



УНИВЕРЗИТЕТ У НИШУ
ПРИРОДНО-МАТЕМАТИЧКИ ФАКУЛТЕТ
ДЕПАРТМАН ЗА РАЧУНАРСКЕ НАУКЕ



Александар Б. Трокицић

**АЛГОРИТМИ ЗА БРЗО АПРОКСИМАТИВНО
СПЕКТРАЛНО УЧЕЊЕ**

ДОКТОРСКА ДИСЕРТАЦИЈА

Ниш, 2021.



UNIVERSITY OF NIŠ
FACULTY OF SCIENCES AND MATHEMATICS
DEPARTMENT OF COMPUTER SCIENCE



Aleksandar B. Trokicić

**ALGORITHMS FOR FAST APPROXIMATE
SPECTRAL LEARNING**

DOCTORAL DISSERTATION

Niš, 2021.

Sadržaj

1	Uvod	15
1.1	Spektralni algoritmi i algoritmi za aproksimaciju matrice	16
1.1.1	Metode aproksimacije	18
1.1.2	Ulazna matrica je proizvoljna kvadratna matrica	20
1.1.3	Ulazna matrica je simetrična i pozitivno semidefinitna	21
1.2	Klasterovanje	22
1.2.1	Algoritam k -srednjih vrednosti	24
1.2.2	Grafovski algoritam klasterovanja: spektralno klasterovanje	26
1.2.3	Graf sličnosti	28
1.2.4	Laplasova matrica grafa	29
1.2.5	Nenormalizovana Laplasova matrica	30
1.2.6	Normalizovana Laplasova matrica	31
1.2.7	Intuitivni prikaz spektralnog klasterovanja: relaksacija problema nalaženja minimalnog reza datog grafa	33
1.3	Kernel regresija	38
2	Klasterovanje sa ograničenjima	41
2.1	Uvod	41
2.2	Spektralno klasterovanje	44
2.2.1	Grasmanova mnogostrukost	46
2.2.2	Spektralno klasterovanje višeslojnog grafa	47
2.3	Spektralno klasterovanje sa ograničenjima	49
2.4	Novi algoritam spektralnog klasterovanja sa ograničenjima linearne vre- menske i memorijske složenosti	52
2.5	Empirijska evaluacija algoritama spektralnog klasterovanja sa ograničenjima	58
2.5.1	Polu nadgledano testiranje	59
2.5.2	Nedosledna ograničenja	61
2.5.3	Skup ograničenja sa 30% šuma	62
2.5.4	Meka ograničenja	62
2.5.5	Meka ograničenja na granama	63
2.5.6	Veliki ulazni skup	63
3	Brzo spektralno klasterovanje pomoću samoučenja	76
3.1	Uvod i prethodni rezultati	76
3.2	Algoritam	78

3.3	Empirijska evaluacija	80
4	O očekivanoj grešci aproksimacije kernel regresije metodom Nistroma i njenom ekstenzijom	85
4.1	Uvod	85
4.2	Brza kernel regresija	88
4.3	Nistromov vektor karakteristika	88
4.4	Aproksimacija spektralne dekompozicije	89
4.5	Vektori slučajnih karakteristika kreirani ekstenzijom Nistromove metode .	91
4.6	Analiza približne greške	93
4.7	Eksperimentalni rezultati	103
5	Zaključak	108

