Segmentacija i klasifikacija pomoću difuzijskih preslikavanja

Olić, Hrvoje

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Hrvoje Olić

SEGMENTACIJA I KLASIFIKACIJA POMOĆU DIFUZIJSKIH PRESLIKAVANJA

Diplomski rad

Voditelj rada:
prof. dr. sc. Zlatko Drmač

Zagreb, 2023.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Svima koji su bili uz mene na ovom putu, hvala vam.

Sadržaj

Uvod

Svakodnevni život teško je zamisliti bez korištenja računala i raznih aplikacija koje se zasnivaju na analizi podataka. Samo danas smo sigurno svi na neki način koristili dostupne podatke u donošenju odluka - provjerili smo vremensku prognozu, planirali put do nekog odredišta, birali što bismo mogli ručati, kupovali ili naručivali, ili možda provjeravali stanje u svijetu na nekom od portala. Činjenica je da nam je analiza podataka od neizmjerne važnosti, iako toga često nismo ni svjesni. Sam napredak podatkovne znanosti nam je stoga također bitan, jer nam može uvelike olakšati svakodnevne aktivnosti.

Količina podataka i informacija koji nas svakodnevno okružuju je enormna i kao ljudi nismo ih u mogućnosti sve procesirati, shvatiti i zapamtiti. Zato koristimo računala, ali čak i današnja računala, koja imaju iznimne računske sposobnosti nisu sposobna pratiti rast podataka. Dobro poznati *Mooreov zakon* kaže da se broj tranzistora u računalnim čipovima udvostruči svake dvije godine (doduše, zakon nije fizički, već empirijski, ali se tijekom godina pokazao ispravnim), no količina podataka raste još brže. Primjerice, prema podacima dostupnima na Forbesu (listopad 2023.), na internetu je preko milijardu stranica indeksiranih na Google-u (stvarni broj se procjenjuje na oko 5 milijardi, prema WorldWideWebsize), i gotovo 5 milijardi ljudi koristi internet, te se predviđa još brži rast broja stranica i korisnika. Naravno, internet je samo jedan od rastućih izvora informacija, računala trebaju obrađivati i ostale, također velike izvore podataka. A s obzirom da se i internet i ostali izvori informacija mijenjaju i rastu, naša računala trebaju nekako moći pratiti takav rast na dosjetljiv način - ako ne možemo povećati računsku moć fizički, onda ju možemo poboljšati domišljatim metodama za analizu podataka.

Jedan od glavnih ciljeva podatkovne znanosti je otkriti kako u podacima izdvojiti one najbitnije, koji čine srž onoga što proučavamo. Primjerice, ako želimo pohraniti fotografiju od 24 megapiksela (tj. 6000×4000 piksela, što je prosječna kvaliteta fotografije za današnje standarde), radimo s 24 milijuna podataka. No vrlo često je puno tih podataka zanemarivo, te se približno ista slika može dobiti iz puno manje količine podataka, korištenjem neke od metoda kompresija slike. I na primjeru možemo vidjeti kako jednostavna metoda kompresije bazirana na jednostavnoj faktorizaciji matrice znatno smanjuje količinu informacije, ali i dalje čuva kvalitetu slike.

Polazna fotografija

Aproksimacija ranga 50

Aproksimacija ranga 100

Čisto za usporedbu, količina podataka koja nam je potrebna za pohranu prve fotografije (dimenzije 1200× 1200 u RGB formatu) je oko 4 milijuna, za drugu, na kojoj se vidi blago smanjenje kvalitete, oko 400 tisuća, a za posljednju 750 tisuća, iako je golim okom vrlo teško vidjeti razliku u odnosu na prvu, potpunu sliku. Dakle, uz značajno smanjenje količine podataka za pohranu možemo postići isti rezultat, samo ih trebamo pametno reprezentirati. U ovom radu ćemo se baviti algoritmom koji ima upravo taj cilj - difuzijskim preslikavanjima.

Difuzijska preslikavanja su algoritam za učenje mnogostrukosti koji se također može koristiti za smanjenje dimenzije problema ili za ekstrakciju značajki u podacima. Osnovna ideja je podatke zamisliti, odnosno reprezentirati kao graf, gdje su susjedne točke blizu onoliko koliko su u nekoj mjeri slične. Taj prikaz koristimo kako bismo podatke preslikali u novi prostor, u kojemu bi standardna udaljenost preslikanih točaka odgovarala udaljenostima originalnih podataka na grafu.

Prednost difuzijskih preslikavanja nad nekim poznatim metodama redukcije dimenzije (primjerice, analiza glavnih komponenti - *PCA*) je ta da ona mogu otkriti jednostavnu strukturu podataka u visokodimenzionalnom prostoru, čak i ako je ona nelinearna - ako imamo podatke dimenzije tisuću, ali svi leže npr. na sferi dimenzije 2, onda možemo podatke prikazati u dvodimenzionalnom prostoru, štedeći ogromnu količinu memorije, a žrtvujemo malu količinu informacija, koja bi bila znatno veća kod nekih drugih metoda, npr. onih koji pretpostavljaju da su strukture na kojima podaci leže linearne.

Osim toga, difuzijska preslikavanja koriste grafovsku strukturu u još većoj mjeri - osim za samo preslikavanje u nove koordinate, koristimo i činjenicu da grafovi nose puno više informacija, primjerice njihovu povezanost. Tu značajku difuzijska preslikavanja koriste kako bi iz poznavanja odnosa među podacima, pomoću šetnje na grafu kroz njih, otkrila globalnu strukturu prostora u kojem leže podaci. To će nam biti od velike koristi u primjenama gdje se pravilnost u odnosu podataka golim okom teško može vidjeti, ali se može izraziti nekom transformacijom.

U ovom radu ćemo dati pregled teorije i primjene difuzijskih preslikavanja na sljedeći način: u prvom poglavlju dajemo teorijski pregled Markovljevih lanaca - slučajnih procesa koji formalno definiraju i daju teorijsku podlogu i intuiciju za matematičku definiciju difuzijskih preslikavanja. Prvo je poglavlje u većem dijelu izloženo prema materijalima s kolegija Markovljevi lanci koji se predaje na Matematičkom odsjeku PMF-a, te je rađeno po materijalima profesora Vondračeka. U drugom poglavlju prezentiramo metode numeričke matematike koje nam omogućavaju da teoriju provedemo u djelo - opisat ćemo i dati teorijsko opravdanje za algoritme koje ćemo koristiti u implementaciji difuzijskih preslikavanja. Ovo poglavlje je velikim dijelom obrađeno po uzoru na gradivo Numeričke analize 1, prema materijalima profesora Drmača. U trećem poglavlju definiramo sama difuzijska preslikavanja, opisujemo njihova svojstva te dajemo uvid u njihovo djelovanje kao i metode kojima ih možemo poboljšati. U četvrtom poglavlju se dotičemo lokalno linearnih ulaganja - još jedne metode za redukciju dimenzije koja ima slična svojstva kao i difuzijska preslikavanja, te primjenjujemo opisane metode u klasifikaciji. U posljednjem, petom poglavlju, opisujemo kako se difuzijska preslikavanja koriste u segmentaciji, a posebnu pažnju posvećujemo Nyströmovoj aproksimaciji - jednoj od vrlo popularnih metoda za aproksimativno računanje difuzijskih preslikavanja onda kada je dimenzija problema takva da standardan račun nije vremenski izvediv.

Poglavlje 1

Markovljevi lanci

Markovljevi lanci su jedan od osnovnih stohastičkih modela, te su kao takvi sveprisutni u znanosti - osim u statistici, koriste se u računalnoj znanosti, prvenstveno za analizu podataka, u fizici kao jedan od glavnih modela u statističkoj mehanici, u bioinformatici kao modeli bioloških nizova.

Postoji nekoliko vrsta Markovljevih lanaca, u ovisnosti o broju elemenata skupa u kojem poprimaju vrijednosti i o diskretnosti, odnosno neprekidnosti vremenskih koraka u kojima promatramo Markovljev lanac. U ovom radu će nam centralni objekt biti Markovljevi lanci u diskretnom vremenu s diskretnim (konačnim ili prebrojivim) skupom stanja.

Glavna ideja iza teorije Markovljevih lanaca je da oni predstavljaju iduću stepenicu u modeliranju slučajnih varijabli u odnosu na nezavisne procese. Markovljevi lanci stoga opisuju modele u kojima je idući korak ovisan samo o trenutnom stanju, a ne o prošlosti, odnsnosno o tome kako smo u trenutno stanje došli - formalnu definiciju ćemo dati kasnije.

Osnovni cilj u teoriji Markovljevih lanaca je procijeniti njihovo dugoročno ponašanje te odrediti stacionarnu distribuciju lanca, jer nam ona daje vrlo efektivan alat za analizu ponašanja lanca kada vrijeme t teži k beskonačnosti.

Najviše ćemo se baviti slučajnim šetnjama na težinskim usmjerenim grafovima. Ideja je da kada prikažemo skup podataka kao težinski graf, gdje težine bridova predstavljaju vjerojatnosti prijelaza iz jednog vrha u drugi, takav proces možemo modelirati kao Markovljev lanac.

U ovom poglavlju, formalno definiramo Markovljev lanac i uvodimo osnovne pojmove potrebne za razumijevanje teorije iza metoda koje implementiramo. U drugom dijelu fokusiramo se na glavne rezultate i svojstva lanaca koje promatramo, kako bi opravdali korištene metode u kasnijim poglavljima.

1.1 Definicije i osnovni pojmovi

Definicija 1.1.1. *Neka je S konačan ili prebrojiv skup (kojeg ćemo od sada nadalje nazivati* skup ili prostor stanja*). Slučajan proces s diskretnim vremenom i prostorom stanja S je familija $X = (X_n : n \in \mathbb{N}_0)$ slučajnih varijabli (ili elemenata) definiranih na nekom vjerojatnosnom prostoru $(\Omega, \mathcal{F}, \mathbb{P})$ s vrijednostima u S. Dakle, $\forall n \geq 0$ je $X_n : \Omega \to S$ slučajna varijabla.*

Napomena: U pravilu će skup stanja S biti $\mathbb{N}$ ili neki njegov podskup - jer je stanja konačno ili prebojivo mnogo, možemo ih enumerirati u $\mathbb{N}$.

Definicija 1.1.2. *Slučajan proces X definiran na vjerojatnosnom prostoru $(\Omega, \mathcal{F}, \mathbb{P})$ s vrijednostima u skupu stanja S je* Markovljev lanac *ako vrijedi:*

$$\mathbb{P}(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, ..., X_0 = i_0) = \mathbb{P}(X_{n+1} = j \mid X_n = i) \tag{1.1}$$

za svaki $n \geq 0$ i za sve $i_0, ..., i_{n-1}, i, j \in S$ za koji su obje uvjetne vjerojatnosti dobro definirane.

Primjer 1.1.1. Jedan od najčešćih primjera Markovljevih lanaca je primjena kumulativnog računa na neki nezavisan jednako distribuiran niz. Primjerice, neka je $(Y_n)_{n\geq 0}$ nezavisan jednako distribuiran niz slučajnih varijabli s vrijednostima u $\mathbb{N}$. Tada će niz $(X_n)_{n\geq 0}$ definiran kao $X_n = \sum_{i=0}^{n} Y_i$ činiti Markovljev lanac. Pokažimo to:

Za proizvoljne $n \geq 0$ i $i_0, ..., i_{n-1}, i, j \in S$ imamo (uz pretpostavku da su $i_0, ..., i_{n-1}, i, j$ takvi da su sve vjerojatnosti s kojima radimo pozitivne):

$$\mathbb{P}(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, ..., X_0 = i_0) = \frac{\mathbb{P}(X_{n+1} = j, X_n = i, X_{n-1} = i_{n-1}, ..., X_0 = i_0)}{\mathbb{P}(X_n = i, X_{n-1} = i_{n-1}, ..., X_0 = i_0)} =$$
$$\frac{\mathbb{P}(Y_0 = i_0, Y_1 = i_1 - i_0, ..., Y_n = i_n - i_{n-1}, Y_{n+1} = j - i)}{\mathbb{P}(Y_0 = i_0, Y_1 = i_1 - i_0, ..., Y_n = i - i_{n-1})} = \text{(nezavisnost)} =$$
$$\frac{\mathbb{P}(Y_0 = i_0)\mathbb{P}(Y_1 = i_1 - i_0)\mathbb{P}(Y_n = i_n - i_{n-1})\mathbb{P}(Y_{n+1} = j - i)}{\mathbb{P}(Y_0 = i_0)\mathbb{P}(Y_1 = i_1 - i_0)\mathbb{P}(Y_n = i - i_{n-1})} = \mathbb{P}(Y_{n+1} = j - i).$$

Time smo lijevu stranu definicije 1.1 sveli na jednostavan oblik. Napravimo isto i s desnom:

$$\mathbb{P}(X_{n+1} = j \mid X_n = i) = \frac{\mathbb{P}(X_{n+1} = j, X_n = i)}{\mathbb{P}(X_n = i)} = \frac{\mathbb{P}(Y_{n+1} = j - i, X_n = i_n)}{\mathbb{P}(X_n = i)} = \text{(nezavisnost)} =$$
$$\frac{\mathbb{P}(Y_{n+1} = j - i)\mathbb{P}(X_n = i)}{\mathbb{P}(X_n = i)} = \mathbb{P}(Y_{n+1} = j - i).$$

Vidimo da imamo definicijsku jednakost - dakle, imamo Markovljev lanac.

Naravno, još jednostavniji primjer Markovljevog lanca je niz nezavisnih jednakodistribuiranih slučajnih varijabli. Ipak, to je puno jednostavnija struktura, te općenito, Markovljev lanac neće činiti nezavisan jednakodistribuiran (n.j.d.) niz. Promotrimo sljedeći primjer:

Primjer 1.1.2. Neka je $(Y_n)_{n\geq 0}$ iz prethodnog primjera niz modificiranih Bernoullijevih slučajnih varijabli s parametrom $p = \frac{1}{2}$i vrijednostima $\{-1, 1\}$, to jest, neka je:

$$Y_n \sim \begin{pmatrix} -1 & 1 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}, \ n \geq 0.$$

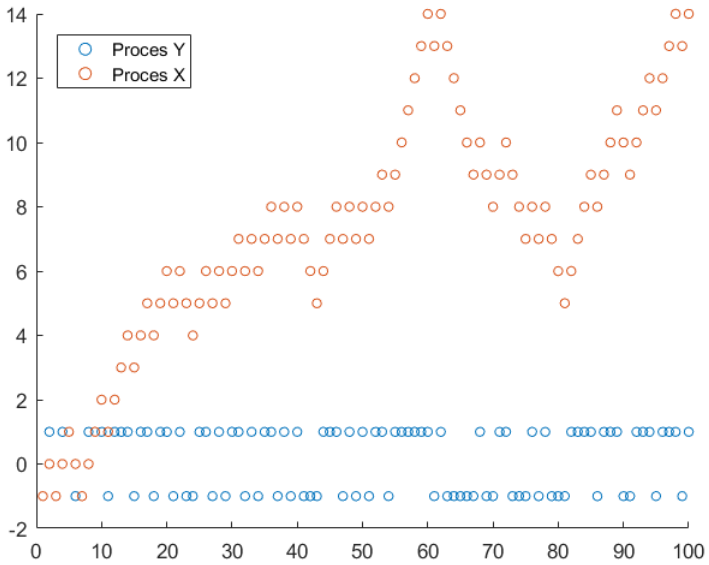
Tada vidimo da $(X_n)_{n\in\mathbb{N}}$ nije n.j.d. jer imamo:

$$\mathbb{P}(X_0 = 2) = \mathbb{P}(Y_0 = 2) = 0,$$

ali je

$$\mathbb{P}(X_1 = 2) = \mathbb{P}(Y_0 = 1, Y_1 = 1) = \frac{1}{4} > 0.$$

Napravimo malu simulaciju procesa (Y_n) i (X_n) iz prethodnog primjera, kako bi lakše vidjeli razliku u ponašanju nizova iz prethodnih primjera.



Slika 1.1: Usporedba Markovljevog lanca i n.j.d. niza - uoči kako Markovljev lanac ima vremenski trend, dok ga n.j.d. niz nema

Napomena: Promatrat ćemo isključivo *homogene* Markovljeve lance. To su oni Markovljevi lanci kod kojih vjerojatnost s desne strane u 1.1 ne ovisi o vremenskom trenutku $n \geq 0$. Takve lance možemo okarakterizirati pomoću *stohastičke matrice* i početne distribucije.

Definicija 1.1.3. *Kažemo da je matrica* $P = (p_{i,j} : i, j \in S)$ stohastička matrica (po retcima) *ako je* $p_{ij} \geq 0$ *za sve* $i, j \in S$ *te ako vrijedi:*

$$\sum_{j \in S} p_{ij} = 1, \text{ za sve } i \in S.$$

Napomena: Analogno kao i stohastičke matrice po retcima možemo definirati i stohastičke matrice po stupcima koje imaju svojstva $p_{ij} \geq 0$ za sve $i, j \in S$ i $\sum_{i \in S} p_{ij} = 1$, za sve $j \in S$. Matrice koje su i stohastičke i po retcima i po stupcima zovemo dvostruko stohastičke matrice.

Skupovi stohastičkih matrica (bilo po retcima, stupcima ili dvostruko) imaju određena zanimljiva svojstva i javljaju se u primjeni te ih se intenzivno proučava (pogledati npr. [15] ili [4]). Neka jednostavna svojstva ćemo i pokazati, kako bi približili pojam stohastičke matrice.

Primjerice, stohastičke matrice (od sada nadalje podrazumijevamo da se radi o stohastičkim matricama po retcima) čine konveksan skup. Naime, neka su A i B stohastičke matrice istog tipa. Nadalje, neka su $\alpha, \beta \in [0, 1]$ takvi da je $\alpha + \beta = 1$. Tada imamo:

$$\sum_{j \in S} (\alpha A + \beta B)_{ij} = \sum_{j \in S} (\alpha A_{ij}) + \sum_{j \in S} (\beta B_{ij}) = \alpha + \beta = 1, \text{ za sve } i, j \in S.$$

Osim toga, još jedno svojstvo je da je produkt stohastičkih matrica opet stohastička matrica. Ovo svojstvo će nam biti vrlo bitno u analizi Markovljevih lanaca, kao što ćemo vidjeti niže. Stoga pokažimo i ovo svojstvo.

Neka su opet A i B stohastičke matrice istog tipa. Tada imamo:

$$\sum_{j \in S} (AB)_{ij} = \sum_{j \in S} \sum_{k \in S} A_{ik} B_{kj} = \sum_{k \in S} \sum_{j \in S} A_{ik} B_{kj} = \sum_{k \in S} A_{ik} \sum_{j \in S} B_{kj} = \sum_{k \in S} A_{ik} = 1.$$

Napomena: Zamjena jednakosti (koja je za konačne skupove očita) u prethodnom dokazu je dozvoljna po Beppo-Levijevom teoremu, koji u posebnom slučaju daje jednakost dvaju suma nenegativnih brojeva, a puni iskaz i dokaz može se naći npr. u [19].

Definicija 1.1.4. *Neka je* $\lambda = (\lambda_i : i \in S)$ *vjerojatnosna distribucija na* S*, te neka je* $P = (p_{ij} : i, j \in S)$ *stohastička matrica. Kažemo da je slučajan proces* $X = (X_n : n \geq 0)$ *definiran na vjerojatnosnom prostoru* $(\Omega, \mathcal{F}, \mathbb{P})$ *s prostrorom stanja* S *homogen Markovljev lanac s početnom distribucijom* λ *i prijelaznom matricom* P *ako vrijedi:*

i.) $\mathbb{P}(X_0 = i) = \lambda_i$,

ii.) $\mathbb{P}(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, ..., X_0 = i_0) = p_{ij}$,

za svaki $n \geq 0$ *i sve* $i_0, ..., i_{n-1}, i, j \in S$.

Napomena: Ovakve lance ćemo kraće zvati samo (λ, P) Markovljevi lanci.

Sljedeći teorem daje vezu Markovljevih lanaca i stohastičkih matrica.

Teorem 1.1.1. *Neka je* X (λ, P) *Markovljev lanac. Tada za sve* $n \geq 0$ *i sva stanja* $i_0, i_1, ..., i_{n-1}, i_n \in S$ *vrijedi:*

$$\mathbb{P}(X_0 = i_0, X_1 = i_1, ..., X_{n-1} = i_{n-1}, X_n = i_n) = \lambda_{i_0} p_{i_0 i_1} ... p_{i_{n-1} i_n} \tag{1.2}$$

Obratno, ako pretpostavimo da je $X = (X_n : n \geq 0)$ *slučajan proces na prostoru stanja* S *s konačnodimenzionalnim distribucijama danim formulom 1.2, gdje je* λ *neka vjerojatnosna distribucija na* S*, a* P *stohastička matrica. Tada je* X (λ, P) *Markovljev lanac.*

Dokaz. Koristit ćemo formulu za uvjetnu vjerojatnost. Iz osnovne definicije uvjetne vjerojatnosti $\mathbb{P}(A \mid B) = \frac{P(A \cap B)}{\mathbb{P}(B)}$ za niz događaja $A_0, A_1, ..., A_{n-1}, A_n$ imamo:

$$\mathbb{P}(A_0 \cap A_1 \cap ... \cap A_n) = \mathbb{P}(A_0)\mathbb{P}(A_1 \mid A_0)...\mathbb{P}(A_n | A_0 \cap A_1 \cap ... \cap A_{n-1})$$

Sada korištenjem $A_k = \{X_k = i_k\}, k = 0, 1, ..., n$ imamo:

$$\mathbb{P}(X_0 = i_0, X_1 = i_1, ..., X_{n-1} = i_{n-1}, X_n = i_n) =$$
$$\mathbb{P}(X_0 = i_0)\mathbb{P}(X_1 = i_1 \mid X_0 = i_0)...\mathbb{P}(X_n = i_n \mid X_0 = i_0, X_1 = i_1, ..., X_{n-1} = i_{n-1}) =$$
$$\lambda_{i_0} p_{i_0 i_1} ... p_{i_{n-1} i_n},$$

gdje zadnji redak dobijemo iz definicije (λ, P) Markovljevog lanca 1.1.4. Time je dokazana nužnost u teoremu.

Za obrat, trebamo pokazati defincijska svojsta *i.*) i *ii.*) iz definicije 1.1.4. No, jasno je da *i.*) dobivamo uzimanjem $n = 0$ u jednakosti iz teorema. Stoga pokažimo još tvrdnju *ii.*). Pretpostavimo da je $\mathbb{P}(X_0 = i_0, ..., X_n = i) > 0$ (da bi uvjetna vjerojatnost uopće bila definirana). Tada je:

$$\mathbb{P}(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, ..., X_0 = i_0) =$$
$$\frac{\mathbb{P}(X_{n+1} = j, X_n = i, X_{n-1} = i_{n-1}, ..., X_0 = i_0)}{\mathbb{P}(X_n = i, X_{n-1} = i_{n-1}, ..., X_0 = i_0)} =$$
$$\frac{\lambda_{i_0} p_{i_0 i_1} ... p_{i_{n-1} i} p_{ij}}{\lambda_{i_0} p_{i_0 i_1} ... p_{i_{n-1} i}} = p_{ij},$$

gdje smo drugu jednakost dobili primjenom jednadžbe iz teorema na brojnik i nazivnik. □

Ovim teoremom smo dali poveznicu Markovljevih lanaca i stohastičkih matrica. Ta veza nam je od iznimne važnosti - nju koristimo kao nit vodilju za konstrukciju difuzijskih preslikavanja, ali i za interpretaciju rezultata i hiperparametara koji će se javljati.

Još jedna stvar koja će nas zanimati je vjerojatnost prijelaza između stanja nakon n koraka. Ideja je da, ako se nalazimo u stanju i i želimo odrediti vjerojatnost prijelaza u j nakon, recimo, 2 koraka, prvo odredimo vjerojatnosti prijelaza u bilo koje stanje nakon 1 koraka, a onda vjerojatnost prijelaza iz tog stanja u j nakon još jednog koraka. Općeniti rezultat dan je u propoziciji:

Propozicija 1.1.1. *Neka je X (λ, P) Markovljev lanac. Tada vrijedi:*

$$\mathbb{P}(X_n = j \mid X_0 = i) = \mathbb{P}_i(X_n = j) = p_{ij}^{(n)} \quad za\ sve\ i, j \in S, \tag{1.3}$$

gdje je $p_{ij}^{(n)}$ element na mjestu (i, j) n-te potencije matrice P.

