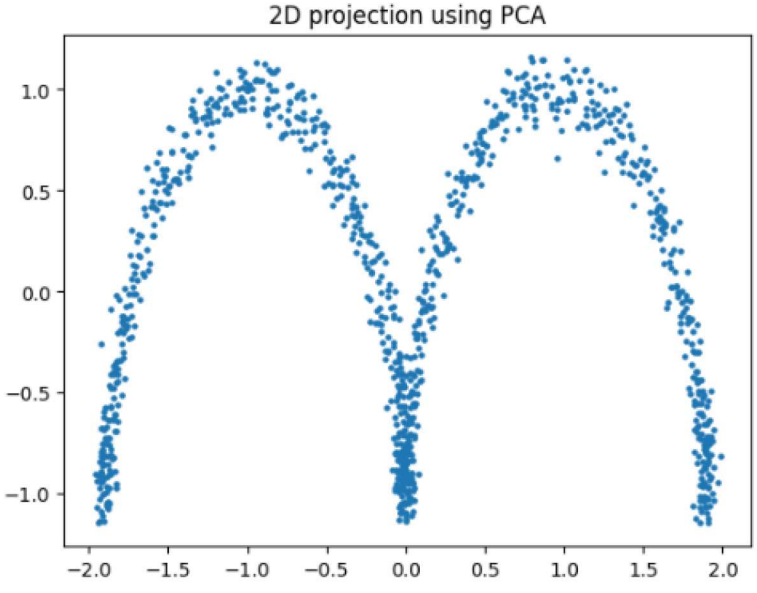
Kako bismo bolje ilustrirali teoretske aspekte PCA opisane u prethodnim dijelovima, demonstrirat ćemo kako ova metoda funkcionira u praksi. U ovom primjeru koristimo spiralne podatke generirane u trodimenzionalnom prostoru. Cilj je prikazati kako PCA može projicirati složene podatke iz višedimenzionalnog prostora u prostor manje dimenzije, uz očuvanje što većeg dijela informacija.

Na slici 2 prikazani su originalni spiralni podaci u 3D prostoru. Ovi podaci tvore kompleksnu trodimenzionalnu spiralu, što otežava njihovu interpretaciju bez dodatnih transformacija.



Slika 2: Originalni spiralni podaci u 3D prostoru.

Primjenom PCA, podaci su projicirani iz trodimenzionalnog prostora u dvodimenzionalni, čime je omogućena lakša vizualizacija. Cilj ove projekcije je maksimalno očuvati varijancu iz originalnog prostora, uz minimalan gubitak informacija. Rezultati ove projekcije prikazani su na slici 3.



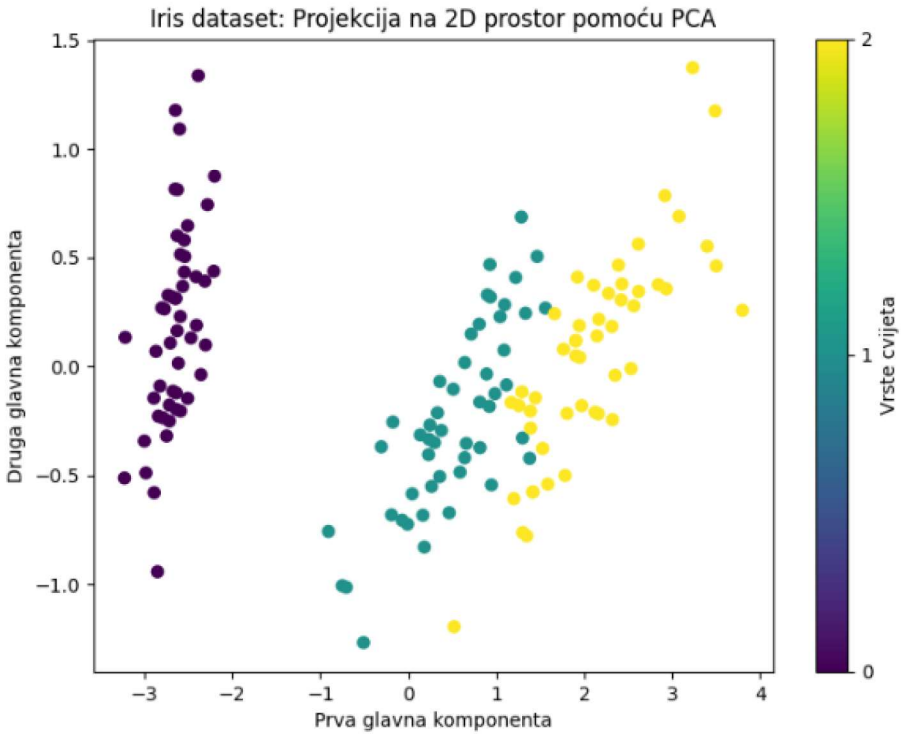
Slika 3: Projekcija spiralnih podataka nakon primjene PCA u 2D prostoru.