Spisak slika

1.1	Slika podataka za klasterovanje.	23
1.2	Slika u kojima su podaci sa slike 1.1 klasterovani u tri klastera.	23
1.3	Slika podataka gde klasterizacija nije jednostavna zato što klasteri imaju različite oblike.	24
1.4	Slike podataka sa različitim skaliranjem koordinata. Oznake podataka i na jednoj i na drugoj slici su x_1 , x_2 , x_3 i x_4 označeno odozdo na gore. . .	25
1.5	Slike podataka koji se mogu klasterovati algoritmom spektralnog klasterovanja a ne algoritmom k -srednjih vrednosti.	38
2.1	Poređenje performansi različitih algoritama spektralnog klasterovanja sa ograničenjima na skupovima podataka iz stvarnog sveta za različit broj poznatih labela koji je dat na x -osi.	67
2.2	Poređenje performansi različitih algoritama spektralnog klasterovanja sa ograničenjima na skupovima podataka iz stvarnog sveta za različit broj poznatih labela koji je dat na x -osi.	68
2.3	Poređenje performansi različitih algoritama spektralnog klasterovanja sa ograničenjima na skupovima podataka iz stvarnog sveta sa nedoslednim (kontradiktornim) ograničenjima za različit broj poznatih labela koji je dat na x -osi.	69
2.4	Poređenje performansi različitih algoritama spektralnog klasterovanja sa ograničenjima na skupovima podataka iz stvarnog sveta, gde skup ograničenja sadrži šum, za različit broj poznatih labela koji je dat na x -osi. . .	70
2.5	Poređenje performansi različitih algoritama spektralnog klasterovanja sa ograničenjima na skupovima podataka iz stvarnog sveta, gde skup ograničenja sadrži meka ograničenja za različit broj poznatih labela koji je dat na x -osi.	71
2.6	Poređenje performansi različitih algoritama spektralnog klasterovanja sa ograničenjima na skupovima podataka iz stvarnog sveta, gde skup ograničenja sadrži meka ograničenja po ivicama za različit broj parova ograničenja koji je dat na x -osi.	72
2.7	Poređenje performansi i vremena izvršenja različitih algoritama spektralnog klasterovanja sa ograničenjima na podskupu skupa podataka covtype sa tri različita tipa ograničenja (nedosledna, meka i meka po granama). Vreme rada je predstavljeno sa logaritamskom skalom. Broj poznatih labela je dat na x -osi.	73

-
- 2.8 Poređenje performansi i vremena izvršenja različitih algoritama spektralnog klasterovanja sa ograničenjima na podskupu skupa podataka covtype sa tri različita tipa ograničenja (nedosledna, meka i meka po granama). Vreme rada je predstavljeno sa logaritamskom skalom. Broj poznatih labela je dat na x -osi. 74
- 2.9 Poređenje performansi i vremena izvršenja različitih algoritama spektralnog klasterovanja sa ograničenjima na podskupu skupa podataka covtype sa tri različita tipa ograničenja (nedosledna, meka i meka po granama). Vreme rada je predstavljeno sa logaritamskom skalom. Broj poznatih labela je dat na x -osi. 75

Spisak tabela

2.1	Skupovi podataka na kojima ćemo evaluirati algoritme klasterovanja sa ograničenjima. Prikazujemo naziv skupa, broj primera u skupu i koliko atributa ima svaki skup.	58
2.2	Poređenje performansi različitih algoritama spektralnog klasterovanja sa ograničenjima na sledećim skupovima podataka iz stvarnog sveta: <i>glass</i> , <i>seeds</i> i <i>ionosphere</i> . Prikazujemo prosečni Randov indeks i njegovu standardnu devijaciju	60
2.3	Poređenje performansi različitih algoritama spektralnog klasterovanja sa ograničenjima na sledećim skupovima podataka iz stvarnog sveta: <i>glass</i> , <i>seeds</i> i <i>ionosphere</i> . Prikazujemo prosečnu normalizovanu združenu količinu informacija i standardnu devijaciju.	61
2.4	Performanse algoritma FAST-CSP-ML na velikim skupovima podataka sa dve različite vrste ograničenja (meko bez šuma i meko sa 30 % šuma). Izveštavamo o prosečnom randovom indeksu i prosečnom vremenu izvršenja programa u sekundama izračunatom iz rezultata 20 izvedenih testova.	64
2.5	Performanse algoritma FAST-CSP-ML na velikim skupovima podataka sa dve različite vrste ograničenja (meko bez šuma i meko sa 30 % šuma). Izveštavamo o prosečnom randovom indeksu i prosečnom vremenu izvršenja programa u sekundama izračunatom iz rezultata 20 izvedenih testova.	66
3.1	Skupovi podataka na kojima ćemo evaluirati algoritme spektralnog klasterovanja sa samoučenjem. Prikazujemo naziv skupa, broj primera u skupu i koliko atributa ima svaki skup.	81
3.2	Poređenje performansi algoritama csp-ml , fsl-sc i sl na sledećim skupovima podataka iz stvarnog sveta: Postures, Orange i Covtype. Prikazujemo prosečan Randov indeks i njegovu varijansu. Ograničenja predstavljaju procenat ulaznih vektora za koje znamo oznaku.	83
4.1	Skupovi podataka na kojima ćemo evaluirati algoritme kernel regresije sa slučajnim karakteristikima. Prikazujemo naziv skupa, broj primera u skupu i koliko atributa ima svaki skup.	103