Dokaz. Dokaz provodimo indukcijom po n. Za $n = 1$ tvrdnja je jasna (slijedi iz prethodnog teorema). Pretpostavimo da jednadžba 1.3 vrijedi za neki $n \in \mathbb{N}$. Tada imamo:

$$\begin{aligned}
&\mathbb{P}(X_{n+1} = j | X_0 = i) = \mathbb{P}(X_{n+1} = j, X_n \in S | X_0 = i) = \\
&\sum_{k \in S} \mathbb{P}(X_{n+1} = j \mid X_n = k, X_0 = i) \mathbb{P}(X_n = k \mid X_0 = i) = \\
&\sum_{k \in S} \mathbb{P}(X_{n+1} = j \mid X_n = k) \mathbb{P}(X_n = k \mid X_0 = i) = \sum_{k \in S} p_{kj} p_{ik}^{(n)} = p_{ij}^{(n+1)}.
\end{aligned}$$

□

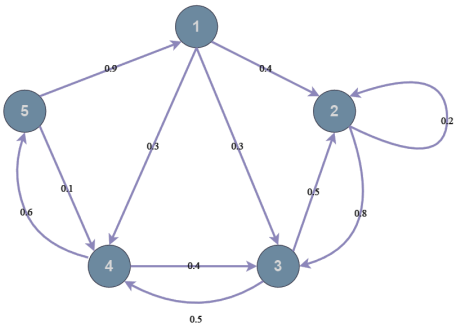
Napomena: U drugoj jednakosti u dokazu smo koristili činjenicu da su uvjetno na vrijednost X_n, X_{n+1} i X_0 nezavisni. Ta tvrdnja je sasvim intuitivna ako o Markovljevom lancu razmišljamo kao o nizu događaja za koji su uvjetno na sadašnjost, prošlost i budućnost nezavisni. Formalan dokaz se može naći u [16].

Dakle, iz Propozicije vidimo da su n-koračne prijelazne vjerojatnosti zapravo dane elementima n-te potencije prijelazne matrice P.

1.2 Slučajne šetnje na grafovima

Jedan od osnovnih primjera Markovljevih lanaca su slučajne šetnje na (ne)usmjerenim grafovima. Ako imamo slučajan proces zadan kao šetnja na grafu, u kojoj je vjerojatnost skoka u susjedni vrh jednaka (ili proporcionalna) težini pripadnog brida, takav proces možemo modelirati kao Markovljev lanac. Štoviše, vrlo lako mu odredimo matricu prijelaza - to je upravo matrica susjedstva danog grafa (uz eventualno skaliranje, ako se težine ne sumiraju u 1).

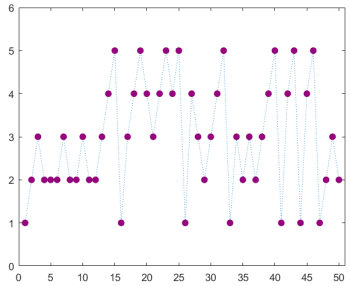
Promotrimo jedan primjer modeliranja slučajne šetnje na grafu kao Markovljev lanac. Uočimo da je za svaki vrh suma težina svih bridova koji izlaze iz promatranog vrha 1 - dakle, matrica susjedstva će biti stohastička i time direktno čitamo matricu prijelaza pripadnog Markovljevog lanca.



Slika 1.2: Primjer težinskog grafa.

$$P = \begin{bmatrix} 0 & 0.4 & 0.3 & 0.3 & 0 \\ 0 & 0.2 & 0.8 & 0 & 0 \\ 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0.4 & 0 & 0.6 \\ 0.9 & 0 & 0 & 0.1 & 0 \end{bmatrix}$$

Slika 1.3: Pripadna matrica prijelaza.

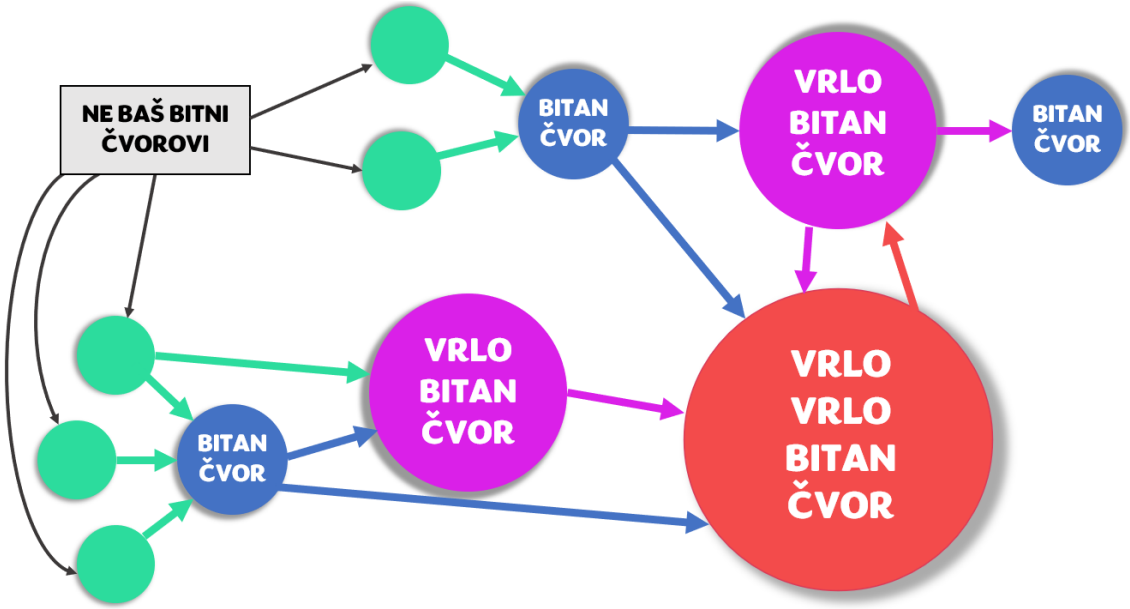


Slika 1.4: Simulacija slučajne šetnje na težinskom grafu.

Naravno, osim ovog umjetnog primjera, slučajne šetnje na grafovima se u iznimno velikoj mjeri javljaju u raznim matematičkim disciplinama, ali i u primjeni. Jedan od najpoznatijih primjera je tzv. *Google lanac*.

Primjer 1.2.1. Google lanac je primjer slučajne šetnje na grafu čiji su vrhovi web stranice, a težine između vrhova odgovaraju vjerojatnostima skoka s jedne stranice na drugu (tu pretpostavljamo da svaki link na nekoj stranici kliknemo s jednakom vjerojatnosti). Ovaj lanac ima za cilj naći optimalno (u nekom smislu - detaljnije u [5]) rangiranje web stranica u kratkom roku, kako bi korisnik interneta brzo mogao doći na ”dobre” stranice za ono što je tražio, odnosno da visoko rangirane stranice budu one na kojima su najbitniji podaci.

Imamo 2 zahtjeva na važnost vrhova u rangiranju - vrh je važan ako na njega pokazuju važni vrhovi, a važnost linkova onih stranica koje pokazuju na puno drugih je proporcionalno manja (možemo ova 2 pravila interpretirati kao "važni vrhovi tvrde da sam važan, pa onda moram biti važan" i "glas onoga tko za sve tvrdi da je važan je manje bitan"). Primjer možemo vidjeti na ilustraciji.



Slika 1.5: Ilustracija Google lanca - veći čvorovi su bitniji, uočavamo da glas većeg čvora igra veću ulogu nego glas malog.

Ako je skup svih vrhova (web stranica) S, te ako za jedan vrh $i \in S$ definiramo skup $L_i \subset S$ kao skup svih stranica koje imaju link na i, onda zahtjeve koje smo postavili na pridjeljivanje važnosti vrhovima možemo zapisati kao:

$$x_i = \sum_{j \in S} \frac{1}{n_j} x_j.$$

Ako zapišemo važnosti u vektor x, a težine linkova u matricu A, kao $A_{ij} = \frac{1}{n_j}$ ako $j \to i$, gdje je n_j broj linkova koji pokazuju iz j, s tim da stavljamo $A_{ij} = 0$ ako $j \not\to i$, možemo sve ovo ostvariti traženjem vektora x koji zadovoljava

$$x = Ax.$$

U kontekstu Markovljevih lanaca, ovo se svodi na problem traženja stacionarne distribucije lanca, a u teoriji matrica, ovo je traženje svojstvenog vektora pripadne svojstvene vrijednosti 1 - obje teme ćemo obraditi u ovom i idućem poglavlju.

Vratimo se na teoriju slučajnih šetnji - osim što dijele zajedničku matricu, sam graf i Markovljev lanac će dijeliti i druga svojstva. Jedno od svojstava grafova koje često analiziramo je (jaka) povezanost grafa. U tu svrhu, uvodimo povezan pojam Markovljevih lanaca - stanja komuniciranja i ireducibilnost.

Definicija 1.2.1. *Neka je* $X = (X_n \mid n \geq 0)$ *Markovljev lanac s prostorom stanja* S *i prijelaznom matricom* P. *Za podskup* $B \subseteq S$ *definiramo* prvo vrijeme pogađanja *tog skupa kao slučajnu varijablu:*

$$T_B = \inf\{n \geq 0 : X_n \in B\},$$

uz dogovor $\inf \emptyset = +\infty$. *Nadalje, ako je* $B = j$ *jednočlan skup, kraće pišemo* T_j, *umjesto* $T_{\{j\}}$.

Definicija 1.2.2. *Za stanja* $i, j \in S$ *kažemo da je* j dostižno *iz* i, *u oznaci* $i \longrightarrow j$, *ako vrijedi*

$$\mathbb{P}_i(T_j < +\infty) = \mathbb{P}(T_j < +\infty \mid X_0 = i) > 0.$$

Intuitivno, stanje j je dostižno iz i ako se, krenuvši iz stanja i u stanje j može doći u konačno mnogo koraka s pozitivnom vjerojatnošću.

Definicija 1.2.3. *Kažemo da stanja* $i, j \in S$ komuniciraju *ako je* i *dostižno iz* j *i ako je* j *dostižno iz* i, *uz oznaku* $i \longleftrightarrow j$. *Nadalje, ako za svaka dva stanja* $i, j \in S$ *vrijedi* $i \longleftrightarrow j$, *kažemo da je lanac* X ireducibilan.

Svojstvo ireducibilnosti je od iznimne važnosti u analizi Markovljevih lanaca, ne samo u vezi s promatranjem slučajnih šetnji na grafovima, kao što ćemo vidjeti u daljnjim teoremima. Zasad, bitno je uočiti da je za slučajnu šetnju na grafu, pripadni lanac ireducibilan, ako i samo ako je graf jako povezan (između svaka dva vrha postoji put koji ih povezuje), što se lako vidi iz sljedećeg: ako je graf jako povezan, tada između svaka dva vrha postoji put (konačan niz koraka) pozitivne vjerojatnosti, pa je posebno vrijeme pogađanja konačno s pozitivnom vjerojatnosti. Obratno, ako je lanac ireducibilan, onda za bilo koja 2 stanja postoji pozitivna vjerojatnost dostizanja jednog stanja iz drugog u konačno mnogo koraka, pa iz aditivnosti vjerojatnosti slijedi da mora postojati konačan put pozitivne težine od jednog stanja do drugog.

1.3 Stacionarna distribucija Markovljevog lanca

Stacionarnost slučajnog procesa znači da se njegova vjerojatnosna svojstva ne mijenjaju s vremenom. Preciznije, ako je $X = (X_n : n \geq 0)$ stacionaran slučajan proces, onda je distribucija svih njegovih komponenti X_n jednaka. Kod Markovljevih lanaca, distribucija u svakom trenutku se lako odredi iz prethodnog. Primjerice, ako je π distribucija

$(\pi, P)-$Markovljevog lanca u trenutku $t = 0$, onda je πP distribucija u trenutku $t = 1$. To motivira definiciju stacionarne distribucije.

Definicija 1.3.1. *Slučajni proces $X = (X_n : n \geq 0)$ na vjerojatnosnom prostoru $(\Omega, \mathcal{F}, \mathbb{P})$ je* stacionaran *ako $\forall k \geq 0$ i $\forall n \geq 0$ vrijedi da su slučajni vektori*

$$(X_0, X_1, ..., X_k) \text{ i } (X_n, X_{n+1}, ..., X_{n+k})$$

jednako distribuirani.

Definicija 1.3.2. *Neka je $X = (X_n : n \geq 0)$ Markovljev lanac sa skupom stanja S i matricom prijelaza P. Vjerojatnosna distribucija $\pi = (\pi_i : i \in S)$ je* stacionarna distribucija *Markovljevog lanca X ako vrijedi $\pi = \pi P$.*

Sljedeći teorem, čiji dokaz je dostupan u [16], govori da je Markovljev lanac čija je početna distribucija stacionarna, stacionaran proces.

Teorem 1.3.1. *Neka je $X = (X_n : n \geq 0)$ (π, P)-Markovljev lanac gdje je π stacionarna distribucija za P. Tada je X stacionaran proces. Preciznije, X je stacionaran uz vjerojatnost $\mathbb{P}_\pi = \sum_{i \in S} \pi_i \mathbb{P}_i$. Nadalje, za svaki $m \geq 0$ je $(X_{m+n} : n \geq 0)$ ponovno (π, P)-Markovljev lanac.*

Radi ilustracije pojma stacionarnosti, navedimo i jedan primjer, koji nam daje i fizikalnu interpretaciju procesa definiranih stohastičkim matricama, kao što su Markovljevi lanci.

Primjer 1.3.1. Pretpostavimo da imamo Markovljev lanac $(X_n)_{n \geq 0}$ sa skupom stanja $S = \{1, 2, 3\}$ i prijelaznom matricom:

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

Zamislimo sljedeće: pretpostavimo da se u početnom trenutku u stanju 1 nalazi čestica mase 1. Njeno dijeljenje opisujemo matricom P - nakon prvog trenutka, pola mase prelazi u stanje 2, a pola u stanje 3. Sada se polovina mase koja se nalazi u stanju 2 podijeli i četvrtina ukupne mase prelazi u stanje 1, a četvrtina u 3. Ista stvar se događa s masom u stanju 3 - četvrtina ukupne ide u 1, četvrtina u 2. Proces možemo nastaviti u beskonačno mnogo trenutaka. Pitamo se postoji li neka jasna pravilnost u kretanju te mase i hoće li se ona s vremenom stabilizirati?

Ako u trenutku t raspodjelu mase možemo dobiti kao $P^t\pi$, gdje je π početna raspodjela mase, zanima nas postoji li limes tog niza vektora. Matricu je lakše potencirati nakon dijagonalizacije, a dijagonalizacija od P se postiže s:

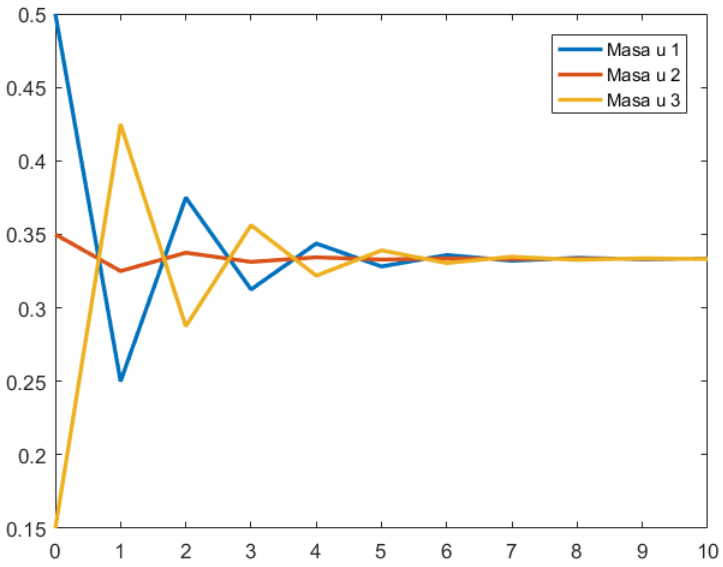
$$P = V\Lambda V^T,\ \Lambda = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\frac{1}{2} & 0 \\ 0 & 0 & -\frac{1}{2} \end{bmatrix},\ V = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{3}} & 0 & -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

Sada se potenciranje matrice P svodi na potenciranje matrice Λ, čije su potencije dijagonalne matrice koje na dijagonali imaju 1 i $\frac{-1}{2^t}$ pa sve skupa matrica P^t teži u

$$\hat{P} = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Stoga je i distribucija u t-tom trenutku za veliki t približno jednaka $\begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$ (neovisno o početnoj distribuciji π. Možemo grafički prikazati simulaciju procesa s početnom distribucijom $\pi = \begin{bmatrix} 0.5 & 0.35 & 0.15 \end{bmatrix}$.

Dakle, na ovom primjeru vidimo da se s vremenom distribucija u trenutku t stabilizira - ovo je primjer *granične distribucije* Markovljevog lanca, koju ćemo definirati i diskutirati u idućoj sekciji.



Slika 1.6: Prikaz stabilizacije mase u 10 koraka, uočavamo da već nakon par koraka nema puno promjene.

Što bi se dogodilo da smo odmah krenuli s distribucijom $\hat{\pi} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$? Primijetili bi da je u svakom idućem koraku distribucija jednaka, u što se lako uvjerimo množenjem matrice P s $\hat{\pi}$. To je upravo zato jer je distribucija $\hat{\pi}$ stacionarna za Markovljev lanac s matricom prijelaza P.

Još valja uočiti i distribucija u koju se stabilizira distribucija mase polazeći od bilo koje početne distribucije upravo $\hat{\pi}$. To nije slučajnost, a teorem čiji će rezultat govoriti o upravo toj pojavi ćemo također iznijeti u idućoj sekciji.

Dotaknimo se još postojanja stacionarne distribucije za Markovljev lanac. U tu svrhu, prvo se prisjetimo pojma prvog vremena pogađanja skupa (odnosno, u ovom slučaju, vremena prvog povratka u točku):

$$T_i = \min\{n > 0 : X_n = i\}.$$

Ova varijabla nam je bitna za definiciju svojstva pozitivne povratnosti za neko stanje $i \in S$, za koje se ispostavlja da je ekvivalentno postojanju stacionarne distribucije Markovljevog lanca.

Definicija 1.3.3. *Kažemo da je stanje $i \in S$* pozitivno povratno *ako je $\mathbb{E}_i[T_i] < \infty$.*

Sljedeći teorem, čiji dokaz se može naći u [16] kao teorem 7.14, je ključan u analizi Markovljevih lanaca:

Teorem 1.3.2. *Neka je X ireducibilan Markovljev lanac s prijelaznom matricom P. Tada je ekvivalentno:*

1. *Svako stanje $i \in S$ je pozitivno povratno.*
2. *Postoji pozitivno povratno stanje $j \in S$.*
3. *X ima stacionarnu distribuciju π.*

1.4 Granična distribucija Markovljevog lanca

U ovom potpoglavlju ćemo definirati graničnu distribuciju lanca i povezati ju sa stacionarnom. Ta veza će nam davati informaciju o ponašanju potencija matrice prijelaza Markovljevog lanca kada vrijeme t teži u beskonačnost.

Definicija 1.4.1. *Neka je $X = (X_n : n \geq 0)$ Markovljev lanac na skupu stanja S s prijelaznom matricom P. Vjerojatnosna distribucija $\pi = (\pi_i : i \in S)$ se naziva* graničnom distribucijom *Markovljevog lanca X (odnosno matrice P) ako za sve $i, j \in S$ vrijedi:*

$$\lim_{n\to\infty} p_{ij}^{(n)} = \pi_j.$$

Teorem 1.4.1. *Neka je π granična distribucija Markovljevog lanca X. Tada je π i stacionarna distribucija.*

Dokaz. Bez smanjenja općenitosti, možemo uzeti $S = \{0, 1, 2, ...\}$. Za svako stanje $j \in S$ i svaki $M \in \mathbb{N}$ vrijedi:

$$\pi_j = \lim_{n\to\infty} \sum_{k=0}^{\infty} p_{ik}^{(n)} p_{kj} \geq \lim_{n\to\infty} \sum_{k=0}^{M} p_{ik}^{(n)} p_{kj} = \sum_{k=0}^{M} \lim_{n\to\infty} p_{ik}^{(n)} p_{kj} = \sum_{k=0}^{M} \pi_k p_{kj}.$$

Pustimo $M \to \infty$ čime dobijemo:

$$\pi_j \geq \sum_{k=0}^{\infty} \pi_k p_{kj}, \quad \text{za sve } j \in S. \tag{1.4}$$

Želimo dokazati jednakost u prethodnoj nejednakosti za sve $j \in S$. U tu svrhu, pretpostavimo da postoji neki $j_0 \in S$ za koji je nejednakost stroga. Tada je:

$$\pi_{j_0} > \sum_{k=0}^{\infty} \pi_k p_{kj_0}. \tag{1.5}$$

Zbrojimo nejednakosti 1.4 po $j \in S$. Uzevši u obzir nejednakost u 1.5 dobivamo

$$\sum_{j\in S} \pi_j > \sum_{j\in S} \sum_{k\in S} \pi_k p_{kj} = \sum_{k\in S} \pi_k \left(\sum_{j\in S} p_k j \right) = \sum_{k\in S} \pi_k = 1,$$

gdje prva jednakost slijedi po Lebesgueovom teoremu monotone konvergencije. Naravno, dobili smo kontradikciju. Dakle, u 1.4 imamo jednakost za sve $j \in S$ što znači da je π stacionarna distribucija. □

Dakle, ako imamo graničnu, imamo i stacionarnu distribuciju. Prirodno pitanje koje se javlja je: kada granična distribucija postoji? Tim pitanjem se bavimo u ostatku poglavlja.

Definicija 1.4.2. *Neka je X Markovljev lanac s prijelaznom matricom P. Za stanje $i \in S$, označimo s $d(i)$ najveći zajednički djelitelj skupa $\{n \geq 1 : p_{ii}^{(n)} > 0\}$, uz $d(i) = 1$ ako je taj skup prazan. Kažemo da je stanje i* aperiodičko*, ako je $d(i) = 1$. U suprotnom je i periodičko stanje, a $d(i)$ je* period *od i.*

Aperiodičnost je svojstvo koje će nam dati obrat teorema 1.4.1. Tada ćemo postojanje granične distribucije svesti na postojanje stacionarne, za što imamo dobro razvijenu teoriju.

Teorem 1.4.2. *Neka je λ proizvoljna vjerojatnosna distribucija na skupu stanja S. Pretpostavimo da je $X = (X_n : n \geq 0)$ (λ, P)-Markovljev lanac koji je ireducibilan i aperiodičan, te ima stacionarnu distribuciju π. Tada je*

$$\lim_{n\to\infty} \mathbb{P}(X_n = j) = \pi_j, \text{ za sve } j \in S. \tag{1.6}$$

Specijalno je i

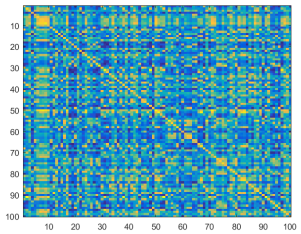
$$\lim_{n\to\infty} p_{ij}^{(n)} = \pi_j, \text{ za sve } i, j \in S, \tag{1.7}$$

to jest, stacionarna distribucija je ujedno i granična.

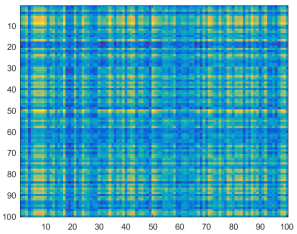
Dokaz teorema se zasniva na tehnici sparivanja Markovljevih lanaca, te je relativno tehnički zahtjevan, pa ga preskačemo, a može se naći u poglavlju 8 u [16].

Za kraj poglavlja, promotrimo još jedan primjer ponašanja stohastičke matrice kroz vrijeme kako bismo bolje shvatili pojam granične distribucije.

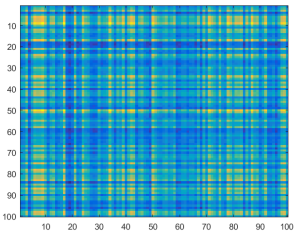
Primjer 1.4.1. Krećemo s nasumično generiranom stohastičkom matricom reda 100. Želimo uočiti pravilnost u njenom ponašanju kroz vrijeme.



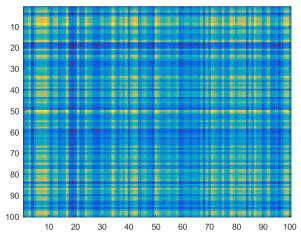
(a) $t = 1$



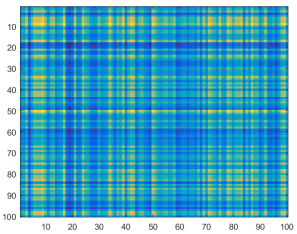
(b) $t = 2$



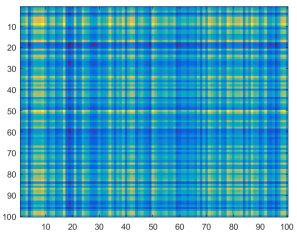
(c) $t = 3$



(d) $t = 4$



(e) $t = 5$



(f) $t = 6$

Uočavamo da već nakon 3-4 koraka imamo matricu koja je približno ranga 1, a onda nakon 6 i više pogotovo - svi stupci su kolinearni, jedina razlika je u njihovom skaliranju, ali da je u pitanju rang 1 se jasno vidi - upravo to je primjer granične distribucije, nakon nekog

vremena će se matrica stabilizirati te neovisno o tome gdje smo krenuli, te distribucija nakon dovoljnog broja vremenskih koraka postaje stabilna.