Projekcija na 2D prostor omogućila je zadržavanje glavnih karakteristika podataka, dok je treća dimenzija, koja sadrži manje relevantne informacije, eliminirana.

Na slici 3 jasno je vidljivo kako PCA uspješno reducira dimenzionalnost podataka, dok ključne značajke strukture ostaju sačuvane. Ova metoda je osobito korisna kada je cilj vizualizirati složene višedimenzionalne podatke na intuitivan način, kao što je prikazano u ovom primjeru.

Nakon što smo demonstrirali kako PCA funkcionira na umjetno generiranim podacima, logičan slijed bio je primijeniti PCA na stvarnim podacima. Korištenje stvarnih podataka omogućuje nam bolje razumijevanje kako PCA može pomoći u analizi i vizualizaciji kompleksnih dataset-ova. Za ovaj primjer koristili smo poznati *Iris dataset*, koji se često koristi u statistici i strojnome učenju za klasifikacijske zadatke.

Iris dataset sadrži četiri značajke: duljinu i širinu latice, te duljinu i širinu čaške za tri vrste cvijeta: *Setosa*, *Versicolor* i *Virginica*. Svaka od tri vrste cvijeta ima po 50 uzoraka, što znači da ukupno imamo 150 redaka podataka. Cilj primjene PCA bio je smanjiti dimenzionalnost s četiri značajke na dvije glavne komponente, čime smo omogućili vizualizaciju podataka u dvodimenzionalnom prostoru.



Slika 4: Iris dataset: Projekcija na 2D prostor pomoću PCA.

Na slici 4 vidimo rezultat primjene PCA na Iris dataset-u. Boje označavaju različite vrste cvijeta: ljubičasta za *Setosa*, zelena za *Versicolor*, i žuta za *Virginica*. Svaki uzorak predstavlja pojedinačni cvijet u dvodimenzionalnom prostoru definiranom prvom i drugom glavnom komponentom.

Projekcija pokazuje da je prva glavna komponenta uspjela jasno odvojiti vrstu *Setosa* od ostale dvije vrste, dok druga glavna komponenta daje dodatne informacije o razlikama između vrsta *Versicolor* i *Virginica*. Iako se te dvije vrste djelomično preklapaju, PCA je pomogla smanjiti složenost podataka, omogućujući jasniju analizu njihove strukture.

Ovaj primjer jasno demonstrira kako PCA može biti koristan alat za smanjenje di-

menzionalnosti i vizualizaciju podataka. Iako originalni podaci imaju četiri značajke, njihovo sažimanje na dvije glavne komponente omogućuje jednostavnije razumijevanje i usporedbu različitih skupova podataka.

3 Dekompozicija na singularne vrijednosti (SVD)

Dekompozicija na singularne vrijednosti (SVD) je ključna linearna tehnika koja se koristi u mnogim područjima, uključujući pretraživanje podataka, kompresiju, denoising[1] i rješavanje sustava linearnih jednadžbi. Njezina univerzalnost leži u činjenici da se može primijeniti na bilo koju matricu $A \in \mathbb{R}^{m \times n}$, bez obzira na njezin rang.

SVD omogućuje stabilne i robusne numeričke operacije čak i u situacijama gdje tradicionalna svojstvena dekompozicija ne može biti primijenjena. Primjerice, može se koristiti za aproksimaciju velikih i rijetkih matrica[5], kao što je slučaj u sustavima za preporučivanje[3]. Također, SVD pomaže u pronalaženju latentnih struktura[3] u podacima.

Jedna od najvažnijih primjena SVD-a je u analizi podataka, gdje je cilj smanjiti dimenzionalnost uz očuvanje što više relevantnih informacija. Ova metoda nudi optimalnu aproksimaciju niskog ranga, omogućujući pojednostavljenu interpretaciju složenih podataka uz minimalan gubitak informacija. Primjena SVD-a je također široko rasprostranjena u redukciji buke, kompresiji i vizualizaciji podataka, što dodatno potvrđuje njezinu svestranost i korisnost u znanosti o podacima i računalnim znanostima.

3.1 Motivacija i definicija

Dekompozicija na singularne vrijednosti (SVD) motivirana je geometrijskom interpretacijom djelovanja matrice na jediničnu sferu. Kada bilo koja matrica $A \in \mathbb{R}^{m \times n}$ djeluje na jediničnu sferu S u prostoru $\mathbb{R}^n$, slika te sfere pod tom transformacijom postaje hiperelipsoid u prostoru $\mathbb{R}^m$. SVD omogućuje precizan opis ove transformacije, identificirajući glavne smjerove u kojima matrica rasteže ili sabija prostor te intenzitet tih promjena.

Hiperelipsoid je višedimenzionalna generalizacija elipse, a može se definirati kao površina koja se dobiva rastezanjem jedinične sfere duž ortogonalnih smjerova u prostoru $\mathbb{R}^m$. Ti ortogonalni smjerovi definirani su pomoću jediničnih vektora $u_1, \ldots, u_m \in \mathbb{R}^m$, a svako rastezanje opisano je pomoću faktora $\sigma_1, \ldots, \sigma_m$.