4.2	Poređenje performansi algoritama RNF i NF na skupovima podataka iz stvarnog sveta. Pokazujemo srednje vreme izvršenja u sekundama. Broj slučajnih karakteristika je 10, a broj izabranih kolona je 50.	106
4.3	Poređenje performansi algoritama RNF i NF na skupovima podataka iz stvarnog sveta. Pokazujemo srednju grešku predikcije i njenu standardnu devijaciju. Broj slučajnih karakteristika je 10, a broj izabranih kolona je 50.	107

Spisak algoritama

1	Algoritam k -srednjih vrednosti.	26
2	Algoritam spektralne bisekcije grafa.	36
3	Spektralno klasterovanje.	37
4	Spektralno klasterovanje.	45
5	Spektralno klasterovanje višeslojnog grafa.	47
6	Algoritam spektralnog klasterovanja sa ograničenjima kao spektralno klasterovanje višeslojnog grafa.	52
7	Brzi algoritam spektralnog klasterovanja sa ograničenjima.	57
8	Brzo spektralno klasterovanje koje se samouči.	81
9	Kernel regresija: NF algoritam.	89
10	Kernel regresija: RNF algorithm.	92

Glava 1

Uvod

U ovoj tezi ćemo razvijati i analizirati algoritme mašinskog učenja. Algoritmi mašinskog učenja analiziraju realne skupove podataka i traže obrasce u njima. Danas imamo masivne skupove podataka jer je razvoj tehnologije doveo do toga da imamo pristup velikim kolekcijama podataka sakupljenim iz heterogenih izvora (internet, pametni uređaji, saobraćaj itd.). Tipovi podataka su takođe raznoliki (tekst, zvuk, itd). Zbog veličine samih skupova javlja se potreba za algoritmima koji mogu brzo da izvuku informacije iz takvih skupova i dovedu do rezultata. Mi ćemo ovde istraživati specijalnu klasu algoritama učenja koji se zovu spektralni metodi (poglavlje 1.1), zato što analiziraju spektar nekih matrica izračunatih iz ulaznih primera. Te metode ćemo primenjivati na problemu klasterovanja ili grupisanja podataka (poglavlje 1.2) i problemu kernel regresije (poglavlje 1.3).

Struktura teze je sledeća. U prvoj glavi je dat uvod u kome su opisani osnovni pojmovi, postojeće metode iz literature i definisan je problem koji se rešava. U drugoj glavi su razvijeni algoritmi za spektralno klasterovanje sa ograničenjima: jedan sa kvadratnom složenosti i jedan sa linearnom složenosti, koji je aproksimirana verzija kvadratnog algoritma. Data je teorijska analiza aproksimacije i empirijska analiza oba algoritma. U trećoj glavi je definisan algoritam spektralnog klasterovanja sa samoučenjem sa linearnom složenosti koji je empirijski analiziran na podacima iz stvarnog sveta. U četvrtoj glavi je analizirano ubrzanje kernel regresije ekstenzijom Nistromove metode. Teorijski je analizirana greška regresije i empirijski je pokazano da je ekstenzija Nistromove metode bolja od same Nistromove metode na problemu kernel regresije. Na kraju u poslednjoj glavi je dat zaključak. Rezultati predstavljeni u glavama (2 i 4) su objavljeni u radovima [Trokić and Todorović, 2020, 2019b,a].

Želeo bih da se iskreno zahvalim Prof. Dr. Branimiru Todoroviću na velikoj pomoći, idejama, motivaciji, savetima i strpljenju za vreme doktorskih studija, kao i za to što me je upoznao sa disciplinom mašinskog učenja. Takođe bih izrazio zahvalnost i svim Profesorima sa Prirodno-matematičkog fakulteta na svemu čemu su me naučili i na svim diskusijama koje smo imali za vreme mojih studija. Na kraju bih se najtoplije zahvalio svojoj porodici na nesebičnoj podršci za vreme mog školovanja

1.1 Spektralni algoritmi i algoritmi za aproksimaciju matrice

U ovom poglavlju ćemo opisati spektralne metode učenja i aproksimativne metode koje se koriste za njihovo ubrzanje. Pratimo sledeću literaturu i opisujemo neke algoritme, metode i zaključke iz [Halko et al., 2011, Chen et al., 2020, Mahoney, 2011, Kishore Kumar and Schneider, 2017, Golub and Van Loan, 1996].