Poglavlje 2

Numerička linearna algebra i problem svojstvenih vrijednosti

Pri računanju difuzijskih preslikavanja, jedan od glavnih zadataka će biti rješavanje svojstvenog problema matrice P, tj. traženje vektora x i vrijednosti λ koji zadovoljavaju relaciju $Px = \lambda x$.

Ovo je vrlo zahtjevan problem, te je vrlo često nemoguće (u razumnom vremenu) odrediti sve takve vrijednosti i vektore egzaktno, stoga koristimo razne numeričke metode, koje na uštrb točnosti daju aproksimacije svojstvenih vrijednosti i vektora u zadovoljavajućem vremenu. U ovom poglavlju ćemo opisati rješavanje svojstvenog problema, u kojim situacijama njegovo rješenje postoji i kako struktura matrice može olakšati njegovo rješavanje.

Bavit ćemo se isključivo realnim matricama, jer račun koji ćemo provoditi u algoritmu računanja difuzijskih preslikavanja ne zahtijeva kompleksne vrijednosti. Ipak, čak i u slučaju realnih matrica, ne možemo uvijek izbjeći kompleksne svojstvene vrijednosti i vektore. Posebnu pažnju posvetit ćemo simetričnim matricama, jer su nam upravo one bitne u difuzijskim preslikavanjima, a i pokazuje se da imaju vrlo lijepa svojstva spektra.

2.1 Teorija matrica i dijagonalizacija

Definicija 2.1.1. *Kažemo da je $\lambda \in \mathbb{C}$* svojstvena vrijednost *kvadratne matrice $A \in \mathbb{R}^{n\times n}$ ako postoji vektor $x \in \mathbb{R}^n$ različit od nulvektora takav da vrijedi $Ax = \lambda x$. Skup svih svojstvenih vrijednosti nazivamo* spektar *matrice A i označavamo*

$$\sigma_A = \{\lambda \in \mathbb{C} : \exists x \neq 0 \text{ takav da } Ax = \lambda x\}.$$

Jedan od glavnih alata u numeričkom računanju svojstvenih vrijednosti je Schurova dekompozicija:

Teorem 2.1.1. *Neka je $A \in \mathbb{C}^{n\times n}$ i neka su $\lambda_1, \lambda_2, ..., \lambda_n$ svojstvene vrijednosti od A u proizvoljnom poretku. Postoji unitarna matrica U i gornjetrokutasta matrica T tako da je $A = UTU^*$ i $T_{ii} = \lambda_i$, $i = 1, 2, ..., n$. Ako je $A \in \mathbb{R}^{n\times n}$ i ako su sve svojstvene vrijednosti od A realne, onda je T također realna i U se može odabrati realna ortogonalna. Zapis $A = UTU^*$ zovemo* Schurova dekompozicija *od A, a matrica T zove se* Schurova forma *od A.*

Dokaz. Uzmimo svojstvenu vrijednost λ_1 i pripadni svojstveni vektor u_1 tako da je $Au_1 = \lambda_1 u_1$ te neka je u_1 normiran: $\|u_1\|_2 = 1$. Možemo odrediti matricu $\hat{U}_1 \in C^{n\times(n-1)}$ takvu da je $U_1 = (u_1 \hat{U}_1)$ unitarna (dopuna do baze unitarnog prostora). Sada vrijedi:

$$U_1^* A U_1 = \begin{pmatrix} u_1^* \\ \hat{U}_1^* \end{pmatrix} \begin{pmatrix} Au_1 & A\hat{U}_1 \end{pmatrix} = \begin{pmatrix} \lambda_1 & u_1^* A \hat{U}_1 \\ \mathbf{0} & A_2 \end{pmatrix}, \; A_2 = \hat{U}_1^* A \hat{U}_1.$$

Iz blok dijagonalnosti desne strane imamo $\det(A - \lambda I_n) = (\lambda_1 - \lambda)\det(A_2 - \lambda I_{n-1})$ pa slijedi da su $\lambda_2, ..., \lambda_n$ svojstvene vrijednosti matrice A_2. Ako je $n > 2$ (u protivnom smo gotovi) uzmimo λ_2 i pripadni svojstveni vektor u_2 takav da $A_2 u_2 = \lambda_2 u_2$, te opet pretpostavimo normiranost $\|u_2\|_2 = 1$. Kao i za λ_1, radimo dopunu u_2 do unitarne matrice $U_2 = \begin{pmatrix} u_2 & \hat{U}_2 \end{pmatrix} \in \mathbb{C}^{(n-1)\times(n-1)}$, te izračunamo:

$$U_2^* A_2 U_2 = \begin{pmatrix} u_2^* \\ \hat{U}_2^* \end{pmatrix} \begin{pmatrix} A_2 u_2 & A_2 \hat{U}_2 \end{pmatrix} = \begin{pmatrix} \lambda_2 & u_2^* A_2 \hat{U}_2^* \\ \mathbf{0} & A_3 \end{pmatrix}, \; A_3 = \hat{U}_2^* A_2 \hat{U}_2.$$

Ako ova dva koraka napravimo istovremeno, dobijemo:

$$\begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \hat{U}_2{}^* \end{pmatrix} U_1^* A U_1 \begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \hat{U}_2 \end{pmatrix} = \begin{pmatrix} \lambda_1 & * & * \\ 0 & \lambda_2 & * \\ \mathbf{0} & \mathbf{0} & A_3 \end{pmatrix}.$$

Sada kao i u prethodnom koraku zaključujemo da su svojstvene vrijednosti od A_3 upravo $\lambda_3, ..., \lambda_n$ i nastavljamo na isti način. Zaključujemo da induktivnim postupkom po n dobivamo potpunu dekompoziciju matrice $A = U^* T U$.

Još treba opravdati činjenicu da ako je A realna, onda se ova dekompozicija može odabrati tako da je U realna ortogonalna. Ako je matrica A realna s realnim svojstvenim vrijednostima, tada se u svakom koraku može odabrati realan svojstveni vektor, čime dobivamo da je U ortogonalna (umjesto unitarne). □

S obzirom na to da ćemo u ovom poglavlju raditi ocjene grešaka algoritama i mjeriti udaljenost od prave vrijednosti, prisjetimo se i pojma matrične norme:

Definicija 2.1.2. Matrična norma *je svaka funkcija na prostoru matrica* $\nu : \mathbb{R}^{m\times n} \mapsto \mathbb{R}$ *koja zadovoljava sljedeća svojstva:*

i.) $\nu(A) \geq 0$, *za sve* $A \in \mathbb{R}^{m\times n}$,

ii.) $\nu(A) = 0 \iff A = 0$,

iii.) $\nu(\alpha A) = |\alpha|\nu(A)$, *za sve* $\alpha \in \mathbb{R}$, $A \in \mathbb{R}^{m\times n}$,

iv.) $\nu(A + B) \leq \nu(A) + \nu(B)$, *za sve* $A, B \in \mathbb{R}^{m\times n}$.

Napomena: Normu naravno definiramo i općenitije, na prozivoljnom vektorskom prostoru. Prethodna definicija naravno definira i vektorsku normu - samo prostor $\mathbb{R}^{m\times n}$ zamijenimo proizvoljnim vektorskim prostorom. Ipak, od posebne važnosti u ovom radu su matrične norme, koje će imati i dodatnu strukturu.
Uobičajena oznaka za preslikavanja koja zadovoljavaju svojstva norme je $\|\cdot\|$. Uz gore naveden svojstva se za matrične norme često dodaje i sljedeće:

Definicija 2.1.3. *Kažemo da je matrična norma* $\|\cdot\|$ konzistentna *ako vrijedi:*

$$\|AB\| \leq \|A\|\|B\|,$$

uz pretpostavku da su matrice A i B ulančane.

Napomena: Kako ćemo većinom raditi s kvadratnim matricama, čiji je produkt uvijek dobro definiran, svojstvo konzistentnosti ćemo implicitno pretpostavljati kao jedno od definicijskih svojstava matrične norme.
Još ćemo definirati *induciranu* normu - ona povezuje normu na prostoru vektora s matričnom normom na prostoru matrica koje djeluju na spomenute vektore.

Definicija 2.1.4. *Kažemo da je matrična norma* $\|\cdot\|_M$ inducirana *vektorskom normom* $\|\cdot\|$ *ako vrijedi:*

$$\|A\|_M = \max_{x\neq 0} \frac{\|Ax\|}{\|x\|}.$$

Vratimo se sada problemu svojstvenih vrijednosti. Jedan od vrlo bitnih pojmova u teoriji matrica je pojam *spektralnog radijusa* matrice:

Definicija 2.1.5. *Spektralni radijus matrice, u oznaci* $\mathrm{spr}(A)$*, matrice* $A \in \mathbb{R}^{n\times n}$ *je definiran kao maksimalna apsolutna svojstvena vrijednost matrice, tj.* $\mathrm{spr}(A) = \max_{\lambda\in\sigma_A} |\lambda|$.

Lako se vidi da spektralni radijus kao funkcija nije norma. Primjerice, ako uzmemo matricu $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, njen spektar je $\sigma_A = \{0\}$, ali ona očito nije nulmatrica.

S druge strane, spektralni radijus na neki način mjeri veličinu matrice - primjerice, ako matricu skaliramo, skalirat ćemo i njen spektralni radijus. To jest, ako je $\text{spr}(A) = \rho$, onda je $\text{spr}(\lambda A) = |\lambda|\rho$.

Ipak, iako nije norma, u vrlo je bliskoj vezi s matričnim normama, u smislu sljedećeg teorema:

Teorem 2.1.2. *Za proizvoljnu matričnu normu $\|\cdot\|$ na $\mathbb{R}^{n\times n}$ i svaku matricu $A \in \mathbb{R}^{n\times n}$ vrijedi* $\text{spr}(A) \leq \|A\|$. *Nadalje, za svaku matricu $A \in \mathbb{R}^{n\times n}$ i svaki $\epsilon > 0$ postoji inducirana matrična norma $\|\cdot\|_\epsilon$ takva da vrijedi $\|A\|_\epsilon < \text{spr}(A) + \epsilon$.*

Dokaz. Za dokaz prve tvrdnje, neka je $A \in \mathbb{R}^{n\times n}$ proizvoljna matrica, a

$$\|\cdot\|$$

proizvoljna matrična norma. Nadalje, neka je λ proizvoljna svojstvena vrijednost matrice A. Definirajmo matricu $V \in \mathbb{R}^{n\times n}$ kao matricu čije stupce dobijemo repliciranjem pripadnog svojstvenog vektora svojstvene vrijednosti λ n puta. Tada imamo:

$$|\lambda|\|V\| = \|AV\| \leq \|A\|\|V\|,$$

odakle dijeljenjem s $\|V\|$ dobijemo $|\lambda| \leq \|A\|$.

Za dokaz druge tvrdnje, neka je $\epsilon > 0$. Nadalje, neka je $A = UTU^*$ Schurova forma matrice A. Definiramo $D_{\tilde{\epsilon}} = \text{diag}(1, \tilde{\epsilon}, ..., \tilde{\epsilon}^{n-1})$, pri čemu ćemo $\tilde{\epsilon} > 0$ odabrati naknadno. Matrica $T_{\tilde{\epsilon}} = D_{\tilde{\epsilon}}^{-1} T D_{\tilde{\epsilon}}$ se od matrice T razlikuje samo na pozicijama (i, j) u strogo gornjem trokutu, uz $(T_{\tilde{\epsilon}})_{ij} = (T)_{ij}\tilde{\epsilon}^{j-i}$. Definirajmo vektorsku normu $\|x\| = \|(UD_{\tilde{\epsilon}})^{-1}x\|_\infty$. Pripadna operatorska norma zadovoljava:

$$\begin{aligned}
\|A\|_\epsilon &= \max_{x\neq 0} \frac{\|Ax\|_\epsilon}{\|x\|_\epsilon} = \max_{x\neq 0} \frac{\|(UD_{\tilde{\epsilon}})^{-1}Ax\|_\infty}{\|UD_{\tilde{\epsilon}})^{-1}x\|_\infty} \\
&= \max_{y\neq 0} \frac{\|(UD_{\tilde{\epsilon}})^{-1})AUD_{\tilde{\epsilon}}y\|_\infty}{\|y\|_\infty} = \max_{y\neq 0} \frac{\|T_{\tilde{\epsilon}}y\|_\infty}{\|y\|_\infty} \\
&= |||T_{\tilde{\epsilon}}|||_\infty = \max_{i=1:n} \sum_{j=i}^{n} |(T_{\tilde{\epsilon}})_{ij}| \\
&= \max_{i=1:n} \sum_{j=i}^{n} |(T)_{ij}|\tilde{\epsilon}^{j-i} \leq \text{spr}(A) + \epsilon,
\end{aligned}$$

za proizvoljne $\tilde{\epsilon} \in \langle 0, \frac{\epsilon}{|||T|||_\infty}\rangle$. □

Iz prethodnog teorema zaključujemo sljedeće: iako spektralni radijus nije norma, možemo na njega gledati kao na infimum svih matričnih normi za danu matricu. Stoga možemo očekivati da će spektralni radijus davati izuzetno važnu mjeru matrice, kao što ćemo vidjeti u sljedećem odjeljku.

2.2 Dijagonalizacija matrice

Pretežno ćemo se baviti simetričnim i realnim matricama. Za njih vrijedi sljedeća tvrdnja:

Propozicija 2.2.1. *Neka je $A \in \mathbb{R}^{n\times n}$ simetrična matrica. Tada su njene svojstvene vrijednosti realni brojevi.*

Dokaz. Neka je $\lambda \in \sigma_A$ svojstvena vrijednost matrice A, te neka je $v \in \mathbb{R}^n$ pripadni svojstveni vektor. Kako je A realna i simetrična, imamo $A = A^*$. Tada vrijedi:

$$\lambda\langle v, v\rangle = \langle \lambda v, v\rangle = \langle Av, v\rangle = \langle v, A^*v\rangle = \langle v, Av\rangle = \langle v, \lambda v\rangle = \overline{\lambda}\langle v, v\rangle. \tag{2.1}$$

Kako je v svojstveni vektor, različit je od $\mathbf{0}$, pa mu je i norma različita od 0, pa dijeljenjem krajnje lijeve i desne strane prethodnih jednakosti dobijemo $\lambda = \overline{\lambda}$, pa je λ realna. □

Iz teorema o Schurovoj dekompoziciji lako dobivamo sljedeći rezultat:

Propozicija 2.2.2. *Ako je kvadratna matrica $A \in \mathbb{R}^{n\times n}$ simetrična, tada postoje dijagonalna matrica D i ortogonalna matrica S takve da vrijedi:*

$$A = SDS^T.$$

Dokaz. Neka je $A = UTU^*$ Schurova dekompozicija matrice A. Iz drugog dijela teorema 2.1.1 imamo da je, jer je A realna s realnim svojstvenim vrijednostima (po prethodnoj propoziciji), U ortogonalna matrica. Stoga imamo:

$$A = UTU^T \;\&\; A^T = A \implies A = UTU^T = UT^TU^T = A^T \implies T = T^T.$$

Vidimo da je T iz Schurove dekompozicije od A dijagonalna, pa za D iz iskaza propozicije možemo uzeti $D = T$, a za S uzmemo $S = U$. □

Dakle, za realnu simetričnu matricu znamo da je dijagonalizabilna i da su joj svojstvene vrijednosti realne. Ipak, ne znamo ništa o tome kako ih izračunati, te je problem računanja svojstvenih vrijednosti realne simetrične matrice netrivijalan i u praksi vrlo zahtjevan. Stoga pribjegavamo numeričkim metodama za računanje svojstvenih vrijednosti i vektora.

2.3 Numeričko računanje svojstvenih vrijednosti

2.3.1 Metoda potencija

Metoda potencija je vjerojatno najjednostavnija metoda za računanje svojstvenih vrijednosti i vektora. Pretpostavimo za početak da imamo kvadratnu matricu $A \in \mathbb{R}^{n\times n}$ koja je simetrična (takvima ćemo se baviti u cijelom radu), a time i dijagonalizabilna, te za čije svojstvene vrijednosti $\lambda_1, \lambda_2, \lambda_3..., \lambda_n$ vrijedi da je λ_1 po apsolutnoj vrijednosti izdvojena od ostalih, u smislu $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq ... \geq |\lambda_n|$. Metoda polazi od nenul vektora $x \in \mathbb{R}^n$ te uzastopnim djelovanjem matrica A na x i normiranjem, cilj je dobiti svojstveni vektor (odnosno, nakon dovoljno koraka, njegovu dobru aproksimaciju). Opišimo zašto to radi.

Iz pretpostavke da je A simterična slijedi da postoji ortogonalna matrica S i dijagonalna D takve da vrijedi $A = SDS^T$. Posebno, to znači da stupci matrice S, vektori $s_1, s_2, ..., s_n$ čine ortonormiranu bazu za $\mathbb{R}^n$. Stoga proizvoljni vektor x možemo raspisati kao $x = \xi_1 s_1 + \xi_2 s_2 + ... + \xi_n s_n$. Sada imamo:

$$Ax = A(\xi_1 s_1 + \xi_2 s_2 + ...\xi_n s_n) = \xi_1\lambda_1 s_1 + \xi_2\lambda_2 s_2 + ... + \xi_n\lambda_n s_n$$
$$A^k x = ... = \xi_1\lambda_1^k s_1 + \xi_2\lambda_2^k s_2 + ... + \xi_n\lambda_n^k s_n.$$

Kako smo pretpostavili da je $|\lambda_1| > |\lambda_2| \geq ... \geq |\lambda_n|$, zaključujemo da kada $k \to \infty$ vrijedi $\frac{|\lambda_i|}{|\lambda_1|} \to 0,\ i = 1, ..., n$. Uz uzvjet $\xi_1 \neq 0$ (koji će biti zadovoljen gotovo uvijek, jer ako proizvoljni vektor x biramo iz neke neprekidne distribucije, vjerojatnost da ga uzmemo iz prostora okomitog na s_1 je 0), imamo:

$$\|\frac{A^k x}{\lambda_1^k} - \xi_1 s_1\| \leq \frac{|\lambda_2|}{|\lambda_1|}\left(|\xi_2|\|s_2\| + ... + |\xi_n|\|s_n\|\right) \to 0.$$

Dakle, vidimo da $A^k x$ postaje sve više paralelan s s_1. Tijekom računanja aproksimacije i normiramo, čime dobijemo niz $y^{(k)}$:

$$y^{(k)} = \frac{A^k x}{\|A^k x\|} = \frac{\lambda_1^k\left(\xi_1 s_1 + \xi_2(\frac{\lambda_2}{\lambda_1})s_2 + ... + \xi_n\left(\frac{\lambda_n}{\lambda_1}\right)s_n\right)}{|\lambda_1|^k\|\xi_1 s_1 + \xi_2(\frac{\lambda_2}{\lambda_1})s_2 + ... + \xi_n(\frac{\lambda_n}{\lambda_1})s_n\|}$$
$$\approx \left(\frac{\lambda_1}{|\lambda_1|}\right)^k \frac{\xi_1}{|\xi_1|}\frac{s_1}{\|s_1\|} := \eta_k\frac{s_1}{\|s_1\|}.$$

Odavde vidimo da, iako niz iteracija $y^{(k)}$ nije nužno konvergentan (negativan predznak može uzrokovati promjenu orijentacije vektora u svakom koraku), s dovoljno velikim k je uvijek blizu vektoru paralelnom s s_1.

Dakle, metoda će (ako su pretpostavke zadovoljene) vratiti vektor koji je aproksimacija svojstvenog vektora. Još trebamo naći pripadnu svojstvenu vrijednost. Za to koristimo *Rayleighev koeficijent*.

Definicija 2.3.1. *Za matricu* $A \in \mathbb{R}^{n\times n}$ *i vektor* $x \neq 0$ *definiramo* Rayleighev koeficijent *kao:*

$$\rho := \rho(A, x) = \frac{x^T Ax}{x^T x}.$$

Rayleighev koeficijent je u određenom smislu najbolja aproksimacija svojstvene vrijednosti matrice ako je x aproksimacija svojstvenog vektora. Konkretno, jedan od kriterija koje možemo htjeti zadovoljiti je da rezidual $r = Ax - \rho x$ bude minimalne norme. To je rezultat sljedećeg teorema.

Teorem 2.3.1. *Za* $x \neq 0$ *i proizvoljni* $\rho \in \mathbb{R}$ *je* (ρ, x) *svojstveni par matrice* $A - \frac{r}{x^T x}x^T$*, gdje je* $r = Ax - \rho x$*. Matrica* $\delta A = -\frac{r}{x^T x}x^T$ *ima normu* $\|\delta A\|_2 = \frac{\|r\|_2}{\|x\|_2}$*. Pri tome je* $\|r\|_2$ *minimalna ako je* $\rho = \frac{x^T Ax}{x^T x}$*.*

Dokaz. Vidimo da je

$$(A - \frac{r}{x^T x}x^T)x = Ax - rx = \rho x.$$

Također je

$$\| - \frac{r}{x^T x}x^T\|_2 = \frac{\|rx^T\|_2}{\|x\|_2^2} = \frac{\sqrt{\mathrm{tr}((rx^T)^T rx^T)}}{\|x\|_2^2} = \frac{\sqrt{\mathrm{tr}(x^T xr^T r)}}{\|x\|_2^2} = \frac{\|r\|_2\|x\|_2}{\|x\|_2^2} = \frac{\|r\|_2}{\|x\|_2}.$$

Iz teorema o projekciji, znamo da je $\|Ax - \rho x\|$ minimalno ako je $Ax - \rho x$ okomito na x pa dobijemo

$$\rho x = \frac{xx^T}{x^T x}Ax \implies \rho = \frac{x^T Ax}{x^T x}.$$

□

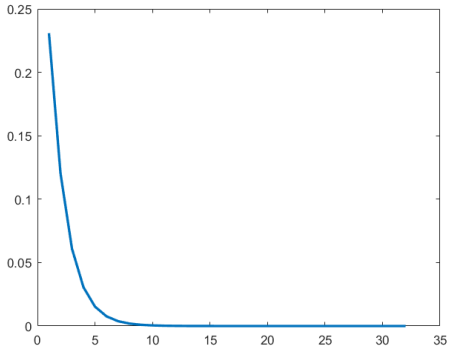
Dakle, kad imamo sve sastojke, algoritam je sljedeći:

Algoritam 1 Metoda potencija

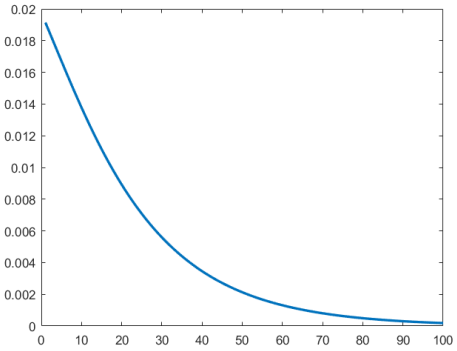
1: **Inputs:** Matrica $A \in \mathbb{R}^{n\times n}$, početni vektor $x_0 \in \mathbb{R}^n$, maksimalan broj koraka K
2: Postavi $y^{(0)} = \frac{x_0}{\|x_0\|}$
3: **for** $k = 1$ **to** K (ili do konvergencije po nekom kriteriju) **do**
4: $\quad x^{(k+1)} = Ay^{(k)}$
5: $\quad y^{(k+1)} = \frac{x^{(k+1)}}{\|x^{(k+1)}\|}$
6: **end for**
7: **Output:** Vrati posljednji izračunati $y^{(k)}$

Uočimo sljedeće: u dokazu konvergencije algoritma, pokazali smo da ako je jedna svojstvena vrijednost po apsolutnoj vrijednosti veća od ostalih, onda reziduali konvergiraju u 0 brzinom koja je određena omjerom dvije apsolutno najveće svojstvene vrijednosti. Dakle, što je najveća vrijednost odvojenija od ostalih, imat ćemo bržu konvergenciju, ali i ako je druga najveća blizu, konvergencija bi mogla biti spora.

Za usporedbu, promotrimo sljedeći primjer. Generirali smo dvije matrice reda 100, jednu kojoj je najveća svojstvena vrijednost 2, a ostale su 1, i jednu kojoj je najveća svojstvena vrijednost 1.05, a ostale 1. Teorija kaže da u oba slučaja imamo konvergenciju, ali je jasno kada će ona biti brža, što vidimo i iz analize reziduala koje dobijemo provedemo li metodu potencija na navedene dvije matrice.



Slika 2.1: Reziduali za matricu sa spektrom $\sigma_A = \{1, 2\}$ - brzi pad u 0.



Slika 2.2: Reziduali za matricu sa spektrom $\sigma_A = \{1, 1.05\}$ - spori pad u 0.

Vidimo da za prvu matricu metoda potencija staje nakon manje od 100 koraka (postavili smo kriterij zaustavljanja koji algoritam prekida kada se dvije uzastopne iteracije po normi razlikuju za manje od 10^{-10}), dok druga matrica treba punih 100 koraka. To nam daje ideju kada je dobro koristiti metodu potencija, a kada je bolje pokušati na drugi način.

Naravno, ponekad želimo računati i druge svojstvene vrijednosti osim one apsolutno najveće. Ipak, kako će nam za računanje difuzijskog preslikavanja biti dovoljno gledati najveće (jednu ili nekoliko njih), pozabavimo se metodom koja poopćuje metodu potencija.