Svaka od ovih vrijednosti σ_i pokazuje koliko matrica A rasteže ili sabija prostor duž određenog smjera. Ako je rang matrice A jednak r, tada će točno r od tih vrijednosti biti različito od nule. Osim toga, ako je $m \geq n$, tada najviše n tih vrijednosti može biti različit od nule.

Lijevi singularni vektori $u_1, u_2, \ldots, u_n$ usmjereni su duž glavnih poluosi hiperelipsoida AS, a desni singularni vektori $v_1, v_2, \ldots, v_n$ predstavljaju praslike tih poluosi. Za svaku singularnu vrijednost σ_i vrijedi:

$$Av_i = \sigma_i u_i.$$

U faktorizaciji SVD-a, lijevi singularni vektori nalaze se u stupcima matrice U, dok se desni singularni vektori nalaze u stupcima matrice V. Ova faktorizacija omogućuje optimalnu aproksimaciju niskog ranga i olakšava primjenu SVD-a u redukciji dimenzionalnosti i kompresiji podataka.

SVD možemo formalno definirati kao metodu koja omogućuje dekompoziciju bilo koje matrice $A \in \mathbb{R}^{m \times n}$ u umnožak triju matrica:

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

gdje su:

- $U \in \mathbb{R}^{m \times m}$ – ortogonalna matrica koja sadrži lijeve singularne vektore, koji su svojstveni vektori matrice AA^T.
- $\Sigma \in \mathbb{R}^{m \times n}$ – dijagonalna matrica koja sadrži singularne vrijednosti, koje su kvadratni korijeni svojstvenih vrijednosti matrica $A^T A$ ili AA^T. Singularne vrijednosti su nenegativni skalari koji su poredani u opadajućem redoslijedu:

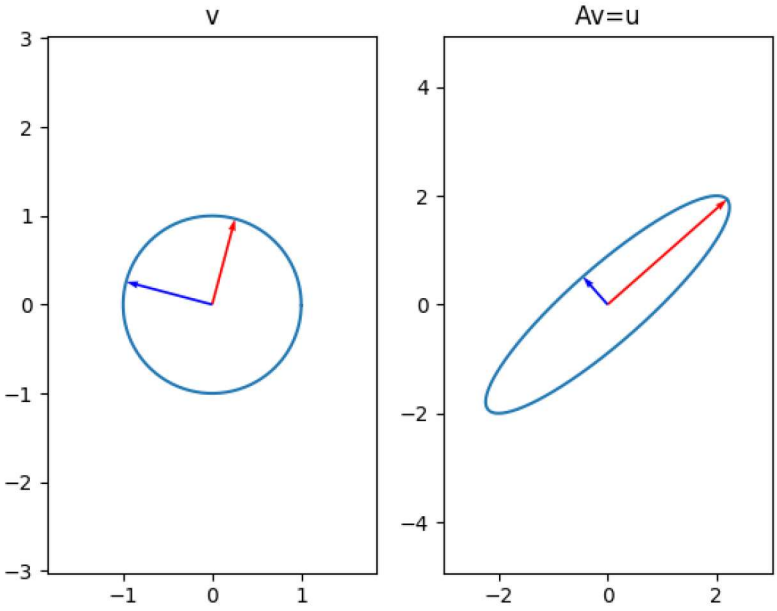
  $$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0,$$

  gdje je $r = \text{rang}(A)$. Veće singularne vrijednosti označavaju smjerove s više informacija ili varijance, dok manje vrijednosti ukazuju na smjerove koji sadrže manje značajne informacije i mogu se zanemariti pri kompresiji podataka.
- $V \in \mathbb{R}^{n \times n}$ – ortogonalna matrica koja sadrži desne singularne vektore, svojstvene vektore matrice $A^T A$. Ovi vektori omogućuju transformaciju podataka u smanjenom prostoru, tj. određuju smjerove u kojima se podaci projiciraju nakon redukcije dimenzionalnosti.

Kako bismo ilustrirali i bolje objasnili napisano, pogledajmo primjer za matricu A definiranu kao:

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix}.$$

Na slici ispod prikazana je transformacija jedinične sfere pomoću matrice A. Lijeva slika prikazuje jediničnu sferu S u izvornom prostoru, s vektorima v_1 i v_2 koji predstavljaju desne singularne vektore. Desna slika prikazuje rezultat djelovanja matrice A na jediničnu sferu, gdje se slika S transformira u elipsoid AS. Glavne osi elipsoida odgovaraju lijevim singularnim vektorima u_1 i u_2, a duljine tih osi su određene pripadajućim singularnim vrijednostima.



Slika 5: Transformacija jedinične sfere pomoću matrice A. Lijeva slika prikazuje vektore v_1 i v_2 u izvornom prostoru, dok desna slika prikazuje elipsoid s lijevim singularnim vektorima u_1 i u_2.

Ova vizualizacija jasno pokazuje kako SVD omogućuje razlaganje matrice na temeljne transformacije tj. rotaciju, rastezanje i eventualno kompresiju, što olakšava interpretaciju učinaka matrice A na prostor podataka.

Teorem 3.1 (Postojanje i jedinstvenost SVD-a). Neka je $A \in \mathbb{R}^{m\times n}$. Tada postoji dekompozicija matrice A u obliku:

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

gdje su U i V ortogonalne matrice, a Σ dijagonalna matrica s nenegativnim dijagonalnim elementima $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$, gdje je $r = \text{rang}(A)$. Ova dekompozicija je jedinstvena do predznaka singularnih vektora.

Dokaz. Dokaz se oslanja na svojstvene vrijednosti i svojstvene vektore simetričnih matrica AA^T i A^TA.

1. Definirajmo simetrične matrice:

$$AA^T \quad \text{i} \quad A^TA.$$

Budući da su ove matrice simetrične, prema *Spektralnom teoremu* imaju ortogonalne svojstvene vektore i nenegativne svojstvene vrijednosti.

2. Ortogonalne matrice U i V sadrže svojstvene vektore matrica AA^T i A^TA:

$$AA^T = U\Lambda U^T \quad \text{te} \quad A^TA = V\Gamma V^T,$$

gdje su Λ i Γ dijagonalne matrice sa svojstvenim vrijednostima.

3. Singularne vrijednosti matrice A definirane su kao kvadratni korijeni svojstvenih vrijednosti:

$$\sigma_i = \sqrt{\lambda_i}, \quad i = 1, \ldots, r.$$

4. Matricu A možemo sada faktorizirati kao:

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

gdje je Σ dijagonalna matrica koja sadrži singularne vrijednosti σ_i.

Ova dekompozicija je jedinstvena do predznaka vektora u matricama U i V.

■

Propozicija 3.1. Neka je $A = U\Sigma V^T$ singularna dekompozicija matrice A. Neka je točno r njenih singularnih vrijednosti različito od nule. Označimo sa u_i i v_i stupce matrica U i V. Tada vrijedi:

(a) $r(A) = r$,

(b) $\ker A = \{v_{r+1}, \ldots, v_n\}$,

(c) $\operatorname{Im} A = \{u_1, \ldots, u_r\}$,

(d) $A = U_r \Sigma_r V_r^T$, gdje je Σ_r gornja lijeva podmatrica matrice Σ reda r, dok su U_r i V_r matrice sastavljene od prvih r stupaca matrica U i V,

(e) $\|A\|_F^2 = \sigma_1^2 + \sigma_2^2 + \cdots + \sigma_r^2$,

(f) $\|A\|_2 = \sigma_1$.

Matrica Σ_r očito ima rang r. Množenje s regularnim matricama ne mijenja rang, pa vrijedi $r(A) = r$.

Za tvrdnje (b) i (c) primijetimo da su vektori $\{v_{r+1}, \ldots, v_n\}$ baza za jezgru, a vektori $\{u_1, \ldots, u_r\}$ baza za sliku matrice A. Iz dekompozicije $A = U\Sigma V^T$ također slijedi:

$$AV = U\Sigma.$$

Za $i \leq r$, $Av_i = \sigma_i u_i$, dok za $i > r$ vrijedi $Av_i = 0$, što dokazuje tvrdnje o jezgri i slici.

Tvrdnja (d) slijedi iz činjenice da je A ekvivalentna sa $U_r \Sigma_r V_r^T$, gdje je Σ_r podmatrica koja sadrži sve nenula singularne vrijednosti.

Frobeniusova norma zadovoljava:

$$\|A\|_F^2 = \|\Sigma\|_F^2 = \sigma_1^2 + \sigma_2^2 + \cdots + \sigma_r^2,$$

što dokazuje tvrdnju (e).

Za (f) koristimo definiciju operatorne norme:

$$\|A\|_2 = \max_{\|x\|=1} \|Ax\| = \sigma_1,$$

gdje je σ_1 najveća singularna vrijednost. Operatorna norma mjeri najveće rastezanje koje matrica A može proizvesti u bilo kojem smjeru. To znači da, među svim jediničnim vektorima x, matrica A rasteže onaj vektor koji odgovara najvećoj singularnoj vrijednosti σ_1. Stoga vrijedi $\|A\|_2 = \sigma_1$.

■

3.2 Odabir između punog i reduciranog SVD-a

Dekompozicija na singularne vrijednosti (SVD) može se koristiti u dva oblika: *puni SVD* i *reducirani SVD*, ovisno o potrebama analize i računalnim resursima. Ovdje pojašnjavamo razlike i prikazujemo situacije kada je koji oblik primjenjiv.