2.3.2 Metoda iteracija potprostora

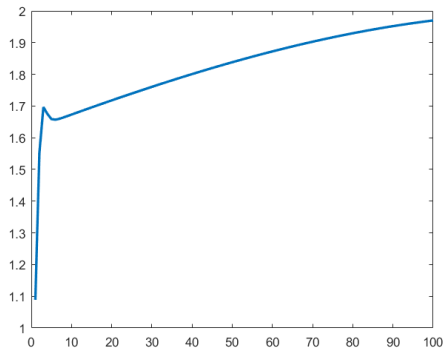
Cilj metode iteracija potprostora je za matricu A naći l-dimenzionalni A-invarijantni potprostor $\mathcal{V}$ reprezentiran pomoću matrice $V \in \mathbb{R}^{n\times l}$ koja zadovoljava $V^T V = I_l$ (tj., stupci u V su vektori ortonormirane baze za $\mathcal{V}$). Ovdje ćemo pretpostaviti da svojstvene vrijednosti od A zadovoljavaju $|\lambda_1| \geq |\lambda_2| \geq .... \geq |\lambda_l| > |\lambda_{l+1}| \geq ... \geq |\lambda_n|$.

Najjednostavniji pokušaj poopćenja metode potencija bi bilo da umjesto jednog polaznog vektora, krećemo s l linearno nezavisnih vektora $x^{(1,0)}, x^{(2,0)}, ..., x^{(l,0)}$, posložimo ih u matricu $X^{(0)} \in \mathbb{R}^{n\times l}$ i radimo identičan korak kao u metodi potencija, samo s matricom, to jest, imamo $X^{(k)} = A^k X^{(0)}$. Naravno, jer metoda potencija radi kako smo opisali ranije, taj niz će težiti matrici ranga 1 - jer svi stupci će težiti prema istom svojstvenom vektoru koji odgovara dominantnoj svojstvenoj vrijednosti matrice A. Taj neželjeni efekt možemo spriječiti tako da u svakom koraku nakon djelovanja matricom A ortonormiramo dobivene vektore stupce (naravno, to ćemo moći ako djelovanje matrice A ne promijeni rang, što će biti zadovoljeno uz pretpostavku o svojstvenim vrijednostima). Sve u svemu, algoritam je dan kao:

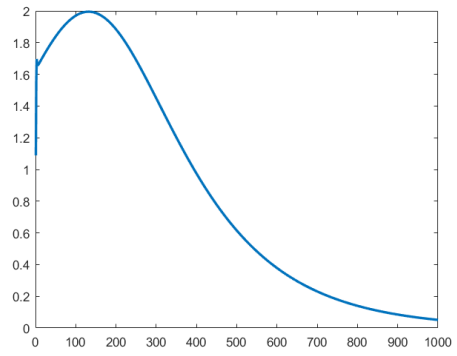
Algoritam 2 Metoda iteracija potprostora

1: **Inputs:** Matrica $A \in \mathbb{R}^{ntimesn}$, početna matrica $X^{(0)} \in \mathbb{R}^{n\times l}$, maksimalan broj koraka K
2: Odredi $Y^{(0)}$ QR-faktorizacijom $X^{(0)} = Y^{(0)}R^{(0)}$
3: **for** $k = 1$ **to** K (ili do konvergencije po nekom kriteriju) **do**
4: $\quad X^{(k+1)} = AY^{(k)}$
5: $\quad$ Provedi QR faktorizaciju $X^{(k+1)} = Y^{(k+1)}R^{(k+1)}$
6: **end for**
7: **Output:** Vrati posljednji izračunati $Y^{(k)}$

Usporedit ćemo metode potencija i iteracija potprostora na novom primjeru. Sada generiramo matricu sa svojstvenim vrijednostima $2, -1.99, 1$. Generirana matrica je dobivena ortogonalnom transformacijom dijagonalne matrice $D = diag(d)$, $d = (2, -1.99, 1, 1, ..., 1)$, pa je svakako dijagonalizabilna i vodeća svojstvena vrijednost je odvojena od ostalih (ali ne puno), pa očekujemo konvergenciju, koja bi u slučaju metode potencija mogla biti spora. Prikažimo grafički dobivene reziduale:

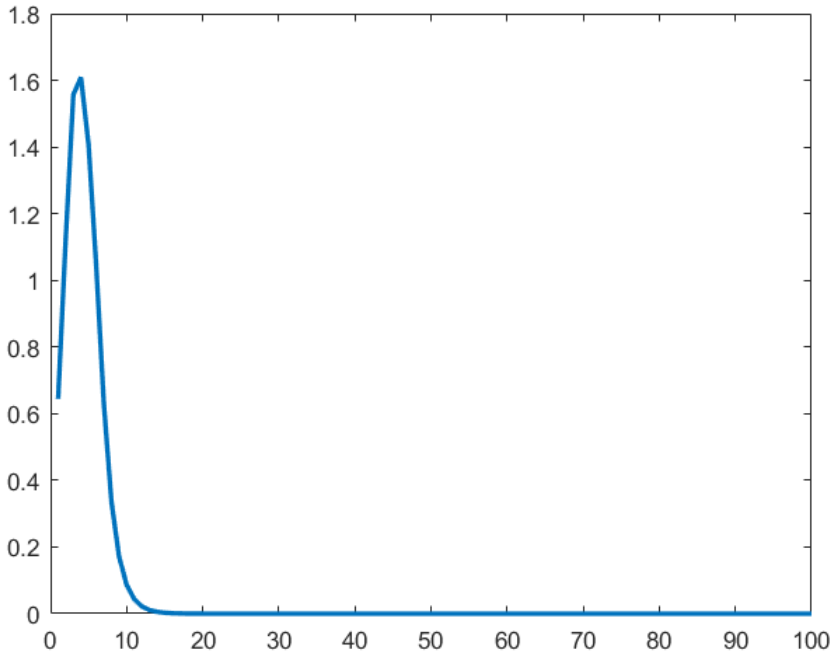


Slika 2.3: Reziduali metode potencija u 100 koraka



Slika 2.4: Reziduali metode potencija u 1000 koraka

Na prvi pogled za metodu potencija konvergencije uopće nema (ako gledamo samo 100 koraka). Ipak, metoda će raditi, ali je konvergencija spora, te joj treba 1000 koraka da bismo dobili zaadovoljavajuće rezultate - to znači da broj koraka mora porasti za čitav red veličine, što nipošto nije poželjno. Za usporedbu, promotrimo reziduale metode iteracija.



Slika 2.5: Reziduali metode iteracija

Dakle, metoda iteracija potprostora će dati dobre rezultate puno brže - daje obje svojstvene vrijednosti koje trebamo i još u manje koraka. Ovo je klasični problem jednostavnih metoda - iako mogu raditi vrlo dobro (i teorija to opravdava), u praksi se isplati malo pomučiti kako bi se dobili bolji rezultati.

2.4 Problem svojstvenih vrijednosti simetričnih matrica

Metode koje smo opisali ranije nigdje ne koriste pretpostavku simetričnosti matrice A. Kako ćemo u algoritmu za računanje difuzijskog preslikavanja raditi sa simetričnom matricom, bilo bi korisno analizirati algoritme koji će koristiti tu pretpostavku kako bi dobili na brzini i preciznosti. Naš prvi korak u formiranju takvog algoritma je Jacobijeva rotacija.

2.4.1 Jacobijeva rotacija

Prvo promotrimo kako dijagonalizirati 2×2 simetričnu matricu

$$M = \begin{bmatrix} \alpha & \gamma \\ \gamma & \beta \end{bmatrix}, \ \gamma \neq 0.$$

Koristit ćemo ravninsku rotaciju (ortogonalnu matricu)

$$J = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix},$$

gdje kut θ određujemo tako da je $J^T MJ$ dijagonalna matrica. Imamo

$$J^T MJ = \begin{bmatrix} \alpha\cos^2(\theta) + \beta\sin^2(\theta) - 2\gamma\sin(\theta)\cos(\theta) & (\alpha-\beta)\sin(\theta)\cos(\theta) + \gamma(\cos^2(\theta) - \sin^2(\theta)) \\ (\alpha-\beta)\sin(\theta)\cos(\theta) + \gamma(\cos^2(\theta) - \sin^2(\theta)) & \alpha\sin^2(\theta) + 2\gamma\sin(\theta)\cos(\theta) + \beta\cos^2(\theta) \end{bmatrix}$$

iz čega slijedi da dijagonalnost dobijemo zadovoljavanjem uvjeta $(\alpha - \beta)\sin(\theta)\cos(\theta) + \gamma(\cos^2(\theta) - \sin^2(\theta)) = 0$.
Rješavanjem jednadžbe po θ dobijemo $\operatorname{ctg}(2\theta) = \frac{\beta-\alpha}{2\gamma}$. Uvođenjem oznake $\zeta = \operatorname{ctg}(2\theta)$ i $\tau = \operatorname{tg}(\theta)$, te iz identiteta za tangens $\zeta = \frac{1-\tau^2}{2\tau}$ dobijemo kvadratnu jednadžbu

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

Manje po modulu rješenje jednadžbe je

$$\tau = \frac{\operatorname{sgn}(\zeta)}{\left(|\zeta| + \sqrt{1+\zeta^2}\right)},$$

iz čega lako odredimo parametre rotacije $\cos(\theta) = \frac{1}{\sqrt{1+\tau^2}}$, $\sin(\theta) = \tau\cos(\theta)$.
Sada lako vidimo da je

$$J^T MJ = \begin{bmatrix} \alpha - \gamma\tau & 0 \\ 0 & \beta + \gamma\tau \end{bmatrix}.$$

Sljedeći korak je vidjeti kako primijeniti Jacobijeve rotacije na veće matrice.

2.4.2 Jacobijeva metoda

Za simetričnu $n \times n$ matricu A promatramo njenu 2×2 podmatricu $\tilde{A}$, u kojoj iz A izdvajamo par indekasa (i, j):

$$A = \begin{array}{c} \\ \\ \\ i \\ \\ j \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccc} & & i & & j & & \end{array} \\ \begin{bmatrix} * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ * & * & \odot & * & \triangle & * & * \\ * & * & * & * & * & * & * \\ * & * & \triangle & * & \square & * & * \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \end{bmatrix} \end{array},$$

$$\tilde{A} = \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ji} & a_{jj} \end{bmatrix} = \begin{bmatrix} \odot & \triangle \\ \triangle & \square \end{bmatrix}.$$

Sada odredimo pripadnu Jacobijevu rotaciju za $\tilde{A}$, $\tilde{U} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$, kako smo opisali u prethodnoj sekciji. Potom $\tilde{U}$ uložimo u $n \times n$ matricu U tako da $\tilde{U}$ zauzima ista mjesta u U kao $\tilde{A}$ u A, pa dobijemo

$$U = \begin{array}{c} \\ \\ \\ i \\ \\ j \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccc} & & i & & j & & \end{array} \\ \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & \cos(\theta) & & \sin(\theta) & & \\ & & & 1 & & & \\ & & -\sin(\theta) & & \cos(\theta) & & \\ & & & & & 1 & \\ & & & & & & 1 \end{bmatrix} \end{array}.$$

Sada vršimo transformaciju $U^T A U$ - to je jedan korak Jacobijeve metode. Jednom transormacijom sličnosti poništili smo 2 izvandijagonalna elementa. Naravno, u idućem koraku, biranjem novog para indeksa (i', j') možemo poništiti ono što smo napravili u prethodnom koraku i nule pretvoriti u nenul elemente. Stoga je bitan redoslijed kojim ćemo poništavati vandijagonalne elemente.

Da bismo ovaj postupak formalizirali i mogli opravdati, uvodimo sljedeću funkciju:

$$\Omega(A) = \|A - diag(A)\|_2 = \sqrt{\sum_{i \neq j} a_{ij}^2}.$$

Funkcija Ω mjeri koliko je velik izvandijagonalni dio od A. Ako je $\Omega(A) = 0$, onda je A dijagonalna. Dakle, cilj je nizom Jacobijevih rotacija $A^{(k+1)} = (U^{(k)})^T A^{(k)} U^{(k)}$ dobiti $\lim_{k \to \infty} \Omega(A^{(k)}) = 0$.

Propozicija 2.4.1. *U jednoj primjeni gore opisane Jacobijeve rotacije* $A^{(k+1)} = (U^{(k)})^T A^{(k)} U^{(k)}$, *s parom indeksa* (i_k, j_k), *je* $\Omega^2(A^{(k+1)}) = \Omega^2(A^{(k)}) - 2(a^{(k)}_{i_k,j_k})^2$.

Dokaz. Jasno je da je $\|A^{(k+1)}\|_2^2 = \Omega^2(A^{(k)}) + \sum_{i=1}^n (a^{(k+1)}_{ii})^2$. Također, Jacobijeva rotacija je ortogonalna sličnost, pa ne mijenja normu, a na dijagonali mijenja samo pripadna 2 elementa na mjestima (i_k, i_k) i (j_k, j_k), pri čemu je (jer Jacobijeva rotacija na 2×2 matrici također čuva normu)

$$(a^{(k+1)}_{i_k,i_k})^2 + (a^{(k+1)}_{j_k,j_k})^2 = (a^{(k)}_{i_k,i_k})^2 + (a^{(k)}_{j_k,j_k})^2 + 2(a^{(k)}_{i_k,j_k})^2.$$

Odavde dobivamo

$$\Omega^2(A^{(k+1)}) = \|A^{(k)}\|_2^2 - \sum_{l=1, l\neq i_k, j_k}^{n} (a^{(k)}_{ll})^2 - ((a^{(k)}_{i_k,i_k})^2 + (a^{(k)}_{j_k,j_k})^2 + 2(a^{(k)}_{i_k,j_k})^2) = \Omega^2(A^{(k)}) - 2(a^{(k)}_{i_k,j_k})^2.$$

□

Propozicija 2.4.2. *Ako u svakom koraku Jacobijeve metode biramo za pivotni element na mjestu* (i_k, j_k) *tako da vrijedi*

$$|a_{i_k,j_k}| = \max_{i\neq j} |a_{i,j}|,$$

onda je

$$\Omega(A^{(k+1)}) \leq \Omega(A^{(k)}) \sqrt{1 - \frac{2}{n(n-1)}},$$

pa je zadovoljeno i $\lim_{k\to\infty} \Omega^2(A^{(k)}) = 0$.

Dokaz. Jasno, jer je element na mjestu (i_k, j_k) po modulu najveći vandijagonalni, imamo

$$\Omega^2(A^{(k)}) \leq n(n-1)(a_{i_k,j_k})^2 \implies (a_{i_k,j_k})^2 \geq \frac{\Omega^2(A^{(k)})}{n(n-1)}.$$

Stoga iz prethodne propozicije imamo i

$$\Omega^2(A^{(k+1)}) \leq \Omega^2(A^{(k)}) - 2\frac{\Omega^2(A^{(k)})}{n(n-1)} = \Omega^2(A^{(k)})(1 - \frac{2}{n(n-1)}).$$

□

Odavde vidimo da se nizom iteracija približavamo dijagonalnosti. Ipak, nemamo još garanciju konvergencije jednoj fiksnoj dijagonalnoj matrici. To je rezultat sljedećeg teorema (dokaz u [1], teorem 10.4.1):

Teorem 2.4.3. *Niz iteracija Jacobijeve metode $A^{(k)}$ je konvergentan s limesom*

$$\lim_{k\to\infty} A^{(k)} = \begin{bmatrix} \lambda_{\pi(1)} & & & \\ & \lambda_{\pi(2)} & & \\ & & \ddots & \\ & & & \lambda_{\pi(n)} \end{bmatrix},$$

gdje je $\pi : \{1, 2, ..., n\} \to \{1, 2, ..., n\}$ neka permutacija.

Trebamo navesti još par bitnih napomena u vezi implementacije Jacobijeve metode.

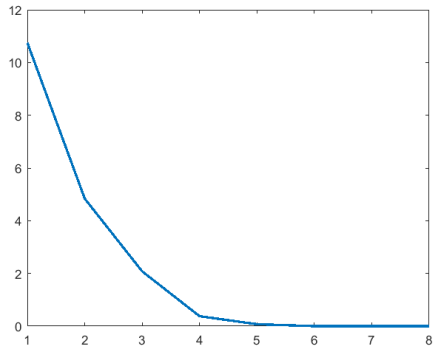
Prvo je važno uočiti da u aritmetici konačne preciznosti ne možemo egzaktno ni odrediti ni provesti korak Jacobijeve metode. Stoga za rotaciju u svakom koraku prvo provjerimo je li pripadni vandijagonalni element dovoljno (po apsolutnoj vrijednosti) malen, te ako je, postavimo ga na 0, i nastavljamo na idući korak. Nadalje, iz 2.4.2 slijedi da, ako je pivot ispod $\zeta = \frac{\epsilon}{n(n-1)}$, možemo se zaustaviti s iteracijama.

Iduća napomena je o biranju pivota. U svakom koraku je traženje najvećeg po apsolutnoj vrijednosti elementa skupo, možemo transformacije uzimati tako da se krećemo *ciklički* po matrici i poništavamo vandijagonalne elemente. Time gubimo rezultate prethodnih propozicija, ali ako sve vandijagonalne elemente smanjimo ispod tražene točnosti, zadovoljni smo rezultatom. Ideja obilaska matrice je po stupcima ili retcima:

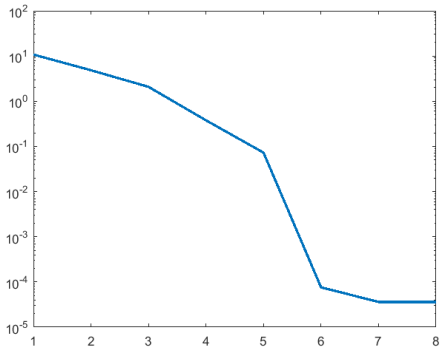
$$\begin{bmatrix} * & \overset{1}{\rightarrow} & \overset{2}{\rightarrow} & \overset{3}{\rightarrow} & \overset{4}{\rightarrow} & \overset{5}{\rightarrow} \\ & * & \overset{6}{\rightarrow} & \overset{7}{\rightarrow} & \overset{8}{\rightarrow} & \overset{9}{\rightarrow} \\ & & * & \overset{10}{\rightarrow} & \overset{11}{\rightarrow} & \overset{12}{\rightarrow} \\ & & & * & \overset{13}{\rightarrow} & \overset{14}{\rightarrow} \\ & & & & * & \overset{15}{\rightarrow} \\ & & & & & * \end{bmatrix}, \text{ ili } \begin{bmatrix} * & \overset{1}{\downarrow} & \overset{2}{\downarrow} & \overset{4}{\downarrow} & \overset{7}{\downarrow} & \overset{11}{\downarrow} \\ & * & \overset{3}{\downarrow} & \overset{5}{\downarrow} & \overset{8}{\downarrow} & \overset{12}{\downarrow} \\ & & * & \overset{6}{\downarrow} & \overset{9}{\downarrow} & \overset{13}{\downarrow} \\ & & & * & \overset{10}{\downarrow} & \overset{14}{\downarrow} \\ & & & & * & \overset{15}{\downarrow} \\ & & & & & * \end{bmatrix}.$$

Cikluse ponavljamo sve dok *svi* vandijagonalni elementi ne postanu dovoljno mali.

Promotrimo primjer rada metode u stvarnosti. Generiramo slučajnu simetričnu matricu reda 50, te ju pustimo kroz Jacobijevu metodu. Ograničili smo broj koraka na 100, a toleranciju postavljamo na 10^{-8} (tj. ako su svi vandijagonalni elementi apsolutno manji od 10^{-8}, zaustavljamo se). Broj koraka koji je trebao metodi je svega 8, te je dala ispravan rezultat (razlika spektra u odnosu na MATLAB-ov rezultat je reda veličine 10^{-10}). Također, proces možemo vizualizirati i grafički i prikazati kako izgleda konvergencija:

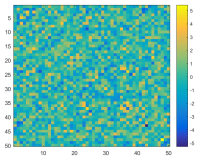


Slika 2.6: Prikaz niza $\Omega(A^{(k)})$ na našoj matrici

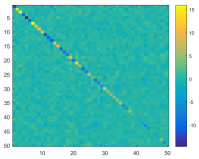


Slika 2.7: Prikaz istog niza u logaritamskoj skali

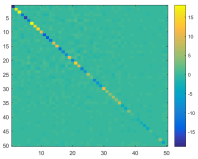
Prikažimo i matrice $A^{(k)}$ pomoću toplinske skale prije svakog koraka:



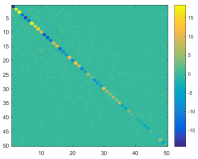
$k = 1$



$k = 2$



$k = 3$



$k = 4$

(Nakon četvrtog koraka u grafičkom prikazu se ne vidi razlika, ali je vidljivo dovoljno da zaključimo da algoritam radi.)

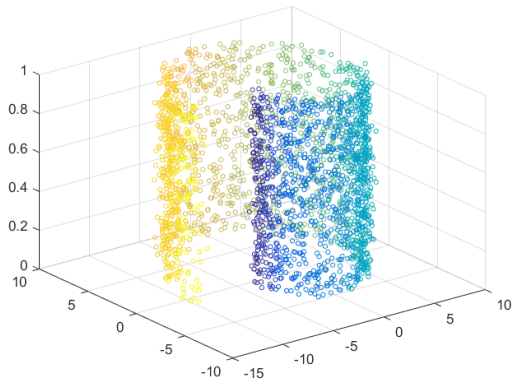
Napomena: Sve ranije opisane metode su validne u problemu svojstvenih vrijednosti kojim ćemo se baviti u kasnijim poglavljima. Ipak, zbog brzine, u većini algoritama ćemo za računanje svojstvenih vrijednosti matrice koristiti MATLAB-ovu inačicu funkcije za rješavanje svojstvenih problema **eig**(), koja u ovisnosti o postavljenim opcijama i o svojstvima matrice na koju se primjenjuje koristi QZ algoritam ili faktorizaciju Choleskog.

Poglavlje 3

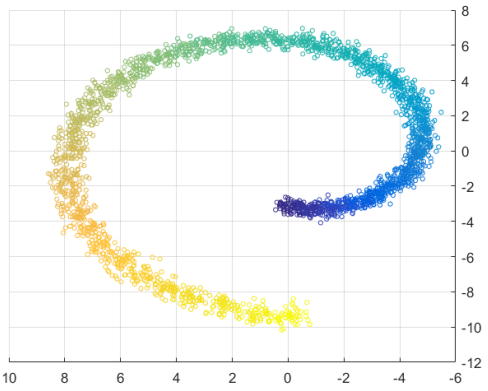
Difuzijska preslikavanja

3.1 Uvod i ideja

Difuzijska preslikavanja su metoda transformacije koordinata polaznog skupa podataka u one u kojima bismo trebali jasnije uočiti globalnu strukturu. Primjerice, u nekim problemima euklidska udaljenost nije dobra mjera jer će podaci, iako su možda euklidski bliski, dolaziti iz različitih klasa ili su strukturno udaljeni. Promotrimo za primjer tzv. "Swiss roll" skup podataka.



Slika 3.1: Skup podataka "Swiss roll"

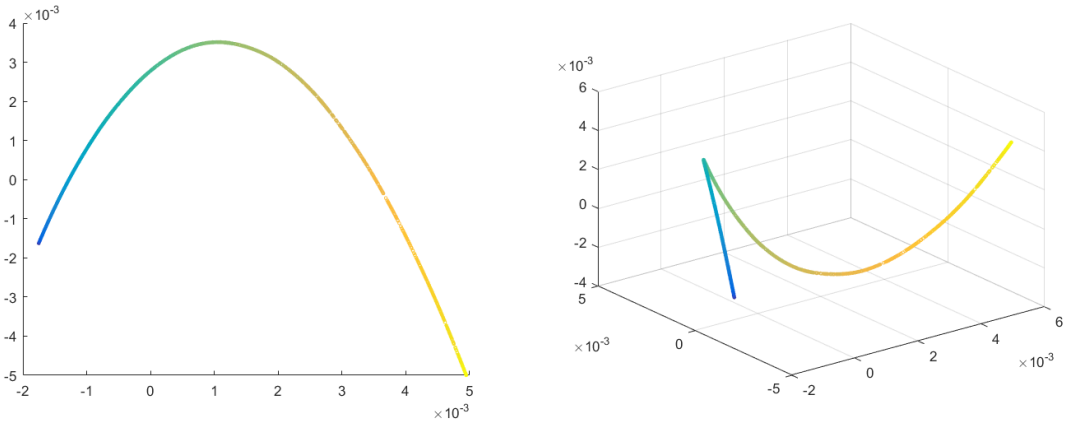


Slika 3.2: Projekcija na *xy*-ravninu.

Možemo uočiti da skup, iako je trodimenzionalnom prostoru, leži u dvodimenzionalnoj plohi u tom prostoru, to jest, može se parametrizirati s 2 slobodna parametra. Naravno, jasno je da ploha na kojoj skup leži nije linearna pa linearno preslikavanje u $\mathbb{R}^2$ neće biti dobra metoda redukcije dimenzije. Stoga je potrebno problem sagledati s druge strane i osmisliti novi pristup.