Puni SVD

Puni SVD uzima u obzir sve singularne vrijednosti i vektore matrice. Za matricu $A \in \mathbb{R}^{m \times n}$, dekompozicija je:

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

gdje je $U \in \mathbb{R}^{m \times m}$ matrica lijevih singularnih vektora, $\Sigma \in \mathbb{R}^{m \times n}$ dijagonalna matrica sa svim singularnim vrijednostima i $V \in \mathbb{R}^{n \times n}$ matrica desnih singularnih vektora.

Kada koristiti puni SVD?

- Kada je potrebno očuvati sve informacije: Puni SVD je koristan kada se žele zadržati sve informacije, bez obzira na računalne troškove.
- Za detaljne analize podataka: Kada je cilj interpretirati sve dimenzije i njihove međuodnose.
- Kada nema ograničenja u resursima: Puni SVD je računalno zahtjevan i koristi se ako vrijeme i resursi nisu kritični faktori.

Reducirani SVD

Reducirani SVD koristi samo najvažnije singularne vrijednosti i odgovarajuće singularne vektore. Aproksimacija reduciranog SVD-a za rang k izgleda ovako:

$$A_k = U_k \Sigma_k V_k^T,$$

gdje je $U_k \in \mathbb{R}^{m \times k}$ matrica koja sadrži samo prvih k lijevih singularnih vektora, $\Sigma_k \in \mathbb{R}^{k \times k}$ dijagonalna matrica s prvih k najvećih singularnih vrijednosti i $V_k \in \mathbb{R}^{n \times k}$ matrica s prvih k desnih singularnih vektora.

Kada koristiti reducirani SVD?

- Kada je cilj smanjenje dimenzionalnosti: Reducirani SVD je idealan za pronalaženje najvažnijih značajki podataka.
- Za ubrzanje algoritama: Često se koristi kada je brzina ključna, jer reducira količinu podataka uz minimalan gubitak informacija.
- Za velike matrice: Kod velikih i rijetkih matrica, kao što su one u preporučivačkim sustavima, reducirani SVD omogućuje efikasnu aproksimaciju.

Napomena 3.1. SVD također omogućuje optimalnu aproksimaciju matrice niskog ranga. Ako uzmemo samo prvih k najvećih singularnih vrijednosti i odgovarajuće vektore, dobivamo aproksimaciju:

$$A_k = U_k \Sigma_k V_k^T,$$

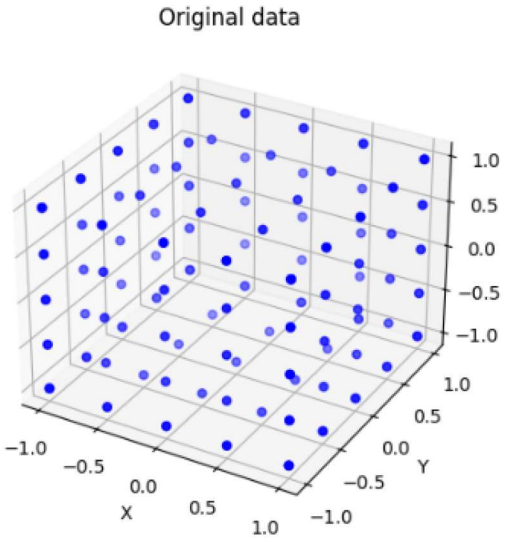
gdje su U_k, Σ_k i V_k podmatrice koje sadrže samo prvih k komponenti. Ova aproksimacija minimizira Frobeniusovu normu razlike između originalne matrice i njezine aproksimacije:

$$\|A - A_k\|_F = \min_{\text{rang}(B)=k} \|A - B\|_F.$$

3.3 SVD u praksi

Nakon što smo opisali teorijske osnove SVD-a i geometrijsku interpretaciju, sada ćemo prikazati praktičnu primjenu SVD metode. U ovom primjeru koristimo *simetrično generirane podatke na površini kocke*, a zatim pomoću SVD-a reduciramo dimenzionalnost ovih podataka kako bismo dobili njihovu 2D projekciju.

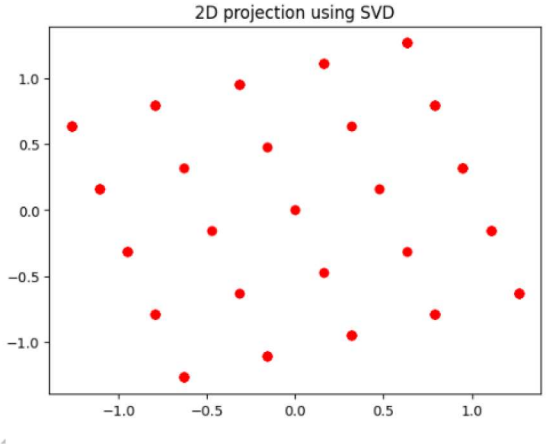
Podaci su generirani tako da predstavljaju točke ravnomjerno raspoređene po površini kocke, što omogućuje jednostavno razumijevanje kako SVD radi s pravilnim i simetričnim strukturama. Originalni podaci prikazani su na sljedećoj slici:



Slika 6: Originalni simetrični podaci na površini kocke.