Novi pogled na problem je da udaljenost između podataka ne mjerimo uzimajući obzir samo dvije promatrane točke, već cijeli skup podataka. Ideja s kojom polazimo je da za dvije točke udaljenost mjerimo preko udaljenosti bliskih im točaka - ako su dvije točke blizu sličnim točkama, ili se na grafu u obje točke može doći sličnim putevima, smatrat ćemo da su bliske, te želimo da se to odrazi u novim koordinatama. Upravo to je pristup koji predlažu difuzijska preslikavanja.

Primjerice, promotrimo kako difuzijska preslikavanja utječu na skup podataka s kojim smo krenuli:



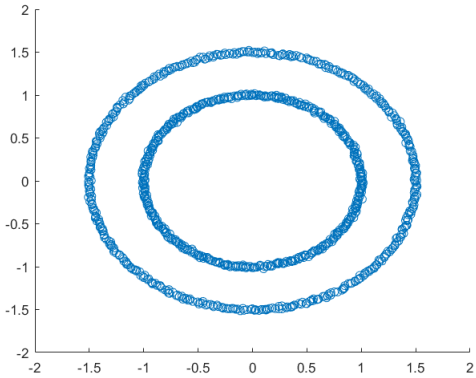
Transformacija u 2D Transformacija u 3D

Slika 3.3: Difuzijsko preslikavanje primijenjeno na prethodni ”Swiss roll” skup podataka.

Možemo da difuzijsko preslikavanje ”razmotava” polazni skup podataka - time pojednostavljujemo strukturu na kojoj se podaci nalaze što nam može znatno olakšati baratanje podacima i njihovu analizu.

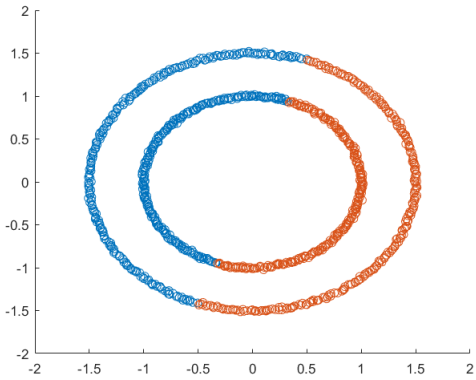
Dakle, ideja je iz lokalne strukture podataka i njihovih udaljenosti, za definiciju difuzijske udaljenosti koristiti *sve* dostupne podatke, ne samo promatrana dva, time ćemo dobiti uvid u globalnu strukturu podataka kojima baratamo, te ćemo moći donositi nove zaključke i koristiti poznate metode koje će raditi efikasnije na trasnformiranim podacima.

Još jedna primjena difuzijskih preslikavanja je u transformaciji koordinata radi smanjenja dimenzije. Neki od najpopularnijih pristupa tom problemu uključuju linearne modele. Ipak, vrlo često nam podaci nisu linearno separabilni. Promotrimo sljedeći primjer, u kojem nam je zadan skup podataka koji možemo vidjeti na slici.



Slika 3.4: Zadani skup podataka

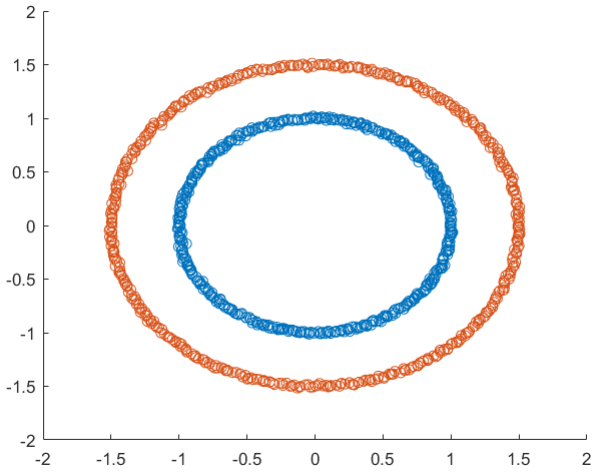
Skup podataka kojim baratamo se sastoji od dvije kružnice, na koje smo dodali normalan šum. Jasno je da imamo 2 klastera podataka koji su vrlo prirodno odvojeni. No valja pripaziti, prirodno su odvojeni jer nam je kada ih vidimo jasno da se radi o dvije kružnice, no u nekom apstraktnom okruženju, gdje možda nemamo jasnu geometriju i nije odmah vidljivo kako raspodijeliti podatke, iako možda znamo da dolaze iz 2 klastera. Kako ih odvojiti u tom slučaju? Možemo pokušati s jednim od standardnih algoritama za klasteriranje, tzv. *k-sredine* (*k-means*).



Slika 3.5: Podjela u 2 segmenta koristeći *k-means* algoritam - boje odgovaraju klasteru koji je algoritam pridružio svakoj točki

Vidimo da rezultat nije zadovoljavajući - podjela je linearna, te je rezultat to da svaku od kružnica podijelimo da 2 polukružnice, što nam ne odgovara, ako je uočljivo da podaci dolaze iz 2 klastera koji čine koncentrične kružnice.

Pogledajmo kako to rade difuzijska preslikavanja - ako pratimo ideju koju smo opisali, podaci na istim kružnicama će biti bliski, jer iz svake točke na jednoj kružnici u nekom broju koraka lako možemo doći do svake točke na istoj kružnici. Skokovi s jedne kružnice na drugu pak rezultiraju time da su u novim koordinatama podaci jako udaljeni.



Slika 3.6: Podjela u 2 segmenta koristeći *k-means* algoritam na podatke koje smo prethodno preslikali difuzijskim preslikavanjem

Ovdje su rezultati upravo ono što bismo očekivali - 2 klastera su upravo dvije kružnice koje čine naši podaci. Dakle, nelinearnost u podacima je svakako vrlo česta pojava pa je za redukciju dimenzije potrebno imati i takav algoritam koji to uzima u obzir.

U ovom poglavlju definiramo nužne pojmove za konstrukciju difuzijskog preslikavanja danog skupa podataka. Nakon toga, objašnjavamo algoritam kojim ih računamo, te potom objašnjavamo kako difuzijske koordinate možemo koristiti u redukciji dimenzije polaznog problema.

3.2 Konstrukcija difuzijskog preslikavanja

Neka je $(X, \mathcal{A}, \mu)$ prostor s mjerom. U ovom radu, X će biti skup podataka koje imamo, pa se radi o konačnom skupu, stoga će definicije biti nešto jednostavnije od originalnih

koje je osmislio autor algoritma u [6] i [12]. Shodno tome, μ će biti mjera na konačnom skupu, pa će najčešće u pitanju biti brojeća mjera, dok će $\mathcal{A}$ jednostavno biti partitivni skup $\mathcal{P}(X)$ (iako sve definicije možemo generalizirati na općeniti prostor s mjerom). U ovom kontekstu definiramo pojam *jezgre*, koji će predstavljati svojevrsnu mjeru sličnosti među točkama skupa X.

Definicija 3.2.1. Difuzijska jezgra *je preslikavanje* $k : X \times X \to \mathbb{R}$ *za koje vrijedi:*

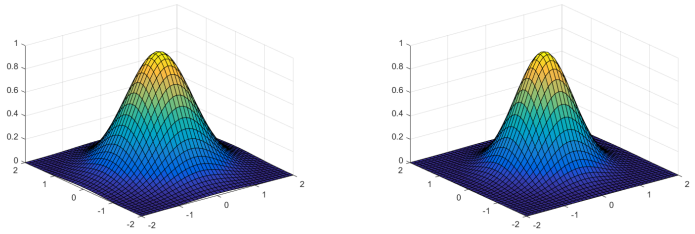
i.) $k(x, y) = k(y, x)$ *(simetrija)*

ii.) $k(x, y) \geq 0$ *(nenegativnost)*

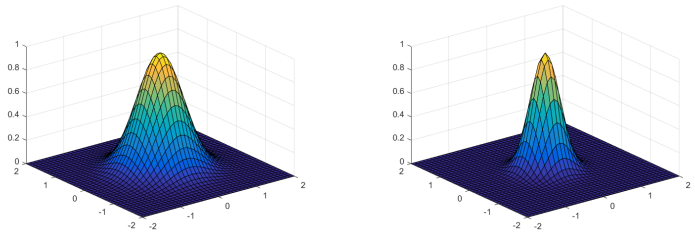
Najčešći primjer jezgre u primjeni, te jezgre koju ćemo najviše koristiti u ovom radu je tzv. *Gaussovska jezgra*:

$$k(x, y) = e^{-\|x-y\|_2^2/\sigma^2}.$$

O jezgri razmišljamo kao o apriornoj mjeri udaljenosti, odnosno ona definira lokalnu geometriju našeg skupa podataka X. Ako koristimo Gaussovsku jezgru, uočavamo da ona dopušta parametar σ, čiji će izbor biti netrivijalan zadatak, jer sama funkcija uvelike ovisi o tom parametru, kao što možemo vidjeti na primjeru - mala promjena parametra uvelike utječe na oblik funkcije.



(a) $\sigma = 1$ (b) $\sigma = 0.75$



(c) $\sigma = 0.5$ (d) $\sigma = 0.25$

Slika 3.7: Utjecaj parametra σ na Gaussovsku jezgru. Uočimo da manji σ "sužava" zvono krivulje.

Sljedeći korak je definirati mjeru udaljenosti koja će bliskima smatrati točke koje prirodno slijede strukturu skupa podataka. U tu svrhu ćemo konstruirati Markovljev lanac koji će prirodno reprezentirati kretanje po skupu podataka, odnosno šetnju po grafu kojeg definiraju naši podaci.

Prvo definiramo za $x \in X$:

$$d(x) = \int_X k(x,y)\, d\mu(y) = \sum_{y \in X} k(x,y).$$

Kako uzimamo da je X konačan skup, možemo njegove elemente enumerirati tako da je $X = \{a_1, a_2, ..., a_n\}$. Tada jezgru i funkciju d možemo gledati kao matrice:

$$K_{ij} = k(a_i, a_j),\ i, j = 1, 2, ..., n$$

$$d_i = d(a_i) = \sum_{j=1}^{n} K_{ij},\ i = 1, 2, ..., n$$

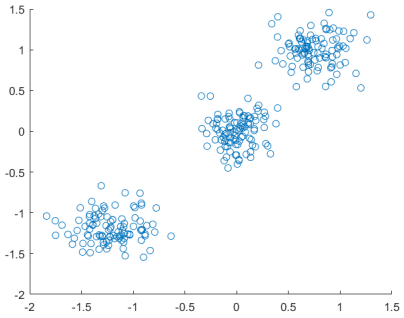
$$D = diag(d_i)_{i=1}^{n}$$

Sada definiramo matricu $P = D^{-1}K$. Uočimo dva bitna svojstva matrice P:

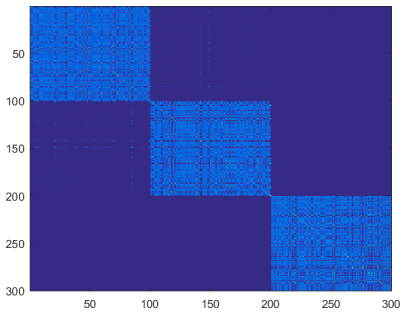
i.) $P_{ij} \geq 0\ i, j = 1, ..., n$

ii.) $\sum_j P_{ij} = \sum_j \frac{K_{ij}}{d_i} = \frac{1}{d_i} \sum_j K_{ij} = \frac{d_i}{d_i} = 1$

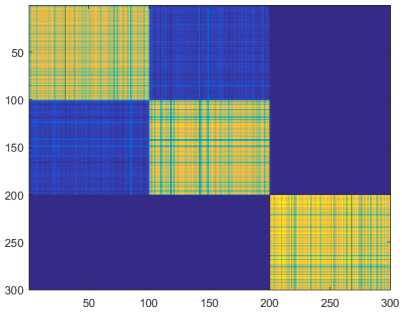
Dakle, radi se o po retcima stohastičkoj matrici. Na nju gledamo upravo kao matricu prijelaza Markovljevog lanca koji opisuje slučajnu šetnju po grafu podataka s kojima radimo. Uočimo, jer smo zahtijevali da je jezgra k simetrična, možemo tu šetnju gledati kao šetnju po neusmjerenom grafu. Vjerojatnost prijelaza iz stanja a_i u a_j je proporcionalna težini brida (a_i, a_j) u odnosu na ostale bridove iz a_i. Promotrimo ponašanje tog lanca i matrice prijelaza na proizvoljno generiranom skupu podataka s 3 klastera.



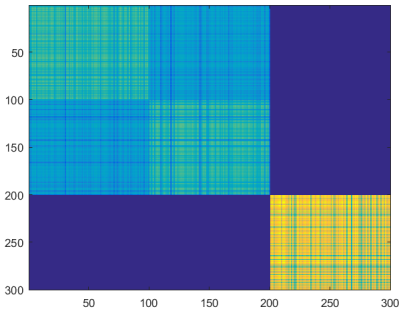
Slika 3.8: Polazni skup podataka od 3 klastera.



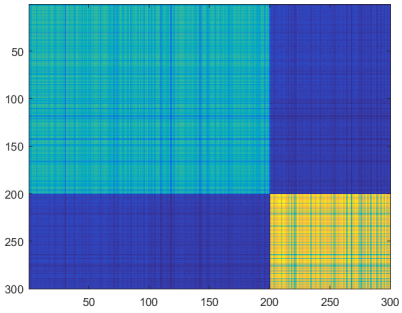
Slika 3.9: Matrica prijelaza P polaznog skupa podataka.



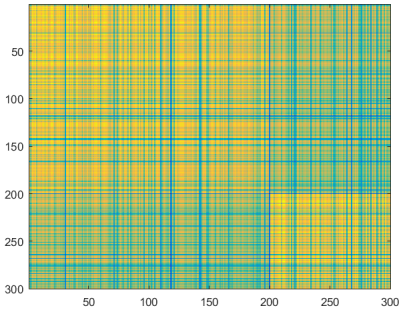
Slika 3.10: 10-koračna matrica prijelaza.



Slika 3.11: 100-koračna matrica prijelaza.



Slika 3.12: 1000-koračna matrica prijelaza, uočavamo grupiranje dva klastera.



Slika 3.13: 10000-koračna matrica prijelaza, uočavamo grupiranje svih klastera.

Dakle, ako polazni skup podataka ima 3 klastera, oni su jasno vidljivi na matrici P (naravno, to je zato jer smo podatke i sortirali po klasterima). Također je uočljivo da, kako potencija t raste, to se klasteri grupiraju, prvo se iz dva bliža klastera formira jedan veći, a onda nakon dovoljno puno koraka se formira superklaster koji sadrži sve podatke. U svakom slučaju, uočavamo da je broj koraka koji uzimamo u definiciji šetnje vrlo bitan - ponekad ćemo htjeti gledati više koraka jer se tako klasteri jasnije vide (jer tako svrstavamo izolirane točke u neke od klastera), ali ipak prevelik broj koraka rezultira time da su nam bliski i oni podaci za koje ne želimo da budu. Stoga ćemo, umjesto samo jedne funkcije, definirati *familiju difuzijskih preslikavanja*.

Definicija 3.2.2. Familiju difuzijskih preslikavanja D_t *definiramo kao familiju funkcija*

$D_t : X \times X \to \mathbb{R},\ t \geq 0$, *gdje je*

$$D_t(a_i, a_j) = \sum_{x \in S} \left(\mathbb{P}_t(a_i, x) - \mathbb{P}_t(a_j, x)\right)^2 w(x), \tag{3.1}$$

pri čemu je $\mathbb{P}_t$ *t−koračna vjerojatnost prijelaza (odnosno t-ta potencija matrice prijelaza P), a w težinska funkcija, npr, najčešće ćemo uzimati* $w(x) = \frac{1}{d(x)}$.

Prethodna definicija uključuje sve elemente koje smo trebali - koristi lokalnu informaciju o udaljenostima točaka (pri formiranju matrice P), broj koraka u šetnji na grafu, te pri definiranju udaljenosti dvaju točaka koristi informaciju o cijelom skupu podataka. Konkretno, udaljenost među točkama je manja kada su točke na sličan način povezane s ostalim točkama skupa.

Sljedeći korak je naći transformaciju koordinata u kojima će euklidska udaljenost odgovarati (nekoj) difuzijskoj udaljenosti u originalnim koordinatama. U tu svrhu, analizirajmo neka svojstva matrice P.

3.2.1 Svojstva matrice prijelaza

Matrica P je retčano stohastička, pa je $\lambda = 1$ njena svojstvena vrijednost, a pripadni svojstveni vektor je $e = (1, 1, ..., 1)^T$.

$$(Pe)_i = \sum_{j=1}^{n} p_{ij} = \sum_{j=1}^{n} \frac{k(a_i, a_j)}{d(a_i)} = 1 \implies Pe = 1 \cdot e.$$

Kako je P stohastička po retcima, vrijedi da je $\|P\|_\infty = \max\limits_{1 \leq i \leq n} \sum_{j=1}^{n} |P_{ij}| = 1$. Nadalje, kako znamo da je $\lambda = 1$ jedna svojstvena vrijednost, a znamo i da je spektralni radijus infimum po svim konzistentnim matričnim normama (teorem 2.1.2), zaključujemo da je spektralni radijus matrice P jednak 1.

Uz pretpostavku ireducibilnosti (koja će u našem slučaju, jer koristimo Gaussovu jezgru, biti zadovoljena), vrijedi Perron-Frobenijusova teorija: $\lambda = 1$ je jednostruka svojstvena vrijednost i njen pripadni svojstveni vektor se može odabrati tako da ima pozitivne komponente.

Imamo $P = D^{-1}K$, a kako su svi elementi matrice K pozitivni, isto vrijedi i za elemente matrice D, a kako je ona dijagonalna s pozitivnom dijagonalom, to je regularna, pozitivno definitna i ima drugi korijen (koji dobijemo korjenovanjem elemenata na dijagonali). Stoga je $P = D^{-1}K = D^{-\frac{1}{2}}\left(D^{-\frac{1}{2}}KD^{-\frac{1}{2}}\right)D^{\frac{1}{2}}$ pa je P slična simetričnoj matrici $P_s = D^{-\frac{1}{2}}KD^{-\frac{1}{2}}$. Posebno, P je dijagonalizabilna, te su joj svojstvene vrijednosti realne.

Ako je $P_s = U\Lambda U^T$ spektralna dekompozicija matrice P_s, tada je

$$P = D^{-\frac{1}{2}} P_s D^{\frac{1}{2}} = D^{-\frac{1}{2}} U\Lambda U^T D^{\frac{1}{2}} = V\Lambda V^{-1},\ V = D^{-\frac{1}{2}} U. \tag{3.2}$$

Jer je $P = D^{-1}K$, $D = diag(d_i)_{i=1}^n$, $d_i = \sum_j k(a_i, a_j) = \sum_j K_{ij}$, imamo

$$d^T P = d^T D^{-1} K = \begin{bmatrix} d_1 & d_2 & \dots & d_n \end{bmatrix} \begin{bmatrix} \frac{1}{d_1} & 0 & \dots & 0 \\ 0 & \frac{1}{d_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{d_n} \end{bmatrix} K = e^T K = d^T. \tag{3.3}$$

Vidimo da je d^T lijevi svojstveni vektor matrice P, pa normiranjem (koje je dobro definirano zbog konačnosti skupa stanja X), dobijemo stacionarnu distribuciju π:

$$\pi = \frac{d^T}{\sum_i d_i},\ \ \pi P = \pi,\ \ \sum_i \pi_i = 1,\ \ \pi_i \geq 0,\ i = 1, 2, ..., n$$

3.2.2 Transformacija koordinata

Cilj nam je naći transformaciju koordinata za koje će vrijediti da difuzijska udaljenost originalnih koordinata odgovara euklidskoj udaljenosti transformiranih. Stoga krećemo od definicije difuzijske udaljenosti:

$$D_t(a_i, a_j)^2 = \sum_{x \in \{a_1, a_2, ..., a_n\}} \left(\mathbb{P}_t(a_i, x) - \mathbb{P}_t(a_j, x)\right)^2 w(x). \tag{3.4}$$

Kao što smo ranije rekli, uzimamo težine $w(x) = \frac{1}{d(x)}$. Definiramo:

$$\Gamma_i = \begin{pmatrix} \mathbb{P}_t(a_i, a_1) \\ \mathbb{P}_t(a_i, a_2) \\ \vdots \\ \mathbb{P}_t(a_i, a_n) \end{pmatrix},\ i = 1, 2, ..., n.$$

Tada 3.4 postaje:

$$D_t(a_i, a_j)^2 = \left(\Gamma_i - \Gamma_j\right)^T D^{-1} \left(\Gamma_i - \Gamma_j\right) = \|\Gamma_i - \Gamma_j\|^2_{D^{-1}},$$

pri čemu je $\|x\|^2_{D^{-1}} := x^T D^{-1} x$. Uočimo da je $\Gamma_i = (P^t(i, :))^T$. Prisjetimo se spektralne dekompozicije od P:

$$P = V\Lambda V^{-1},\ P^t = V\Lambda^t V^{-1},\ V = D^{-\frac{1}{2}} U,\ U^T U = I,$$

$$P^t = V\Lambda V^{-1} = \begin{bmatrix} v_1(i)\lambda_1^t & v_2(i)\lambda_2^t & \dots & v_n(i)\lambda_n^t \end{bmatrix} V^{-1}.$$

Uz oznaku $V^{-1} = \tilde{V} = U^T D^{\frac{1}{2}}$, uočavamo da su $\begin{pmatrix} v_1(i)\lambda_1^t & v_2(i)\lambda_2^t & \dots & v_n(i)\lambda_n^t \end{pmatrix}$ koordinate retka $P^t(i,:)$ u bazi redaka matrice $\tilde{V}$.

Dakle, imamo

$$\Gamma_i = (P^t(i,:))^T = \tilde{V}^T \underbrace{\begin{pmatrix} \lambda_1^t v_1(i) \\ \lambda_2^t v_2(i) \\ \vdots \\ \lambda_n^t v_n(i) \end{pmatrix}}_{\Psi_i^{(t)}} = \tilde{V}^T \Psi_i^{(t)}, \tag{3.5}$$

$$D_t(a_i, a_j)^2 = (\Psi_i^{(t)} - \Psi_j^{(t)})^T \underbrace{\tilde{V} D^{-1} \tilde{V}^T}_{U^T D^{\frac{1}{2}} D^{-1} D^{\frac{1}{2}} U} (\Psi_i^{(t)} - \Psi_j^{(t)}) = \|\Psi_i^{(t)} - \Psi_j^{(t)}\|^2. \tag{3.6}$$

Konačno, možemo definirati difuzijsko preslikavanje kao preslikavanje:

$$a_i \mapsto \Psi_i^{(t)} = \begin{pmatrix} \lambda_1^t v_1(i) \\ \lambda_2^t v_2(i) \\ \vdots \\ \lambda_n^t v_n(i) \end{pmatrix} = \Lambda^T V(i,:)^T. \tag{3.7}$$

Sve u svemu, algoritam za računanje difuzijskog preslikavanja je dan kao:

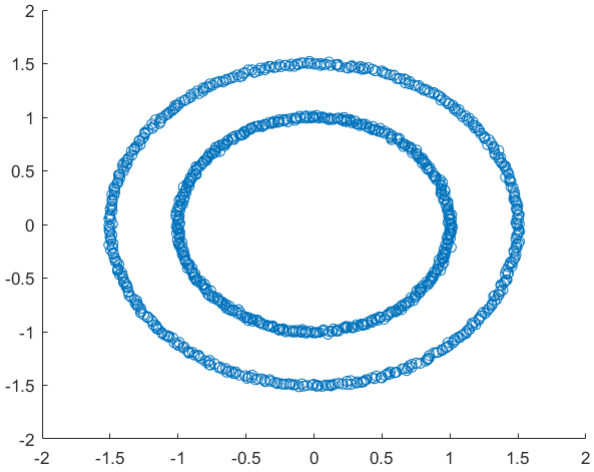
Algoritam 3 Difuzijsko preslikavanje - algoritam

1: **Inputs:** Podaci $a_1, a_2, ..., a_n \in \mathbb{R}^m$, hiperparametri $t \in \mathbb{N}$, $\sigma > 0$, reducirana dimenzija $k \in \{1, 2, ..., m\}$.
2: Za svaki par podataka $i, j = 1, ..., n$ izračunaj vrijednost jezgre $K_{ij} = k(a_i, a_j)$.
3: Izračunaj $d = Ke$, $e = \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix} \in \mathbb{R}^n$ i $D = diag(d) \in \mathbb{R}^{n \times n}$.
4: Izračunaj prijelaznu matricu $P_s = D^{-\frac{1}{2}} K D^{-\frac{1}{2}}$.
5: Izračunaj spektralnu dekompoziciju matrice $P_s = U\Lambda U^T$.
6: Izračunaj $V = D^{-\frac{1}{2}} U$.
7: Izračunaj $\Psi^{(t)} = \Lambda^t V^T$.
8: **Output:** Vrati reducirani model dimenzije k kao prvih k redaka: $\Psi_k^{(t)} = \Psi^{(t)}(1 : k, 1 : n)$.

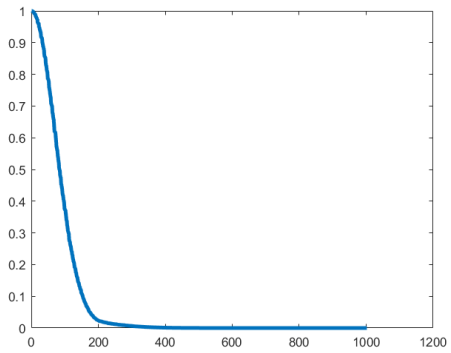
Dakle, redukciju dimenzije vršimo uzimanjem k koordinata, a prethodno provedeni račun nam daje i ideju kako odabrati k - iz izraza 3.7 vidimo da svaku koordinatu množimo

pripadnom svojstvenom vrijednosti (i to uz potenciju t). Kako su sve svojstvene vrijednosti po apsolutnoj vrijednosti manje od 1, za velike t će nam neke od tih koordinata biti vrlo blizu 0, te ih možemo i zanemariti.

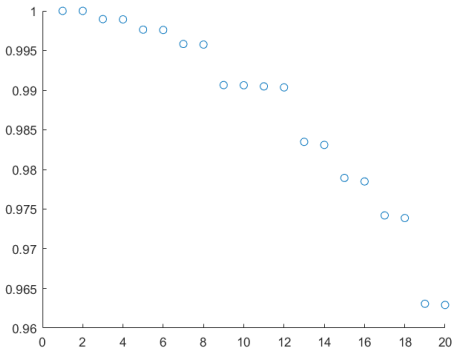
Ako se vratimo na uvodni primjer s dvjema kružnicama, možemo promotriti kako za njih izgledaju svojstvene vrijednosti i pripadna matrica P.



Slika 3.14: Skup podataka s dvjema koncentričnim kružnicama.



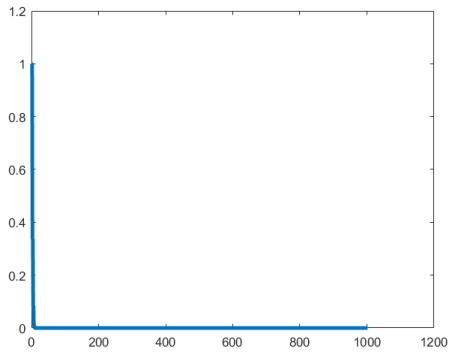
Slika 3.15: Grafički prikaz svih svojstvenih vrijednosti, uočavamo brz pad.



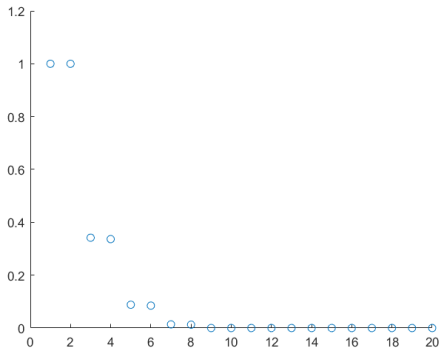
Slika 3.16: Grafički prikaz najvećih 20 svojstvenih vrijednosti, nešto sporiji pad.

Iz grafičkog rikaza vidimo da svojstvene vrijednosti možda i ne padaju prebrzo, ali svakako padaju. Ipak, ne treba se obeshrabriti i odmah pomisliti da ćemo morati gledati

velik broj dimenzija kako bismo dobili zadovoljavajuće rezultate. Primjerice, vrlo dobre rezultate smo postigli sa samo 2 difuzijske koordinate, jer je potencija na koju smo dizali matricu prijelaza bila vrlo visoka, te je tada prikaz pripadnih potencija znatno drugačiji.

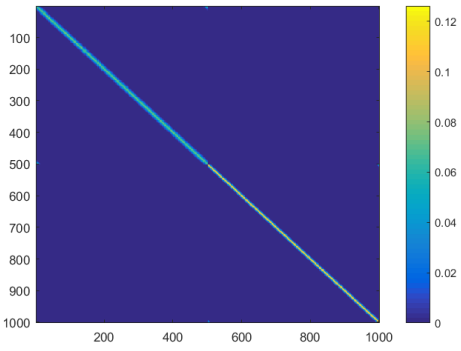


Slika 3.17: Grafički prikaz svih svojstvenih vrijednosti na potenciju 1000.

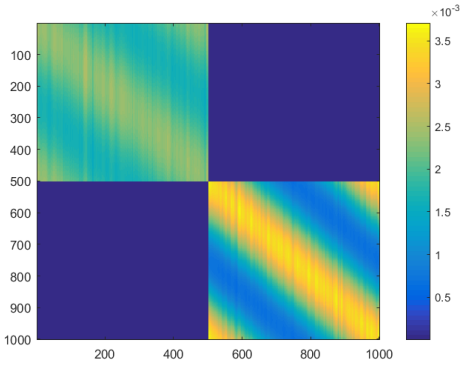


Slika 3.18: Grafički prikaz najvećih 20 svojstvenih vrijednosti na potenciju 1000.

Kako je u ovom primjeru cilj bio odvojiti podatke koje dolaze iz istih klastera, potencija t na koju smo dizali matricu P je bila reda veličine kao i cijeli skup podataka (u našem slučaju, imali smo 1000 točaka, po 500 s obje kružnice), kako bismo dobili da su podaci koji su bliski oni od kojih možemo doći nekom (možda dugačkom, ali vjerojatnom) šetnjom bez preskakanja kružnica. Stoga je korisno i pogledati kako izgleda matrica P dignemo li je na potenciju koju smo primijenili.



Slika 3.19: Grafički prikaz matrice P, ne uočavamo klastere.



Slika 3.20: Grafički prikaz matrice P^{1000}, jasno uočljiva 2 klastera.

Ovdje vidimo da, iako je iz svake točke moguće doći u samo njoj bliske točke, u 1000 koraka možemo doći u bilo koju drugu točku na *istoj* kružnici - stoga je lako podijeliti podatke u 2 klastera, jer je vjerojatnost prijelaza između točaka na istoj kružnici puno veća od one između točaka na različitim kružnicama, što onda daje da su difuzijske udaljenosti točaka na istoj kružnici male, a na različitima velike i lako dobijemo ispravno klasteriranje.

Poglavlje 4

Lokalno linearna ulaganja

4.1 Motivacija i definicija

4.1.1 Motivacija

U brojnim problemima strojnog učenja, dan nam je velik broj opažanja, mjerenja ili značajki (eng. *features*) nekog objekta. Primjena uobičajenih metoda, nadziranih (kao što je klasifikacija) ili nenadziranih (npr. klasteriranje), često može biti problematična zbog procjene velikog broja parametara ili visoke dimenzionalnosti samih podataka. Stoga je standardna procedura prije obrade podataka provesti neku od metoda redukcija dimenzije.

Tradicionalne metode redukcije dimenzije su uglavnom linearne, te uključuju odabir podskupova mjerenja i preslikavanja u nižedimenzionalne linearne potprostore. Naravno, razvijene su i kompleksnije metode za nelinearna preslikavanja, kao neuronske mreže ili višedimenzionalno skaliranje (eng. *Multidimensional scaling - MDS*). Ovisno o problemu koji rješavaju, te metode mogu imati raznih poteškoća, kao odabir krivog modela, vremenska i prostorna složenost, ili zahtijevanje velikog broja podataka kako bi metoda radila ispravno.

Saul i Roweis su u [13] predstavili konceptualno jednostavnu, ali vrlo moćnu metodu redukcije dimenzije: lokalno linearno ulaganje (LLE). Ona polazi od pretpostavke da su podaci smješteni na nekoj kompleksnoj (nelinearnoj) strukturi u visokodimenzionalnom prostoru, dok je sama mnogostrukost znatno manje dimenzije od ambijentnog prostora. Uz poznavanje samih podataka i njihovih odnosa (udaljenosti, odnosno metrike - za svaki par podataka (x_i, x_j) možemo odrediti njihovu udaljenost $\mu(x_i, x_j)$) s ciljem otkrivanja te strukture promatramo samo točke koje su relativno bliske, odnosno, ako su dvije točke udaljene, pretpostavljamo da je informacija o vezi među njima praktično beskorisna ili nepostojeća.

Pažljivi će čitatelj uočiti da smo ovom pričom donekle opisali i ideje kojima smo bili vođeni pri konstrukciji difuzijskih preslikavanja. To nije slučajno, ova metoda (a pogotovo njena verzija s jezgrom) ima vrlo sličan učinak kao i difuzijska preslikavanja, a i teorijska pozadina im se u velikoj mjeri poklapa.

4.1.2 Definicija lokalno linearnog ulaganja

Polazimo od skupa podataka $X = \{x_1, x_2, ..., x_n\} \subset \mathbb{R}^d$ i metrike μ na $X \times X$ - dakle, znamo podatke i njihove međusobne udaljenosti. Cilj je odrediti preslikavanje koje će prikazati podatke u nižoj dimenziji kao $y_1, ..., y_n \in \mathbb{R}^l$.

Možemo pomoću μ za svaki x_i odrediti skup njegovih k najbližih susjeda $X_i = \{x_1^{(i)}, ..., x_k^{(i)}\}$, te ćemo skupove najbližih susjeda koristiti u idućem koraku.

Prvi korak je za svaki i pripadni x_i reprezentirati najbolje moguće kao linearnu kombinaciju njegovih k najbližih susjeda, odnosno odrediti težine $\hat{w_{ij}},\ i = 1, ..., n,\ j = 1, ..., k$ u raspisu

$$\sum_{i=1}^{n} \left\| x_i - \sum_{j=1}^{k} \hat{w_{ij}} x_j^{(i)} \right\|_2^2 \to \min_{\hat{w_{ij}}}.$$

Ako definiramo $w_{ij} = \hat{w_{ij}}$ za $x_j \in X_i$ i $w_{ij} = 0$ za $x_j \notin X_i$, onda je $x_i - \sum_{j=1}^{k} \hat{w_{ij}} x_j^{(i)} = x_i - \sum_{j=1}^{n} w_{ij} x_j$ - stoga možemo minimizirati po w_{ij}.

Drugi korak je, jednom kad smo odredili težine w_{ij} prethodnog koraka, odrediti $y_1, ..., y_n \in \mathbb{R}^l$ tako da je

$$\sum_{i=1}^{n} \left\| y_i - \sum_{j=1} w_{ij} y_j \right\|_2^2 \to \min_{y_1,...,y_n}.$$

4.1.3 Numerički algoritam

Precizirat ćemo prvo kako računamo težine w_{ij} iz prvog koraka. Definiramo $e = (1, .., 1)^T \in \mathbb{R}^n,\ T_i = (x_1^{(i)}, ..., x_k^{(i)}),\ w_i = (w_{i1}, ..., w_{in})^T$. Tada je $\sum_{j=1}^{k} w_{ij} = 1 \iff e^T w_i = 1$.

$$\|x_i - \sum_{j=1}^{k} w_{ij} x_j^{(i)}\|_2^2 = \|x_i(e^T w_i) - T_i w_i\|_2^2 = \|(x_i e^T - T_i) w_i\|_2^2 =$$
$$= w_i^T (x_i e^T - T_i)^T (x_i e^T - T_i) w_i = w_i G_i w_i, \text{ gdje je}$$
$$G_i = (x_i e^T - T_i)^T (x_i e^T - T_i).$$

Stoga je ekvivalentan zapis početnog koraka

$$\sum_{i=1}^{n} w_i^T G_i w_i \to \min_{w_{ij}}, \text{ uz uvjet } \sum_{j=1}^{k} w_{ij} = 1, \ i = 1, ..., n.$$

Problem minimizacije rješavamo korištenjem Lagrangeovih multiplikatora $\lambda = (\lambda_1, ..., \lambda_n)$:

$$L = \sum_{i=1}^{n} w_i^T G_i w_i - \sum_{i=1}^{n} \lambda_i (e^T w_i - 1),$$
$$\frac{\partial L}{\partial w_i} = 2G_i w_i - \lambda_i e \implies \frac{\partial L}{\partial w_i} = 0 \iff w_i = \frac{1}{2}\lambda_i G_i^{-1} e,$$
$$\frac{\partial L}{\partial \lambda_i} = e^T w_i - 1 \implies \frac{\partial L}{\partial \lambda_i} = 0 \iff e^T w_i = 1.$$

Iz ovih uvjeta dobijemo izraze za λ_i i w_i, $i = 1, ..., n$:

$$w_i = \frac{1}{2}\lambda_i G_i^{-1} e,$$
$$1 = e^T w_i = \lambda_i \frac{1}{2} e^T G_i^{-1} e,$$
$$\Downarrow$$
$$\lambda_i = \frac{2}{e^T G_i^{-1} e}, \ w_i = \frac{G_i^{-1} e}{e^T G_i^{-1} e}.$$

Dakle, dobili smo eksplicitni izraz za težine w_i, $i = 1, ..., n$.

Napomena:

1. Ako je G_i regularna matrica, onda je pozitivno definitna, kao i G_i^{-1}, te je $e^T G_i^{-1} e > 0$.

2. Ako je dimenzija d strogo manja od broja promatranih najbližih susjeda k, onda je G_i najviše ranga d, pa je G_i singularna i inverz G_i^{-1} ne postoji.

3. Čak i ako je $d > k$, to ne znači da je G_i regularna, pa ne možemo nužno koristiti inverz.

4. Jedno rješenje, koje ćemo koristiti u algoritmima, je regularizacija - umjesto G_i koristimo $G_i + \epsilon I_k$, s tim da se postavlja pitanje izbora dobrog $\epsilon > 0$.

Sljedeći korak je ulaganje u $\mathbb{R}^l$ - treba odrediti vektore $y_1, ..., y_n \in \mathbb{R}^l$ takve da je:

$$\sum_{i=1}^{n} \left\| y_i - \sum_{j=1}^{n} w_{ij} y_j \right\|_2^2 \to \min_{y_1,...,y_n}.$$

Također ćemo postaviti sljedeće dodatne uvjete:

$$\sum_{i=1}^{n} y_i = 0, \quad \frac{1}{n} \sum_{i=1}^{n} y_i y_i^T = I_l.$$

Prvi uvjet je tu radi "centriranja" rješenja u ishodište. Naime, ako nađemo rješenje $(y_i)_{i=1}^n$ problema koje zadovoljava prvi uvjet, onda i svako rješenje oblika $(\hat{y}_i)_{i=1}^n$ uz $\hat{y}_i = y_i + a$ rješava problem za proizvoljan $a \in \mathbb{R}^l$.

Drugi uvjet dodajemo kako bi izbjegli grupiranje rješenja u točku. Zaista, svako rješenje $y_i = a,\ i = 1, 2, ..., n$ za fiksan $a \in \mathbb{R}^l$ će biti rješenje polaznog problema.

Ako postavimo

$$Y = (y_1, ..., y_n)^T \in \mathbb{R}^{n \times l},\ \hat{w}_i = (w_{i1}, ..., w_{in})^T \in \mathbb{R}^n,\ W = (\hat{w}_1, ..., \hat{w}_n) \in \mathbb{R}^{n \times n},$$

polazni minimizacijski problem postaje

$$\sum_{i=1}^{n} \left\| y_i - \sum_{j=1}^{n} w_{ij} y_j \right\|_2^2 = \sum_{i=1}^{n} \left\| Y^T e_i - Y^T w_i \right\|_2^2 = \left\| Y^T (I - W^T) \right\|_F^2 =$$

$$\text{tr}((I - W^T)^T Y Y^T (I - W^T)) = \text{tr}(Y^T (I - W)^T (I - W) Y) = \text{tr}(Y^T M Y),\ M = (I - W)^T (I - W) = M^T.$$

Uočimo da je $We = e$, te $(I - W)e = 0,\ Me = 0$. Nadalje, imamo

$$\sum_{i=1}^{n} = 0 \iff Y^T e = 0, \quad \frac{1}{n} \sum_{i=1}^{n} y_i y_i^T = I_l \iff \frac{1}{n} Y^T Y = I_l.$$

Dakle, problem minimizacije koji rješavamo je:

$$\text{tr}(Y^T M Y) \to \min_Y,\ \frac{1}{n} Y^T Y = I_k,\ Y^T e = 0.$$

Uz zamjenu varijabli $\tilde{Y} = \frac{1}{\sqrt{n}} Y$ imamo $\tilde{Y}^T \tilde{Y} = I_l$ i $\tilde{Y}^T e = 0$. Jer je $Me = 0$, $\frac{e}{n}$ je normirani svojstveni vektor matrice M, a kako je M simetrična, može se dijagonalizirati u ortonormiranoj bazi. Stoga, jer rješavamo problem minimizacije traga po ortogonalnim matricama čiji su stupci okomiti na e, koji je svojstveni vektor pripadne svojstvene vrijednosti 0, možemo koristiti Ky Fanov teorem.

Teorem 4.1.1 (Ky Fan)**.** *Neka je $M \in \mathbb{R}^{n\times n}$ simetrična matrica sa svojstvenim vrijednostima $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$ i pripadnom matricom svojstvenih vektora $Z = (z_1, ..., z_n)$, $Mz_i = \lambda_i z_i,\ i = 1, ..., n,\ \ Z^T Z = I_n$. Tada je*

$$\min_{\substack{X\in\mathbb{R}^{n\times k}\\ X^T X = I_k}} \operatorname{tr}(X^T M X) = \lambda_1 + ... + \lambda_k.$$

Za $\lambda_k < \lambda_{k+1}$ se skup rješenja na kojima se minimum postiže može zapisati kao

$$\left\{ \begin{pmatrix} z_1 & ... & z_k \end{pmatrix} Q \ :\ Q \in \mathbb{R}^{k\times k} ortogonalna \right\}.$$

Neka su $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$ svojstvene vrijednosti od M i neka su $z_1 = \frac{e}{\sqrt{n}}, z_2, ..., z_n$ pripadni svojstveni vektori. Tada po prethodnom teoremu možemo zaključiti da je rješenje problema:

$$Y = \begin{pmatrix} z_2 & ... & z_{l+1} \end{pmatrix},\ \ Y^T Y = I_l\ \ Y^T e = 0.$$

Stoga je traženo ulaganje (matricu čitamo po recima):

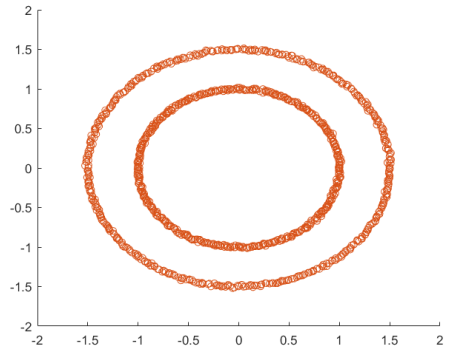
$$Y = \begin{pmatrix} y_1^T \\ \vdots \\ y_n^T \end{pmatrix},$$

odnosno vektore $y_1, ..., y_n$ čitamo kao retke matrice Y.

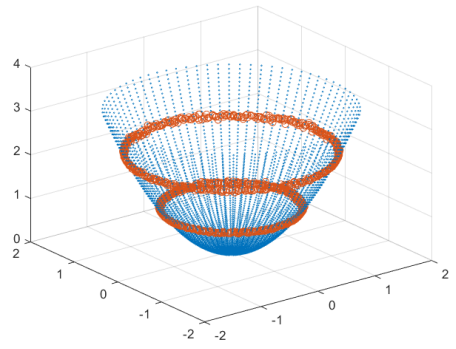
4.2 Jezgreni trik

U nekim situacijama nam može biti pogodno ulaganje podataka u prostor više dimenzije. Primjerice, može se dogoditi da s pogodno odabranim ulaganjem podaci uočimo jasnu pravilnost u podacima koja je ranije bila teško vidljiva.

Ako se vratimo na naš poznati primjer s dvjema kružnicama, znamo da one nisu linearno separabilne, ali ako ih uložimo prostor dimenzije 3, primjerice pomoću preslikavanja $(x, y) \to (x, y, x^2 + y^2)$ dobijemo skup koji se vrlo jednostavno linearno separira hiperravninom.



Slika 4.1: Skup podataka s dvjema kružnicama.



Slika 4.2: Isti skup nagon ulaganja u $\mathbb{R}^3$ (preslikavanje na paraboloid).

Ako imamo podatke $x_1, ..., x_n \in \mathbb{R}^d$, preslikavamo ih u $\mathbb{R}^D$, gdje je $D >> d$. Označimo to preslikavanje s ϕ, tj. imamo:

$$\mathbb{R}^d \ni x_i \to \phi(x_i) \in \mathbb{R}^D, \ i = 1, ..., n.$$

Označimo $\Phi = (\phi(x_1), ..., \phi(x_n)) \in \mathbb{R}^{D\times n}$ i $K = \Phi^T\Phi \in \mathbb{R}^{n\times n}$. Sada prethodno opisanu proceduru LLE provodimo na podacima $\phi(x_1), ..., \phi(x_n)$. Prisjetimo se kako u prvom koraku LLE tražimo k najbližih susjeda, a za to trebamo $\|\phi(x_i) - \phi(x_j)\|_2^2$:

$$\|\phi(x_i) - \phi(x_j)\|_2^2 = (\phi(x_i) - \phi(x_j))^T(\phi(x_i) - \phi(x_j)) =$$
$$\phi(x_i)^T\phi(x_i) - 2\phi(x_i)^T\phi(x_j) + \phi(x_j)^T\phi(x_j) = K_{ii} - 2K_{ij} + K_{jj}.$$

Jer je D potencijalno jako velika dimenzija, ovo može biti skupa operacija. Cilj je naći efikasan način da radimo s velikom dimenzijom tako da D ostane implicitno skrivena u matrici K, zajedno s ϕ. Kako je $K_{ij} = K(x_i, x_j)$, ideja je da zadamo jezgru $K(x, y)$ kao funkciju od $x, y \in \mathbb{R}^d$ s određenim svojstvima te ju potom povežemo s preslikavanjem ϕ. Promotrimo jedan primjer.

Primjer 4.2.1. Za $x, y \in \mathbb{R}^d$ stavimo $K(x, y) = (y^T x + b)^2$, gdje je b neka konstanta. Konstruirat ćemo preslikavanje $\phi : \mathbb{R}^d \to \mathbb{R}^D$ tako da je $K(x, y) = \phi(x)^T\phi(y)$.

$$K(x, y) = (y^T x + b)^2 = (\sum_{i=1}^{d} x_i y_i + b)^2 = (\sum_{i=1}^{d} x_i y_i)^2 + 2b\sum_{i=1}^{d} x_i y_i + b^2.$$

Analizirajmo svaki od izraza.

$$(\sum_{i=1}^{d} x_i y_i)^2 = (\sum_{i=1}^{d} x_i y_i)(\sum_{j=1}^{d} x_j y_j) = \sum_{i=1}^{d}\sum_{j=1}^{d} x_i y_i x_j y_j = \sum_{i=1}^{d} x_i^2 y_i^2 + \sum\sum_{i\neq j} x_i y_i x_j y_j =$$
$$\sum_{i=1}^{d}(x_i)^2(y_i)^2 + \sum_{i=2}^{d}\sum_{j=1}^{i-1}(\sqrt{2}x_i x_j)(\sqrt{2}y_i y_j).$$

Sada je

$$(y^T x + b)^2 = (\sum_{i=1}^{d} x_i y_i)^2 + 2b\sum_{i=1}^{d} x_i y_i + b^2 =$$
$$\sum_{i=1}^{d}(x_i)^2(y_i)^2 + \sum_{i=1}^{d}\sum_{j=2}^{i-1}(\sqrt{2}x_i x_j)(\sqrt{2}y_i y_j) + \sum_{i=1}^{d}\sqrt{2b}x_i\sqrt{2b}y_i + b\cdot b = \phi(y)^T\phi(x),$$

gdje je

$$\phi(x) = (x_1^2, ..., x_d^2,\ [\sqrt{2}x_i x_j; i = 2:d; j = 1:i-1],\ \sqrt{2b}x_1, ..., \sqrt{2b}x_d, b).$$

Dakle, uspjeli smo pomoću funkcije $K(x, y)$ (koju računamo u prostoru manje dimenzije $d << D$) izraziti kako računamo skalarne produkte $\phi(y)^T\phi(x)$. To omogućuje da provodimo LLE algoritam bez eksplicitnog računanja funkcije ϕ.