Primijenili smo SVD kako bismo smanjili dimenzionalnost podataka s 3D na 2D. SVD nam omogućuje očuvanje najvažnije informacije o podacima, uz zanemarivanje redundantnih komponenti. U ovom primjeru smo uzeli samo dvije najveće singularne vrijednosti i odgovarajuće vektore.

Nakon redukcije dimenzionalnosti, dobiveni podaci prikazani su u *2D prostoru* na sljedećoj slici:



Slika 7: Reducirani podaci s površine kocke u 2D prostoru.

Korištenjem SVD-a na ovim simetričnim podacima, pokazali smo kako algoritam može pronaći ključne smjerove varijance i sažeti podatke na način da se očuva većina korisnih informacija.

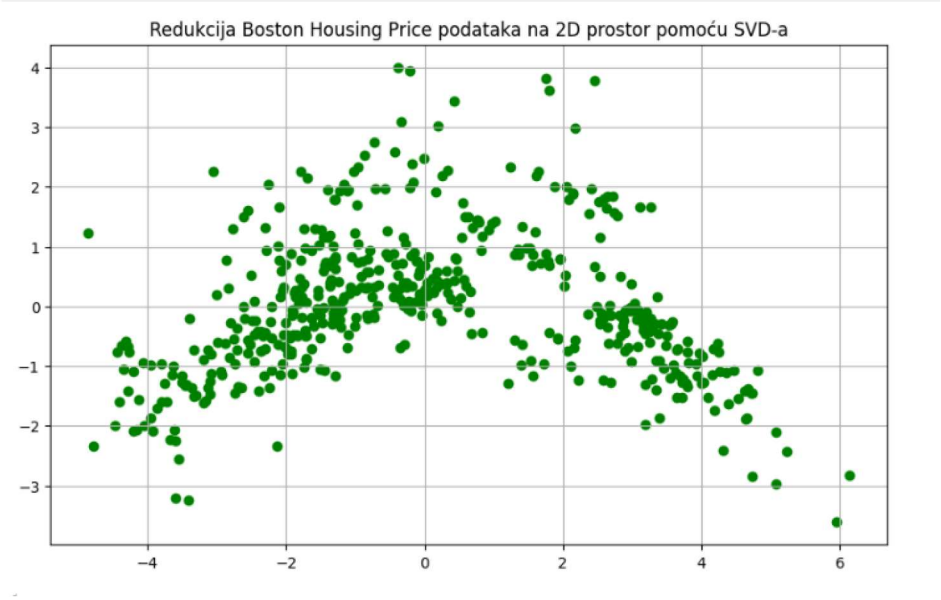
Kocka u 3D prostoru imala je sve tri dimenzije odjednom prikazane. Nakon primjene SVD-a:

- Eliminirali smo redundantnu dimenziju.
- Podaci su prikazani u 2D prostoru, ali su ključne karakteristike zadržane.

Primjena SVD-a u ovakvim situacijama korisna je jer omogućuje jednostavnije vizualizacije, kao i smanjenje složenosti modela bez gubitka važnih informacija.

Kako bismo dodatno istražili primjenu dekompozicije na singularne vrijednosti u praksi, koristimo stvarni skup podataka *Boston housing prices*. Ovaj skup podataka sadrži 506 uzoraka i ukupno 13 značajki.

Naš cilj je smanjiti dimenzionalnost ovog skupa podataka s 13 značajki na dvodimenzionalni prostor, kako bismo omogućili jednostavniju vizualizaciju, a istovremeno sačuvali što više ključnih informacija. Primjena SVD-a omogućava ovu redukciju uz očuvanje glavnih struktura u podacima.



Slika 8: Redukcija *Boston Housing Prices* podataka na 2D prostor pomoću SVD-a.

Na slici 8 prikazali smo rezultat projekcije podataka iz višedimenzionalnog prostora u 2D prostor. Svaka točka na grafu predstavlja jedan uzorak iz skupa podataka. Iako je broj dimenzija smanjen s 13 na 2, vidljivi su obrasci u podacima, što sugerira da su sačuvane najvažnije informacije.

4 Usporedba PCA i SVD-a i odabir optimalne dimenzije

Kada se razmatraju metode za redukciju dimenzionalnosti podataka, analiza glavnih komponenti (PCA) i dekompozicija na singularne vrijednosti (SVD) su dvije najistaknutije tehnike. Iako dijele sličan cilj odnosno smanjenje dimenzionalnosti uz očuvanje relevantnih informacija,postoje značajne razlike u njihovoj primjeni, matematičkoj podlozi i ograničenjima.

4.1 Razlike između PCA i SVD

- **Cilj metode:** PCA maksimizira varijancu duž ortogonalnih smjerova kako bi sažela podatke, dok je SVD općenitija metoda za dekompoziciju matrica bez obzira na njihov rang.
- **Primjena:** PCA se koristi prvenstveno za analizu i vizualizaciju podataka, dok je SVD primjenjiv u širem spektru zadataka, uključujući kompresiju podataka i sustave za preporučivanje.
- **Matematička podloga:** PCA se temelji na svojstvenim vrijednostima i vektorima kovarijacijske matrice, dok SVD koristi singularne vrijednosti i vektore za dekompoziciju izvorne matrice.
- **Svojstva:** PCA je ograničena na pozitivno polu-definitivne matrice, dok SVD može raditi s bilo kojom matricom, uključujući i ne-kvadratne.

4.2 Izazovi i ograničenja korištenja PCA i SVD

Iako su PCA i SVD moćni alati, njihova primjena dolazi s određenim izazovima:

Gubitak informacija: Jedan od najvećih izazova pri redukciji dimenzionalnosti je rizik gubitka relevantnih informacija. Iako metode pokušavaju sažeti podatke uz očuvanje ključnih struktura, neka informacija neizbježno se gubi, osobito kada se zadržava samo mali broj glavnih komponenti ili singularnih vrijednosti.

Pretjerana kompresija: Preveliko smanjenje dimenzionalnosti može rezultirati gubitkom važnih uzoraka ili varijacija u podacima, što može negativno utjecati na performanse modela. U praksi je ključno pronaći ravnotežu između smanjenja složenosti i očuvanja informacija.

Računska složenost: PCA i SVD mogu biti računalno zahtjevni za velike skupove podataka. Iako reducirani SVD pomaže u rješavanju ovog problema, potreba za računalnim resursima ostaje važan faktor.

Osjetljivost na šum: Podaci često sadrže šum, koji može utjecati na performanse algoritama. PCA i SVD mogu nenamjerno naglasiti šum, što otežava interpretaciju rezultata. Prethodna obrada podataka, poput filtriranja šuma, ključna je za osiguranje kvalitetnih rezultata.

Međutim, u situacijama kada podaci pokazuju kompleksne nelinearne odnose, ove linearne tehnike mogu postati ograničene. Iako PCA i SVD osiguravaju smanjenje dimenzionalnosti uz očuvanje što više moguće varijance, često se suočavaju s problemima kada ne mogu adekvatno uhvatiti nelinearne strukture podataka[1]. Stoga je važno razmotriti dodatne pristupe koji mogu pružiti bolje rezultate u takvim slučajevima. Kombinacija linearnih i nelinearnih metoda može donijeti dodatnu vrijednost u analizi, osobito u scenarijima gdje je očuvanje i linearnosti i složenih obrazaca ključno.

Primjena naprednijih tehnika, poput t-SNE ili autoenkodera, može nadopuniti PCA i SVD, omogućujući bolje razumijevanje i vizualizaciju podataka.

t-SNE: Tehnika t-SNE[11] (t-distributed Stochastic Neighbor Embedding) je nelinearna tehnika redukcije dimenzije namijenjena za vizualizaciju podataka u niže-dimenzionalnim prostorima. Pomaže u grupiranju sličnih uzoraka, ali može biti računalno intenzivna i osjetljiva na parametre kao što su broj iteracija i perpleksnost. Omogućuje bolju vizualizaciju podataka grupirajući slične uzorke. Iako pruža impresivne rezultate, izazov ostaje u pronalaženju optimalnih parametara i visokoj računalnoj zahtjevnosti.

Autoenkoderi: Autoenkoderi[12] su neuronske mreže koje uče reprezentaciju podataka kroz proces kompresije(kodiranja) i dekompresije(dekodiranja). Korisni su za detekciju anomalija i prepoznavanje nelinearnih obrazaca, ali zahtijevaju značajan trud za treniranje i optimizaciju arhitekture. Nude veću fleksibilnost u prepoznavanju nelinearnih obrazaca. Koriste se u detekciji anomalija i kompresiji, no njihova implementacija zahtijeva vrijeme i pažljivo podešavanje arhitekture.

Kombinacija linearnih i nelinearnih tehnika: Kombiniranje PCA ili SVD s nelinearnim tehnikama može omogućiti bolje rezultate u analizi kompleksnih podataka. Na primjer, PCA se može koristiti za inicijalno smanjenje dimenzija, nakon čega t-SNE poboljšava vizualizaciju. Alternativno, autoenkoderi mogu identificirati složene obrasce koje linearne metode ne mogu prepoznati. Ova kombinacija nudi uravnotežen pristup koji osigurava očuvanje ključnih informacija, dok istovremeno omogućuje otkrivanje složenih nelinearnih struktura.

4.3 Kako odabrati najbolju dimenziju?

Odabir optimalnog broja dimenzija predstavlja ključni izazov. Premalo dimenzija može dovesti do gubitka informacija, dok previše dimenzija otežava analizu i povećava računalne troškove.