Neka su $\phi(x_i^{(1)}), ..., \phi(x_i^{(k)})$ najbližih k susjeda točke $\phi(x_i)$. Sada trebamo odrediti težine w_{ij}, ali kako taj račun uključuje minimizaciju u D-dimenzionalnom prostoru, moramo primijeniti trik s jezgrom. Imamo:

$$\sum_{i=1}^{n}\|\phi(x_i) - \sum_{j=1}^{k} w_{ij}\phi(x_i^{(j)})\|_2^2 \to \min_{w_{ij}},\quad \sum_{j=1}^{n} w_{ij} = 1,\ i = 1, ..., n.$$

Računamo

$$\sum_{i=1}^{n}\|1\phi(x_i) - \sum_{j=1}^{k} w_{ij}\phi(x_i^{(j)})\|_2^2 = \sum_{i=1}^{n}\|\sum_{j=1}^{k} w_{ij}(\phi(x_i) - \phi(x_i^{(j)}))\|_2^2.$$

Uvodimo oznaku $\Phi_i = (\phi(x_i) - \phi(x_i^{(1)}), ..., \phi(x_i) - \phi(x_i^{(k)})$ te dobivamo

$$\sum_{i=1}^{n}\|\sum_{j=1}^{k} w_{ij}(\phi(x_i) - \phi(x_i^{(j)}))\|_2^2 = \sum_{i=1}^{n}\|\Phi_i w_i\|_2^2 = \sum_{i=1}^{n} w_i^T \underbrace{\Phi_i^T\Phi_i}_{K_i} w_i.$$

Sve u svemu, minimizacijski problem postaje

$$\sum_{i=1}^{n} w_i^T K_i w_i \to \min_{w_{ij}}, \quad \sum_{j=1}^{n} w_{ij} = 1, \ i = 1, ..., n.$$

Rješenje koje je dano eksplicitnim formulama (uz pretpostavku da je dobro definirano) je

$$w_i = \frac{K_i^{-1} e}{e^T K_i^{-1} e}, \ i = 1, ..., n.$$

Dakle, imat ćemo riješen problem ako uspijemo K_i zapisati bez eksplicitnog računanja $\phi(x_i)$, tako da izbjegnemo račun u visokoj dimenziji.

Odredimo element na poziciji (p, q) u matrici K_i:

$$\begin{gathered} K_i(p,q) = (\phi(x_i) - \phi(x_i^{(q)}))^T (\phi(x_i) - \phi(x_i^{(p)})) = \\ \phi(x_i)^T \phi(x_i) - \phi(x_i)^T \phi(x_i^{(p)}) - \phi(x_i^{(q)})^T \phi(x_i) + \phi(x_i^{(q)})^T \phi(x_i^{(p)}) = \\ K(x_i, x_i) - K(x_i, x_i^{(q)}) - K(x_i^{(p)}, x_i) + K(x_i^{(p)}, x_i^{(q)}) = \\ K_{ii} - K_{i_p,i} - K_{i,i_q} + K_{i_p,i_q}. \end{gathered}$$

Konačno, sad kada imamo K_i izražen pomoću jezgre K, imamo formulu za računanje težina w_{ij}. Posljednji dio računa, određivanje novih koordinata Y provodimo na isti način kao i ranije (jer ono ne zahtijeva račun u dimenziji D). Time imamo cijeli teoretski "skupi" račun bez da smo igdje eksplicitno morali baratati visokodimenzionalnim objektima.

Još je ostalo jedno neriješeno pitanje - možemo li uvijek provesti ovaj postupak? Napravili smo ga za jednostavnu, polinomijalnu jezgru, ali što ako imamo općenitiju funkciju k, postoji li i za nju preslikavanja ϕ koje joj odgovara? Na to pitanje nam odgovor naje sljedeći teorem:

Teorem 4.2.1 (Mercer)**.** *Neka je dan skup podataka $X = \{x_1, ..., x_n\} \subset \mathbb{R}^d$ te neka je $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ simetrična, neprekidna i nenegativna funkcija koja generira pozitivno semidefinitnu matricu K preko:*

$$K_{ij} = k(x_i, x_j), \ i, j = 1, ..., n.$$

Tada postoji funkcija $\phi : \mathbb{R}^d \to \mathbb{R}^D$ (gdje je D dimenzija potencijalno puno veća od d) takva da vrijedi:

$$K_{ij} = k(x_i, x_j) = \phi(x_i)^T \phi(x_j), \ i, j = 1, ..., n.$$

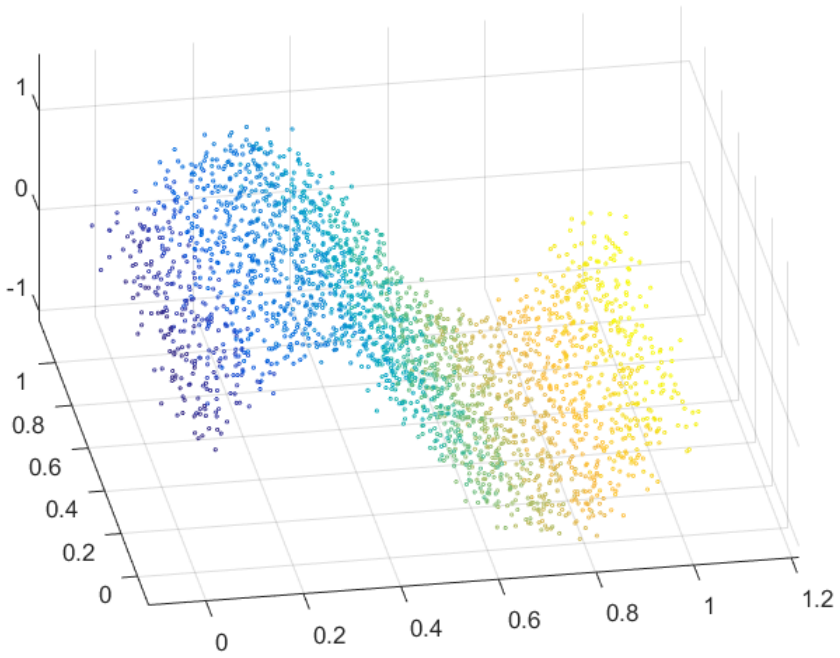
Ovim teoremom smo zaokružili priču - ako imamo funkciju k koja odgovara uvjetima teorema (a većina jezgara s kojima radimo, kao što su polinomijalna ili Gaussova, je takva) imamo opisan cijeli postupak provođenja algoritma LLE s jezgrom.

4.3 Klasifikacija

Primijenit ćemo do sad opisane algoritme za klasifikaciju. Problem klasifikacije je jedan od najčešćih problema podatkovne znanosti i strojnog učenja, a bavi se pridjeljivanjem odgovarajuće klase (od njih konačno mnogo) ulaznom podatku. Primjerice, naše algoritme ćemo testirati klasifikacijom rukom pisanih znamenki. Svaka slika će prikazivati neku od znamenki od 0 do 9, te je naš cilj svakoj slici pridružiti znamenku koju ona prikazuje.

4.3.1 Implementacija

U samom algoritmu LLE vidjeli smo da postoje 3 parametra koja treba zadati - broj najbližih susjeda koje uzimamo u obzir k, dimenziju prostora u koji preslikavamo podatke m i parametar regularizacije r. Algoritam je izuzetno osjetljiv na promjenu u svakom od parametara. Promotrimo kako njihovo variranje utječe na izlazne podatke algoritma.

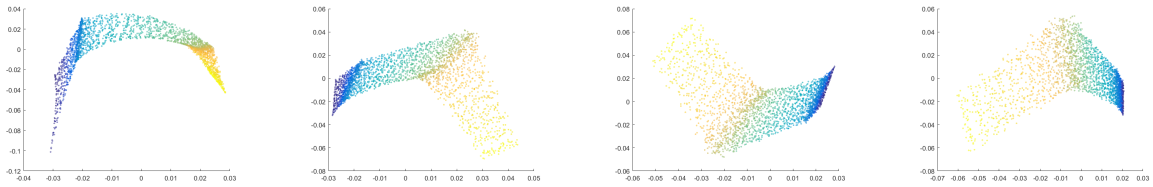


Slika 4.3: Ulazni skup podataka - podaci su generirani tako da su (x, y) koordinate 50×50 mreža na $[0, 1] \times [0, 1]$, a treća koordinata je $z = \sin(x)$, te je dodan normalan šum.

Primjer 4.3.1. Polazni skup podataka s kojima radimo je dvodimenzionalan - sinusoidalna ploha u $\mathbb{R}^3$. Stoga ćemo testirati LLE ulaganje u prostor dimenzije 2 iz dva razloga: jer to sugeriraju podaci i jer u ravnini lako možemo vizualizirati rezultate.

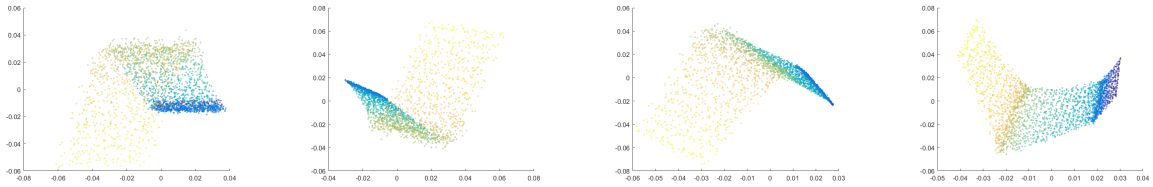
Algoritam ćemo testirati tako da variramo jedan parametar, a ostale držimo fiksnima na nekoj vrijednosti za koju smo procijenili dobre izlazne podatke (nema smisla uspoređivati utjecaj jednog parametra ako drugi potpuno kvari rezultate).

Način na koji podatke vizualiziramo je sljedeći: u polaznom skupu podataka, svakoj je točki pridružena boja, pomoću oznake npr. enumeracijom, koja je određena njihovom x i y koordinatom. Primjenom preslikavanja točaka u novi prostor, one zadržavaju svoju oznaku, tako da ista boja na različitim slikama odgovara istoj točki - time možemo pratiti kako se npr. grupacije bliskih točaka ponašaju u raznim verzijama preslikavanja.



(a) $k = 10$ (b) $k = 20$ (c) $k = 30$ (d) $k = 40$

Slika 4.4: Utjecaj broja susjeda k na LLE, uz $r = 2 \cdot 10^{-3}$, $m = 2$ - veći k "raspršuje" podatke.



(a) $r = 10^{-4}$ (b) $r = 5 \cdot 10^{-4}$ (c) $r = 10^{-3}$ (d) $r = 5 \cdot 10^{-3}$

Slika 4.5: Utjecaj parametra r na LLE, uz $k = 30$, $m = 2$ - r mijenja oblik plohe na kojima leže podaci.

Također, valja napomenuti da i izbor dimenzije prostora ulaganja m također bitno utječe na izlaz LLE algoritma (ali nemamo mogućnosti to vizualizirati, bar za velike m).

Iz rezultata vidimo da će izbor parametara biti vrlo bitan, ali i netrivijalan. Naime, utjecaj promejene svakog od parametara je uočljiv, ali nije očito kako ih odabrati. Ni znatnim

povećanjem niti smanjenjem ne dobivamo "ljepše" (odnosno u nekom smislu "bolje") ulaganje. Stoga ćemo se prvo kratko pozabaviti automatskim odabirom parametara.

Ideja za odabir parametara će doći iz *Analize glavnih komponenti* (eng. *Principal Component Analysis - PCA*). Naime, to je metoda koja ima vrlo sličan cilj kao i LLE -preslikati podatke u prostor manje dimenzije, ali tako da se originalna struktura promijeni što je manje moguće. PCA ima statistički pristup, te se preslikavanje traži uz uvjet da se varijabilnost u podacima što više objasni preslikavanjem u manju dimenziju (tj. da varijabilnost u "ostatku" dimenzije bude minimalna). Detaljnije o PCA se može pročitati u [11], a detaljna usporedba dvaju algoritama, kao i preciznija objašnjenja kako i zašto biramo parametre, je objašnjena u [7].

Prvo komentirajmo kako odabiremo dimenziju prostora ulaganja m. U PCA se ona ponekad određuje tako da zadamo udio varijance koji zahtijevamo da bude objašnjen ulaganjem u prostor manje dimenzije (naravno, on raste s dimenzijom - veća dimenzija, bolje objašnjeni podaci). Može se pokazati da je varijanca podataka koju smo objasnili dana kao zbroj svojstvenih vrijednosti matrice korelacije pripadnih svojstvenih vektora koje smo koristili u konstrukciji preslikavanja, a ona koju nismo zbroj preostalih svojstvenih vrijednosti. Kako će spektar matrice korelacije susjeda točke x_i odgovarati spektru naših matrica G_i, mi ćemo koristiti istu metodu za određivanje m_i - uzimat ćemo minimalan m_i tako da za svojstvene vrijednosti $\lambda_1, ..., \lambda_n$ od G_i vrijedi:

$$v \leq \frac{\sum_{j=1}^{m_i} \lambda_j}{\sum_{k=1}^{n} \lambda_k}.$$

(Važno je napomenuti da podrazumijevamo da je prvi korak sortirati svojstvene vrijednosti). Konačni m sada odabiremo kao najčešću vrijednost m_i.

Što se parametra r tiče, u PCA se varijanca koju nismo objasnili (odnosno koja nije sadržana u preslikanim podacima) procjenjuje kao prosjek svojstvenih vrijednosti onih vektora koji nisu korišteni u konstrukciji preslikavanja. Mi ćemo opet u svakom računu matrice G_i odrediti preostale svojstvene vrijednosti i njihov prosjek koristiti za r.

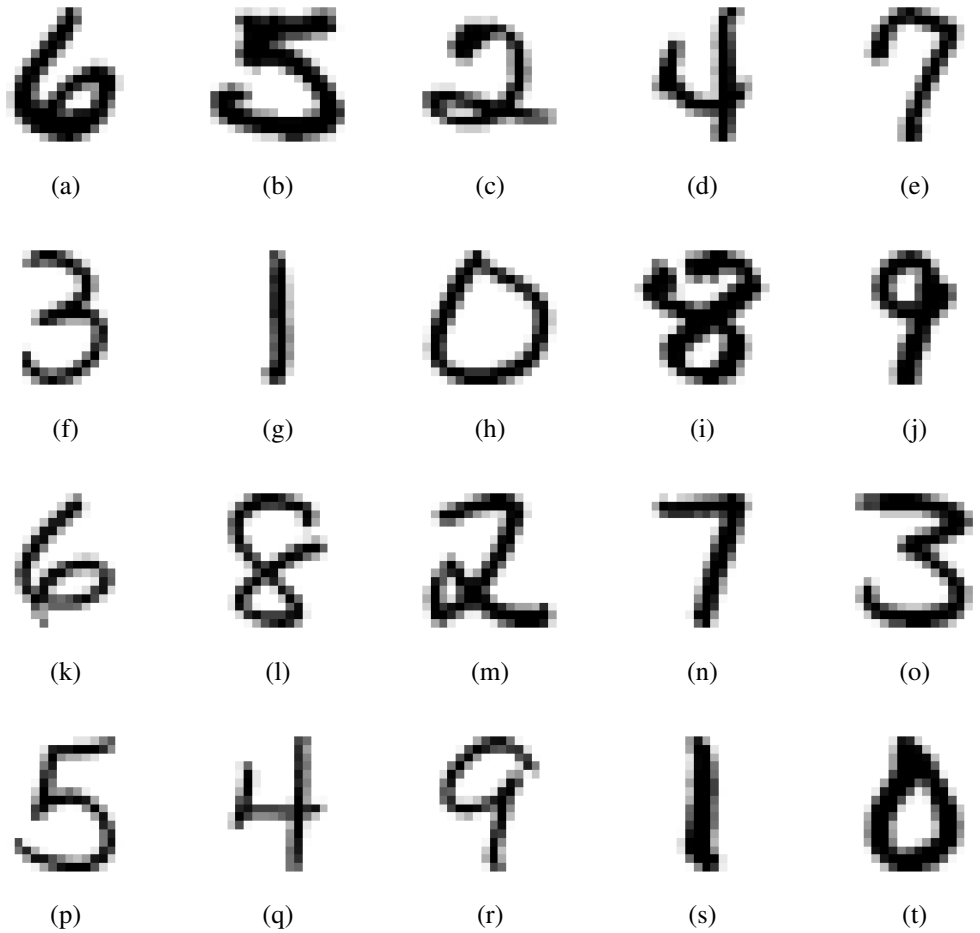
Odabir broja susjeda k je najviše ovisan o podacima, i tu nemamo dobru podlogu za intuiciju i odabir istog. Znamo da ako gledamo velik broj susjeda, onda preslikavamo podatke na linearnu plohu, što ako polazimo od nelinearne strukture nikako nije dobro. S druge strane, premali k rezultira gubitkom globalne strukture. Odabir parametra će stoga u našim algoritmima biti dobiven postupkom pokušaja i pogreške.

4.3.2 Rezultati

Usporedit ćemo rezultate klasifikacije triju ulaganja podataka u nižu dimenziju - koristimo LLE, PCA i difuzijsko preslikavanje. Podaci s kojima radimo su rukom pisane znamenke

0-9, gdje imamo 1707 podataka dimenzije 16×16. Dakle, radimo s 256-dimenzionalnim vektorima. Provest ćemo LLE i PCA ulaganje i difuzijsko preslikavanje na skupu podataka, te ćemo u novim koordinatama niže dimenzije moći raditi brže nego u originalnim, jedino je pitanje kvalitete rezultata i kako ona ovisi o dimenziji prostora u kojega preslikavamo podatke.

Pogledajmo prvo kako izgleda naš skup podataka - prikazali smo neke podatke na slikama. Stvarni podaci su vektorizirani (dakle, umjesto u $R^{16\times16}$, radimo u $\mathbb{R}^{256}$). Također, podaci su zadani u crno-bjelom (eng. *grayscale*) formatu koji poprima vrijednosti u $[-1, 1]$, gdje je -1 crna boja, a 1 bijela.

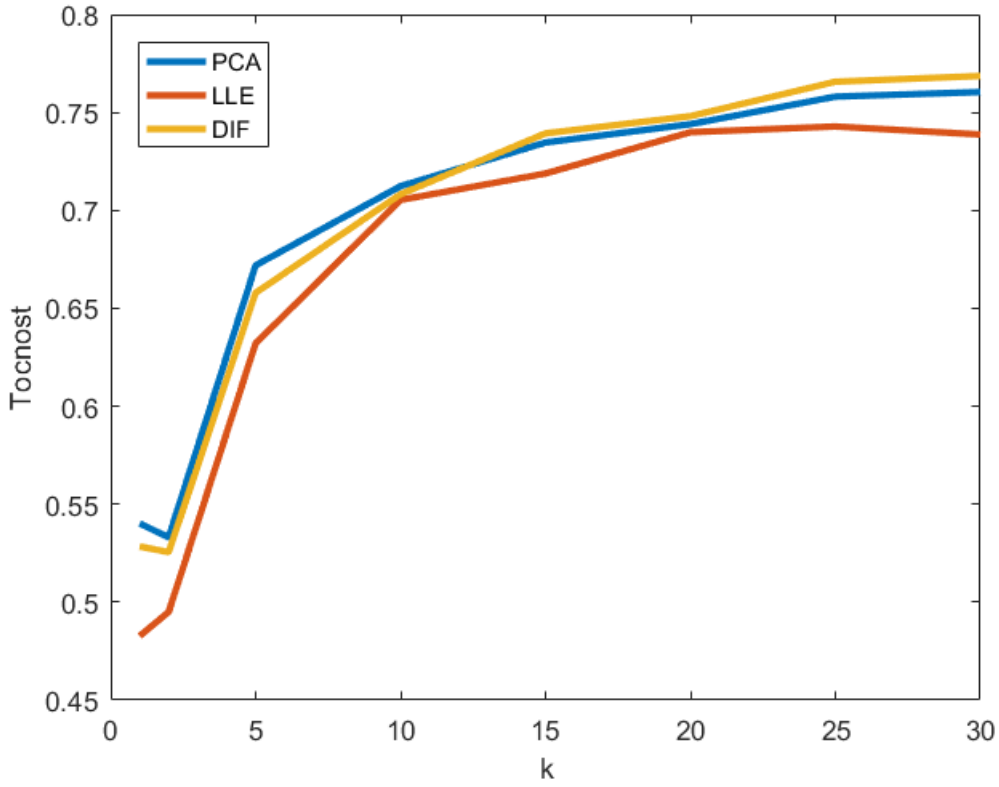


Slika 4.6: Prikaz podataka kojima baratamo - rukom pisane znamenke.

Prvi korak je izračunati 3 tražena preslikavanja - difuzijsko, PCA i LLE. Parametar difuzijskog preslikavanja σ (koji koristimo s Gaussovom jezgrom) ćemo uzeti da je 8 (uzi-

mamo red veličine kakve su prosječne udaljenosti među podacima), a parametar LLE preslikavanja k (broj susjeda koje gledamo) ćemo uzeti 50.

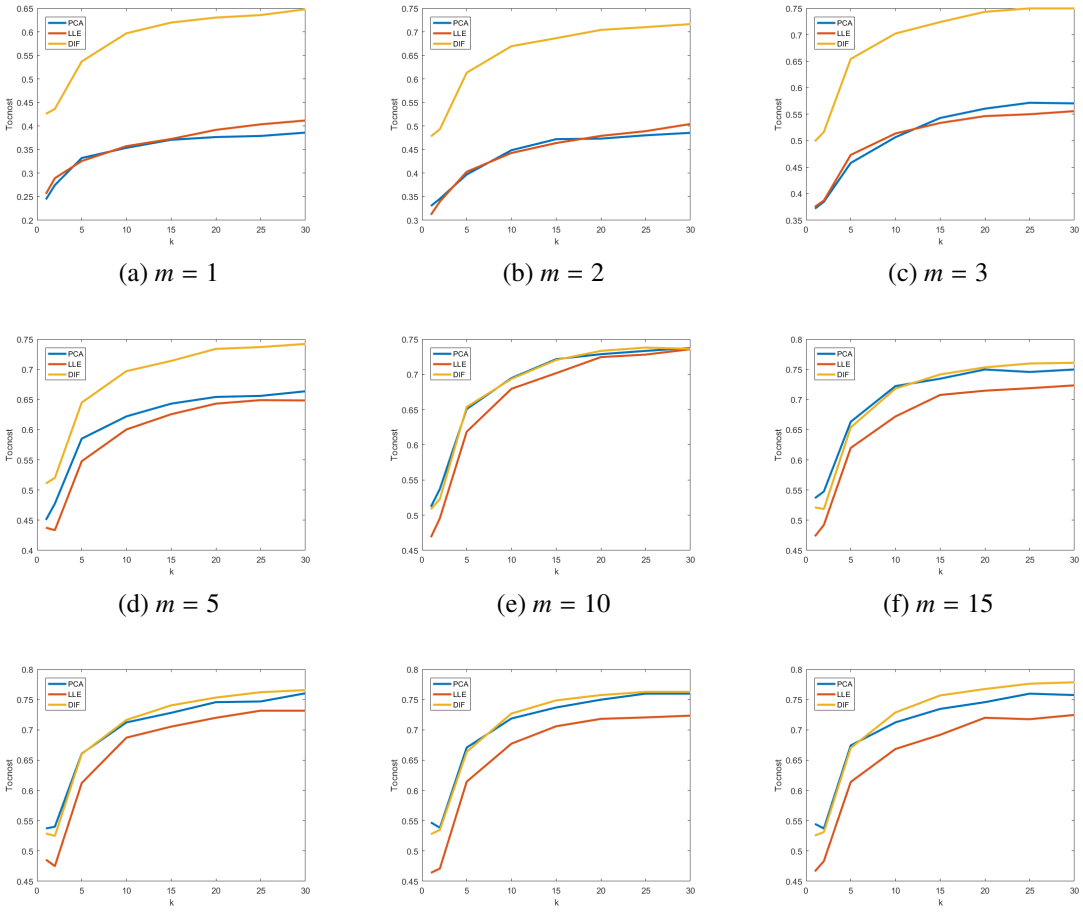
U prvom testu smo koristili automatski odabir parametara r i m za LLE (kako je opisano u prethodnoj sekciji), te smo isti m koristili i za PCA i difuzijsko preslikavanje (kako bi rezultate usporedili na "pošten" način - s istom dimenzijom). Potom smo na preslikanim podacima radili kNN klasifikaciju s različitim k-ovima (pažnja: ovaj k nema veze s parametrom k iz LLE algoritma) i dobili sljedeće rezultate.



Slika 4.7: Usporedba proporcije točno klasificiranih znamenki za PCA, LLE i difuzijsko preslikavanje u ovisnosti o k u kNN algoritmu za klasifikaciju - PCA i difuzijska preslikavanja su bolji od LLE.

Uočavamo da PCA i difuzijsko preslikavanje rade podjednako dobro (PCA je nešto točniji s manjim k, no difuzijsko preslikavanje ga sustiže za veće k), dok je LLE manje točan (zaostaje nekoliko postotaka za sve k)- Ovo su rezultati testa za parametar $v = 0.95$ iz ranije opisanog načina biranja dimenzije m. Uz taj izbor, rezultirajuća dimenzija ulaganja je $m = 23$, dakle, ni 10% početne dimenzije (koja je 256).