Kumulativna varijanca: Za PCA, graf kumulativne varijance pomaže identificirati optimalan broj glavnih komponenti:

$$\text{Kumulativna varijanca}(n) = \frac{\sum_{i=1}^{n} \lambda_i}{\sum_{i=1}^{d} \lambda_i},$$

gdje λ_i označava svojstvene vrijednosti, a d ukupan broj značajki. U praksi, optimalan broj komponenti je onaj koji objašnjava barem 90% ukupne varijance.

Graf "koljena": Graf „koljena"[9] omogućuje vizualni uvid u točku gdje dodatne komponente više ne donose značajne koristi.

Primjena na SVD: Kod SVD-a se biraju najveće singularne vrijednosti koje odgovaraju najvažnijim strukturama u podacima, a manje vrijednosti se zanemaruju kako bi se postigla veća učinkovitost.

Praktične smjernice za odabir dimenzija

- Analizirajte graf kumulativne varijance i odaberite broj komponenti do točke „koljena".
- Zadržite broj dimenzija koji objašnjava najmanje 90% varijance.
- Za specifične zadatke, poput vizualizacije, može biti dovoljno 2 ili 3 dimenzije.

Odabir optimalnog broja dimenzija omogućuje balans između preciznosti i računalne učinkovitosti, čime se postiže najbolji mogući rezultat u analizi podataka.

5 Zaključak

U ovom radu istražili smo dvije ključne metode za redukciju dimenzionalnosti podataka: *analizu glavnih komponenti* (PCA) i *dekompoziciju na singularne vrijednosti* (SVD). Obje metode pokazale su se korisnima za pojednostavljenje analize, kompresiju i vizualizaciju podataka u prostoru manje dimenzionalnosti, uz očuvanje ključnih informacija.

PCA projicira podatke na smjerove s najvećom varijancom, dok SVD omogućuje dekompoziciju matrice bez obzira na njezin rang, pružajući širu primjenu u kompresiji, prepoznavanju obrazaca i sustavima za preporučivanje. Demonstrirali smo primjenu ovih tehnika na umjetno generiranim podacima, poput spiralnih i kockastih struktura, te na stvarnim podacima o cijenama nekretnina u Bostonu. Ove analize pokazale su kako SVD i PCA pomažu u vizualizaciji i interpretaciji složenih podataka smanjenjem njihove dimenzionalnosti.

Bitan aspekt ovog rada bio je odabir optimalnog broja dimenzija. Kod PCA, analizirali smo graf kumulativne varijance kako bismo odredili odgovarajući broj glavnih komponenti. Kod SVD-a smo istaknuli važnost reduciranog oblika kako bi se postigla računalna učinkovitost bez značajnog gubitka informacija.

Iako obje metode nude značajne prednosti, imaju i svoja ograničenja. PCA je osobito korisna za identificiranje ortogonalnih smjerova koji maksimiziraju varijancu, ali je osjetljiva na šum u podacima. S druge strane, SVD je fleksibilniji alat primjenjiv na raznovrsne zadatke, ali može biti računalno zahtjevan za velike skupove podataka.

Zaključno, PCA i SVD su neophodni alati za analizu podataka i smanjenje dimenzionalnosti. Pravilnim odabirom metode i broja dimenzija može se poboljšati interpretacija podataka, pojednostaviti računalna obrada i očuvati ključne informacije. U budućnosti, istraživanje bi se moglo proširiti kombiniranjem linearnih metoda, poput PCA i SVD-a, s nelinearnim tehnikama kao što su t-SNE i autoenkoderi. Ova kombinacija omogućila bi još bolju analizu složenih podataka, osobito kada linearne metode nisu dovoljne za hvatanje nelinearnih struktura u podacima.

Bibliografija

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

[2] Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer, 2006.

[3] Leskovec, J., Rajaraman, A., Ullman, J., *Mining of Massive Datasets*, Cambridge University Press, 2014.

[4] Trefethen, L. N., Bau, D., *Numerical Linear Algebra*, SIAM, 1997.

[5] Golub, G. H., Van Loan, C. F., *Matrix Computations*, Johns Hopkins University Press, 1996.

[6] Abdi, H., Williams, L. J., "Principal Component Analysis", *Wiley Interdisciplinary Reviews: Computational Statistics*, 2010.

[7] Jolliffe, I. T., *Principal Component Analysis*, Springer, 2016.

[8] Strang, G., *Introduction to Linear Algebra*, Wellesley-Cambridge Press, 2009.

[9] Hastie, T., Tibshirani, R., Friedman, J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2009.

[10] Meyer, C. D., *Matrix Analysis and Applied Linear Algebra*, SIAM, 2000.

[11] Van der Maaten, L., Hinton, G., "Visualizing Data using t-SNE", *Journal of Machine Learning Research*, 2008.

[12] Goodfellow, I., Bengio, Y., Courville, A., *Deep Learning*, MIT Press, 2016.

[13] Scitovski, R., *Numerička matematika*, drugo izdanje, Odjel za matematiku, Sveučilište J. J. Strossmayera u Osijeku, 2004.