S obzirom da je izbor parametra $v = 0.95$ bio arbitraran, usporedit ćemo algoritme i u ovisnosti o dimenziji prostora u koji ulažemo podatke trima preslikavanjima. To nam može dati indikator u kojem slučaju je koji algoritam efikasniji. Primjerice, ako nam je bitno maksimalno uštediti na prostoru i vremenu, koristit ćemo algoritam koji radi bolje s manjim dimenzijama, a ako nam je bitnija točnost, spremni smo uzeti veću dimenziju ulaganja, pa ćemo koristiti algoritam koji bolje radi u toj situaciji. U ovom testu smo opazili sljedeće rezultate:



(a) $m = 1$ (b) $m = 2$ (c) $m = 3$

(d) $m = 5$ (e) $m = 10$ (f) $m = 15$

(g) $m = 20$ (h) $m = 25$ (i) $m = 30$

Slika 4.8: Usporedba triju algoritama u uspješnosti u klasifikaciji u ovisnosti o dimenziji prostora ulaganja m. Difuzijska preslikavanja se pokazuju kao najbolji izbor.

U ovom primjeru uočavamo da su na našem skupo rukom pisanih znamenki algoritmi PCA i difuzijsko preslikavanje efikasniji od LLE algoritma. Također, uočavamo da pri

vrlo malim dimenzijama prostora u koji ulažemo podatke difuzijsko preslikavanje najbolji izbor.

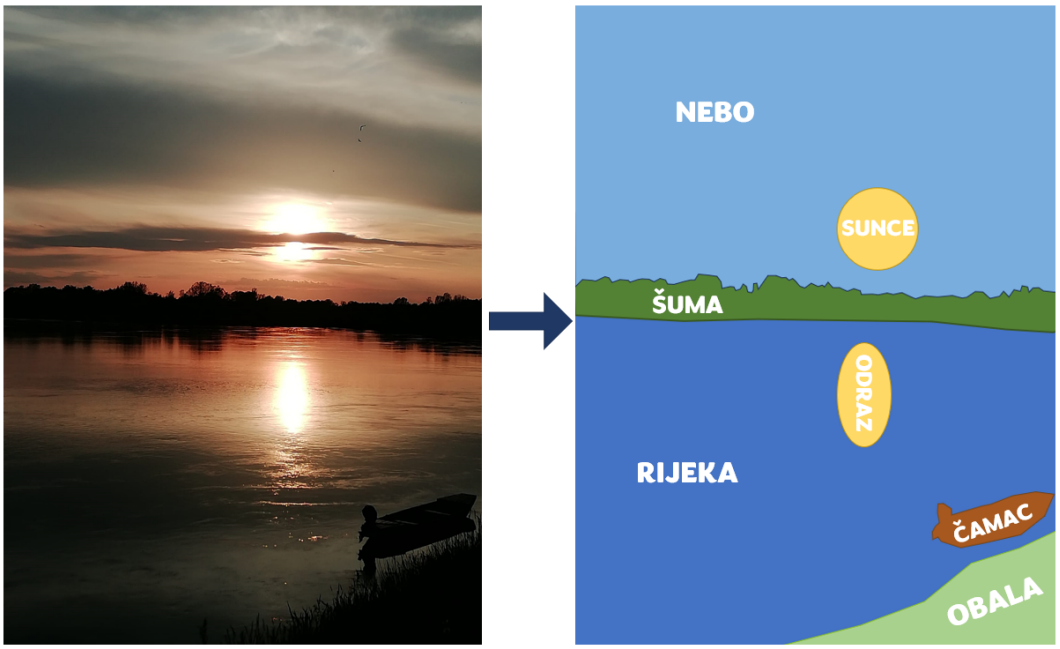
Ipak, treba biti oprezan - iako je difuzijsko preslikavanje najefikasnije u niskim dimenzijama, i dalje je bolji izbor uzeti nešto veću dimenziju prostora preslikavanja kako bismo mogli garantirati veću točnost. Naime, preslikavanje potencijalno kompleksnih struktura u malu dimenziju može rezultirati gubljenjem informacija o originalnoj strukturi te time gubimo veze među podacima.

Poglavlje 5

Segmentacija

5.1 Problem

Problem segmentacije u strojnom vidu i u procesiranju slika se bavi particioniranjem slike kao skupa piksela u segmente, regije ili objekte. Cilj je segmentacije pojednostaviti ili promijeniti reprezentaciju slike tako da dobijemo jasnije značenje, odvojimo smislene objekte ili pojednostavimo analizu.



Slika 5.1: Primjer jednostavne segmentacije slike.

Tipično segmentaciju koristimo za uočavanje objekata i njihovih rubova, ili odvajanje objekata, odnosno regija na slici. Osim toga, jedna od vrlo važnih primjena segmentacije je u medicini - u kompjutoriziranoj tomografiji, za lakše uočavanje nepravilnosti ili atipičnih pojava i radi jasnije vizualiacije finih tkiva. Također se često koristi u 3D rekonstrukciji, gdje se iz stoga slika u kombinaciji s geometrijskim rekonstrukcijskim algoritmima dobiva kompletna 3D struktura.

U primjeni difuzijskih preslikavanja na segmentaciju, naš skup podataka je slika, a njegovi elementi su pikseli na slici. Međutim, to bi moglo uzrokovati problem. Naime, ako gledamo samo jednu sliku relativno male rezolucije, npr. 400×400, imamo $N = 160000$ podataka. Kako difuzijska preslikavanja zahtijevaju izračun matrice jezgre $K = K(x_i, x_j),\ i, j = 1, ..., N$, imali bismo matricu dimenzije 160000×160000 - puno previše za jedno računalo (a u današnjim standardima, rezolucija 400×400 se smatra malom slikom, odnosno slikom niske kvalitete). Dakle, pristup "po kalupu" koji je opisan u poglavlju 3 neće biti izvediv. Stoga je prvi korak smisliti način da uz korištenje manje informacija i dalje izvučemo što je moguće vjerniji prikaz stvarnog preslikavanja - to postižemo Nyströmovom aproksimacijom.

5.2 Nyströmova aproksimacija

Pozabavimo se kratko samom implementacijom difuzijskih preslikavanja u segmentaciji. Radimo s pravokutnim slikama u RGB formatu - dakle, imamo vektore u prostoru $\mathbb{R}^{m \times n \times 3}$ (m i n su dimenzije slike, 3 je za RGB format). Na sliku gledamo kao skup piksela, te ćemo svakom pikselu računati njegovu difuzijsku transformaciju i potom segmentaciju vršiti kao klasteriranje transformiranog skupa podataka.

Prije svega, kratko ćemo opisati motivaciju za difuzijska preslikavanja iz prespektive analize. Polazeći od funkcije jezgre k definirane na skupu podataka X, s ranije spomenutim svojstvima simetrije i pozitvnosti, definiramo mjeru

$$\nu(x) = \int_X k(x, y) d\mu(y).$$

Ovu mjeru možemo gledati kao analogon matrice D iz poglavlja 3 - ona predstavlja "stupanj" svakog vrha, odnosno ukupnu težinu svih bridova koji izlaze iz vrha x, ako na X gledamo kao usmjereni graf.

Idući korak je definirati funkciju $p(x, y) = \frac{k(x,y)}{\nu(x)}$ - ovo je očito analogon matrice $P = D^{-1}K$, koju dobijemo "normiranjem" matrice K, odnosno normiranjem redaka tako da dobijemo stohastičku matricu P.

Konačno, definiramo difuzijski operator kao

$$\mathbf{P}f(x) = \int_X p(x,y)f(y)d\mu(y), \quad f \in \mathcal{L}^1(X).$$

U ovom pristupu, difuzijska preslikavanja dobivamo kao svojstvene funkcije difuzijskog operatora. Možemo uočiti da su difuzijska preslikavanja kako smo ih definirali u poglavlju 3 samo poseban slučaj ovog pristupa - dobijemo ih ako uzmemo da je skup podataka X konačan, a mjera μ na njemu brojeća mjera.

Vratimo se sada na aproksimaciju difuzijske jezgre. Nju ćemo dobiti koristeći tzv. *Nyströmovu aproksimaciju*, koja se originalno koristila za numeričko rješavanje Fredholmove integralne jednadžbe:

$$\int_X a(x,y)\phi_i(x,y)d\mu(y) = \lambda_i\phi_i(x). \tag{5.1}$$

Sličnost s analitičkom definicijom difuzijskih preslikavanja je očita. Ideja Nyströmove metode je aproksimirati jednadžbu 5.1 tako da umjesto integrala računamo uzorački prosjek:

$$\frac{1}{l}\sum_{i=1}^{l} a(x_i,x_j)\phi_j(x_j) \approx \lambda_i\phi_i(x_j).$$

Ova ideja je potakla bogato teoriju i istraživanje, te je detaljnije opisana u [18] i u [**?**]. Mi ćemo se fokusirati samo na slučaj koji smo ranije opisali u poglavlju 3 i navesti kako se Nyströmova metoda koristi za aproksimaciju difuzijskih preslikavanja u našem posebnom slučaju.

Matrica koju želimo dijagonalizirati je $P_s = D^{-\frac{1}{2}}KD^{-\frac{1}{2}}$. Kao što smo i ranije spomenuli, matrica $K \in \mathbb{R}^{n\times n}$ je prevelike dimenzije i stoga ju ne želimo (a vjerojatno i ne možemo) spremati u memoriju i računati s njom kao takvom - isto vrijedi i za matricu P_s. Ipak, svaki element matrice K možemo pojedinačno izračunati poznavanjem elemenata x_i, x_j i funkcije jezgre k.

Elemente matrice P_s tada dobijemo dijeljenjem elemenata matrice K s pripadnim korijenima sume redaka, tj. imamo $P_{s_{ij}} = \frac{K_{ij}}{\sqrt{d_i d_j}}$, gdje je $d = Ke,\ e = (1,...,1)^T \in \mathbb{R}^n$.

Sada ćemo nasumično odabrati manji broj elemenata skupa podataka p i računati podmatricu A čiji su elementi $A_{ij} = P_{s_{\pi(i),\pi(j)}},\ i,j = 1,...,p$, gdje je π preslikavanje koje indeksima od 1 do p pridružuje pripadni indeks podatka u originalnom skupu. Radi lakšeg daljnjeg računa, pretpostavimo da smo odabrali prvih p podataka za uzorak (tj. $\pi(i) = i,\ i = 1,...,p$). Tada imamo sljedeću podjelu matrice P_s na podmatrice:

$$P_s = \begin{bmatrix} A & B^T \\ B & C \end{bmatrix},\ A \in \mathbb{R}^{p\times p},\ B \in \mathbb{R}^{(n-p)\times p},\ C \in \mathbb{R}^{(n-p)\times(n-p)}$$

Ovdje nastupa Nyströmova aproksimacija - ona se zasniva na aproksimiranju cijele matrice P_s preko matrica A i B (s kojima možemo računati, jer matrica koja sadrži većinu veza između podataka je C i nju aproksimiramo). Aproksimacija je sljedeća:

$$P_s \approx \begin{bmatrix} A \\ B \end{bmatrix} A^{\dagger} \begin{bmatrix} A & B^T \end{bmatrix}$$

(S $A^{\dagger}$ smo označili Moore-Penroseov inverz matrice A, to jest $A^{\dagger} = A$).

Jedno potencijalno objašnjenje ove aproksimacije iz matrične teorije je činjenica da, ako je rang matrice P_s jednak rangu podmatrice A i ako je taj rang jednak broju stupaca matrice A (odnosno ako je A regularna), onda u procesu Gaussovih eliminacija dobijemo sljedeću jednakost:

$$P_s = \begin{bmatrix} A & B^T \\ B & C \end{bmatrix} = \begin{bmatrix} I & \mathbf{0} \\ BA^{-1} & I \end{bmatrix} \begin{bmatrix} A & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} I & A^{-1}B \\ 0 & I \end{bmatrix} = \begin{bmatrix} A \\ B \end{bmatrix} A^{-1} \begin{bmatrix} A & B^T \end{bmatrix}$$

Dakle, u ovom slučaju, aproksimacija je egzaktna. Uz dodatnu pretpostavku da je matrica P_s pozitivno definitna, teoretski možemo računati njenu faktorizaciju Choleskog $P_s = FF^T$. Uvedemo li oznake $E = \begin{bmatrix} A \\ B \end{bmatrix}$, imamo da je $P_s = EA^{-1}E^T$, pa je matrica F iz faktorizacije Choleskog jednaka (odnosno u našem slučaju aproksimativno jednaka) $F = EA^{-\frac{1}{2}} \in \mathbb{R}^{N \times p}$ (ovdje gledamo matrični drugi korijen). Ako izračunamo SVD dekompoziciju matrice F, imamo

$$F = U\Sigma V^T,\ U \in \mathbb{R}^{n \times p},\ \Sigma \in \mathbb{R}^{p \times p},\ V \in \mathbb{R}^{n \times p}.$$

Sada je:

$$P_s = FF^T = (U\Sigma V^T)(U\Sigma V^T)^T = U\Sigma^2 U^T.$$

Posebno, lijevi singularni vektori U čine p dominantnih svojstvenih vrijednosti od P_s, dok su same svojstvene vrijednosti tada kvadrati dijagonalnih elemenata u Σ.

Konačno, svojstvene vektore matrice $P = D^{-1}K$ koje trebamo za difuzijsko preslikavanje dobijemo dijeljenjem vektora iz U sumama redaka, tj. $\overline{U} = D^{-\frac{1}{2}}U$. Same difuzijske koordinate tada dobijemo uzimanjem odgovarajućeg broja elemenata u $\overline{U}$ pomnoženih s kvadratima dijagonalnih elemenata u Σ (uz potenciju za koju smo gledali difuzijsko preslikavanje).

5.3 Implementacija i rezultati

Algoritam za segmentaciju će u suštini biti vrlo jednostavan - jednom kad primijenimo difuzijsko preslikavanje na sliku, dobijemo točke s novim koordinatama na koje tada primijenimo neki od algoritama za klasteriranje. Mi ćemo koristiti najjednostavniji algoritam

- *k-means*. Alternativna opcija je koristiti i spektralno klasteriranje, ali u tom slučaju nailazimo na isti problem kao i kod računanja difuzijskih preslikavanja: podataka je puno, a algoritam zahtijeva računanje udaljenosti (ili mjere afiniteta) za svaki par podataka, što nije izvedivo.

Još jedan bitan detalj u implementaciji dobivamo iz sljedećeg - kako RGB slika od informacije za piksel sadržava samo njegovu boju, rezultat standardnog difuzijskog preslikavanja je da će pikseli slične (a pogotovo iste) boje biti vrlo blizu (jer im je euklidska udaljenost jako mala, pa to vrijedi i za difuzijsku). Stoga ubacujemo dvije nove koordinate - položaj piksela na slici, kako bismo udaljene objekte smatrali različitima. Dakle, ako je piksel x_{ij} na slici na položaju (i, j), onda su njegove koordinate (x_r, x_g, x_b, i, j) (gdje su x_r, x_g, x_b pripadne RGB vrijednosti).

Uz to ćemo u algoritam ćemo ubaciti i parametar težine s kojim množimo položaj piksela. Kako su RGB vrijednosti piksela (u MATLAB-u) vrijednosti između 0 i 1, moramo skalirati i sami položaj, u protivnom on preuzima svu težinu i gubimo informaciju boje. Dodatno, taj nam parametar dopušta da sami biramo je li nam bitnija boja ili položaj piksela. Stoga konačne koordinate svakog piksela (one koje šaljemo u difuzijsko preslikavanje) imaju vrijednosti $(x_r, x_g, x_b, x_i, x_j)$, gdje, ako je dimenzija slike $m \times n$, onda su $x_i = c\frac{i}{m}$, $x_j = c\frac{j}{n}$ (c je parametar težine u $[0, 1]$, njega biramo ručno).

Proveli smo segmentaciju pomoću difuzijskih preslikavanja na nekoliko slika i dobili sljedeće rezultate:

(a) Polazna slika (b) 5 klasa (c) 6 klasa

Slika 5.2: Segmentacija slike u različite brojeve segmenata (parametri su $c = 0.02, \sigma = 1.5\|x - y\|_2^2$, gdje su x i y prvi i središnji piksel, $r = 30, k = 5$).

(a) Polazna slika (b) 3 klase (c) 4 klase

Slika 5.3: Segmentacija slike u različite brojeve segmenata (parametri su $c = 0.04, \sigma = 1.5\|x - y\|_2^2$, gdje su x i y prvi i središnji piksel, $r = 30, k = 5$).

(a) Polazna slika (b) 3 klase (c) 4 klase

Slika 5.4: Segmentacija slike u različite brojeve segmenata (parametri su $c = 0.5, \sigma = 1.5\|x - y\|_2^2$, gdje su x i y prvi i središnji piksel, $r = 30, k = 5$).

Vidimo da algoritam, iako vrlo jednostavan, daje zadovoljavajuće rezultate. Ipak, mogu se naći i primjeri gdje rezultati nisu ispravni:

(a) Polazna slika (b) 3 klase (c) 4 klase

Slika 5.5: Pokušaj segmentacije s različitim izborima parametara - ne dobivamo zadovoljavajuće rezultate.

Potencijalni problem koji možemo uočiti je taj da ako je isti objekt ili segment na slici u nekoliko različith boja, algoritam ima problema s odvajanjem tog objekta od ostatka slike, već isti objekt razdvoji na dijelove različitih boja. Ovo je donekle i za očekivati jer se algoritam zasniva na udaljenosti piksela koja je rezultat različitosti u boji i položaju na slici - ako je isti objekt dovoljno velik i mijenja boju, teško ga je odrediti.

Zaključak koji bismo trebali donijeti iz ovog algoritma, ali i onog za klasifikaciju je taj da su difuzijska preslikavanja zaista moćan i koristan alat, ali su i dalje samo alat - nisu magično rješenje cijelog problema. Najbolja primjena im je upravo kao jedan korak u cijelom algoritmu namijenjenom za određenu svrhu, kao što primjerice možemo vidjeti u [10] i [17], gdje su difuzijska preslikavanja koriste samo kao djelić u velikom algoritmu kojim na kraju dobivamo vrhunske rezultate.

Bibliografija

[1] G. Allaire, K. Trabelsi i S.M. Kaber, *Numerical Linear Algebra*, Texts in Applied Mathematics, Springer New York, 2008, ISBN 9780387689180, `https://books.google.hr/books?id=PlUWqQzY-4EC`.

[2] Nachman Aronszajn, *Theory of reproducing kernels*, Transactions of the American mathematical society **68** (1950), br. 3, 337–404.

[3] Salomon Bochner, *Hilbert distances and positive definite functions*, Annals of Mathematics (1941), 647–656.

[4] Richard A Brualdi, *Some applications of doubly stochastic matrices*, Linear Algebra and its Applications **107** (1988), 77–100.

[5] Kurt Bryan i Tanya Leise, *The $25,000,000,000 eigenvector: The linear algebra behind Google*, SIAM review **48** (2006), br. 3, 569–581.

[6] Ronald R Coifman i Stéphane Lafon, *Diffusion maps*, Applied and computational harmonic analysis **21** (2006), br. 1, 5–30.

[7] Dick De Ridder i Robert PW Duin, *Locally linear embedding for classification*, Pattern Recognition Group, Dept. of Imaging Science & Technology, Delft University of Technology, Delft, The Netherlands, Tech. Rep. PH-2002-01 (2002), 1–12.

[8] Petros Drineas, Michael W Mahoney i Nello Cristianini, *On the Nyström Method for Approximating a Gram Matrix for Improved Kernel-Based Learning.*, journal of machine learning research **6** (2005), br. 12.

[9] Zlatko Drmač, Vjeran Hari, Miljenko Marušić, Mladen Rogina, Sanja Singer i Saša Singer, *Numerička analiza*, PMF-Matematički odjel, Sveučilište u Zagrebu (2003).

[10] Zeev Farbman, Raanan Fattal i Dani Lischinski, *Diffusion maps for edge-aware image editing*, ACM Transactions on Graphics (TOG) **29** (2010), br. 6, 1–10.

[11] Harold Hotelling, *Analysis of a complex of statistical variables into principal components.*, Journal of educational psychology **24** (1933), br. 6, 417.

[12] Stéphane S Lafon, *Diffusion maps and geometric harmonics*, Yale University, 2004.

[13] Sam T Roweis i Lawrence K Saul, *Nonlinear dimensionality reduction by locally linear embedding*, science **290** (2000), br. 5500, 2323–2326.

[14] Saul A Teukolsky, Brian P Flannery, WH Press i WT Vetterling, *Numerical recipes in C*, SMR **693** (1992), br. 1, 59–70.

[15] Behrouz Touri i Angelia Nedić, *Product of random stochastic matrices*, IEEE Transactions on Automatic Control **59** (2013), br. 2, 437–448.

[16] Z. Vondraček, *Markovljevi lanci - skripta*, (2012), `https://web.math.pmf.unizg.hr/~vondra/ml12-predavanja.html`.

[17] Xun Wang, Jianqiu Jin i Bailin Yang, *Diffusion map based interactive image segmentation*, Multimedia Tools and Applications **76** (2017), 17497–17509.

[18] Christopher Williams i Matthias Seeger, *Using the Nyström method to speed up kernel machines*, Advances in neural information processing systems **13** (2000).

[19] H. Šikić, *Mjera i integral - skripta*, (2011), `https://web.math.pmf.unizg.hr/nastava/mii/files/mii-predavanja-sikic.pdf`.

Sažetak

Difuzijska preslikavanja su metoda transformacije koordinata s ciljem učenja nelinearnih ploha u visokodimenzionalnim prostorima. Osnovna ideja iza ove metode je na novi način definirati udaljenosti među podacima, onu koja će u svojoj definiciji koristiti sve podatke, a ne samo onaj par između kojih mjerimo udaljenost, kako bi globalnu strukturu imali sadržanu u lokalnoj funkciji. S tom mjerom udaljenosti, želimo koordinate transformirati tako da standardna euklidska udaljenost u novim koordinatama odgovara novo definiranoj mjeri udaljenosti.

U ovom radu dajemo vjerojatnosnu interpretaciju difuzijskih preslikavanja preko Markovljevih lanaca, s fokusom na slučajne šetnje na grafu. Za dane podatke, koristeći difuzijsku jezgru, definiramo matricu prijelaza koja odgovara slučajnoj šetnji na grafu, te pri računu transformacije koordinata u difuzijske, računamo svojstvene vrijednosti i vektore pripadne matrice. Taj se problem pokazuje prezahtjevnim za egzaktan račun, stoga pribjegavamo numeričkim metodama za rješavanje svojstvenog problema, s fokusom na simetrične matrice, kako bismo u konačnici dobili algoritam za računanje difuzijskih preslikavanja koji je i brz i točan.

Jednom kada imamo potreban algoritam, primjenjujemo ga u klasifikaciji i segmentaciji. Za klasifikaciju imamo nekoliko kandidata s kojima možemo usporediti našu metodu, te biramo vrlo popularan PCA i metodu koja je suštinski vrlo povezana s difuzijskim preslikavanjima - lokalno linearna ulaganja. U primjeni za segmentaciju, javlja nam se problem s količinom podataka, koji u ovom slučaju uspješno rješavamo koristeći Nyströmovu aproksimaciju, te za obje primjene dobivamo kvalitetne rezultate.

Summary

Diffusion maps are a coordinate transformation method with the goal of learning nonlinear manifolds in a high dimensional ambient space. The main idea behind this method is to redefine the distance between our data points, using a measure which will incorporate all available data points instead of focusing simply on the two points of interest, so that the global structure can be reflected locally. With this new measure of distance, we transform the coordinates so that the standard euclidean distance equates to newly defined distance measure.

In this paper we present the probabilistic interpretation of diffusion maps based on Markov chains, specifically, random walks on a graph. For a given dataset, by using a diffusion kernel, we define a transition matrix for a random walk on a graph, for which we calculate corresponding eigenvalues and eigenvectors needed to transform starting coordinates to diffusion coordinates. The eigenproblem proves to be too challenging for exact calculation, which compels us to use numerical methods for eigenvalues, with symmetrical matrices as our main focus, which in the end gives us an efficient and precise algorithm for calculating diffusion maps.

With our algorithm in hand, we apply it to problems of classification and segmentation. For classification we compare our algorithm to a very popular and widely used algorithm known as PCA, and we also compare it to a very theoretically similar method to diffusion maps known as Locally Linear Embedding method. When it comes to segmentation, the problem of data size arises, with which we deal with using Nyström approximation, so that finally we obtain an algorithm which provides decent results.

Životopis

Rođen sam 3.3.1998. u Novoj Gradiški. Odrastao sam u Davoru, selu blizu Nove Gradiške, gdje sam ujedno i pohađao osnovnu školu Matija Antun Relković. Nakon osnovne škole, upisujem Gimnaziju Nova Gradiška, koju završavam 2016. godine. Iste godine sam upisao nastavnički studij Matematike i fizike na Prirodoslovno-matematičkom fakultetu u Zagrebu, gdje sam uz veliku pomoć profesora, predavača i dragih kolega shvatio svoju ljubav prema matematici koja me motivirala da joj se posvetim u potpunosti te se 2018. prebacujem na preddiplomski studij Matematike, koji završavam 2021. godine te iste godine upisujem diplomski studij Matematička statistika, kojeg ovim radom završavam. Tijekom studija sam držao i demonstrature iz kolegija Analitička geometrija, Osnove matematičke analize, Osnove fizike 1, 2, 3 i 4, Elementarna matematika 1 i 2, Integrali funkcija više varijabli i Mjera i integral, a na kraju oba studijska programa sam nagrađen nagradom Matematičkog odsjeka za najuspješnije studente.