Količina podataka koja je dostupna danas je enormna i dolazi iz raznolikih izvora, pa se javlja potreba za algoritmima koji imaju malu memorijsku i vremensku složenost. Ne samo što je količina podataka velika, ona se i dalje povećava i to sve brže i brže. Razlog za to je veliki tehnološki napredak koji se dogodio. Izvori tih podataka su internet, rasprostranjenost pametnih telefona i ostalih pametnih uređaja, socijalne mreže, baze genetskih podataka dostupne zbog razvoja u tehnologije analize samih genetskih materijala, zdravstveni kartoni koji su u mnogome digitalizovani, finansijska tržišta, razne prodavnice koja su takođe digitalizovane sa povećanjem prodaje preko interneta, baze podataka sakupljenih iz saobraćaja, razni senzori itd. Vrste tih podataka mogu biti raznovrsne kao na primer: tekst, slika, video, biološke sekvence, razne mreže (biološke, socijalne, saobraćajne, ...), zvuk, itd. Danas je dosta olakšano skladištenje podataka i njihov prenos. Pored toga povećanje kompjuterske brzine omogućava obradu velike količine podataka. I pored velike dostupnosti podataka njihov kvalitet je često nedovoljno veliki. Na primer skupovi podataka često nisu kompletni (fale im neki atributi ili njihove vrednosti), podaci često imaju šum itd. Vrlo često su dostupni podaci u formi matrice i to izuzetno velike matrice koja ponekad ne može ni da stane u memoriju.

U skorije vreme spektralni algoritmi su se pokazali efikasni u rešavanju ovih problema [Chen et al., 2020]. Oni nekada menjaju klasične metode mašinskog učenja, a nekada mogu biti korišćeni kao prvi korak u nekom standardnom algoritmu (na primer preslikavanjem podataka u drugi prostor kao što se koristi u spektralnom klasterovanju). Ulazni parametar spektralnih metoda je ili neka ulazna matrica ili matrica na formirana iz ulaznih podataka. Osnova samih algoritama je analiza spektra tih matrica. Prema tome u ovim algoritmima analiziramo sopstvene vrednosti i sopstvene vektore matrica ili singularne vrednosti i singularne vektore matrica. Našli su vrlo široku lepezu primena:

- detekcija zajednica u socijalnim mrežama [Abbe, 2017, Newman, 2006],
- analiza slika [Chen and Candès, 2018, Shi and Malik, 2000],
- klasterovanje na grafu [Von Luxburg, 2007, Ng et al., 2002],
- redukcija dimenzija podataka [Belkin and Niyogi, 2003],
- aproksimacija matricom niskog ranga [Halko et al., 2011, Bach, 2013, Achlioptas and McSherry, 2007],
- estimacija kovarijanske matrice [Fan et al., 2018],
- analiza finansijskih podataka [Fan et al., 2020],

1.1. SPEKTRALNI ALGORITMI I ALGORITMI ZA APROKSIMACIJU MATRICE

- analiza bioloških podataka [Kalecky and Cho, 2018],
- rangiranje [Chen et al., 2019],

Algoritmi koje ćemo analizirati uglavnom analiziraju velike matrice, jer su njihove dimenzije jednake broju ulaznih primera. Znamo da spektralne metode, pogotovu kada su u pitanju razne dekompozicije matrica, imaju kvadratnu ili kubnu složenost po broju ulaznih primera. Zbog tog ćemo mi koristiti razne metode aproksimacije da bismo razvili algoritme linearne složenosti. Koristićemo metode aproksimacije matrice dekompozicija da bi smo dobili aproksimirane sopstvene vektore, a samom tom aproksimacijom ćemo dobiti i aproksimaciju matrice malog ranga. U nekim slučajevima matrice će biti toliko velike da neće moći da stanu u memoriju, ali će i u tim slučajevima ove metode pomagati. Recimo metode koje su bazirane na Nistromovoj metodi ne zahtevaju na ulazu kompletnu matricu da bi izračunali aproksimativnu dekompoziciju.

Metode aproksimacije koje ćemo istraživati su bazirane na slučajnom izboru neke male matrice nezavisne od ulaza ili na slučajnom izboru matrice koja je podmatrica date matrice, i ta matrica se koristi za računanje aproksimacije (*eng. randomized methods*). Prednosti ovih metoda u poređenju sa klasičnim determinističkim metodama su sledeće (po [Halko et al., 2011]):

- imaju malu (često proizvoljnu) vremensku složenost,
- vrlo su jednostavni i efikasni,
- dešava se u praksi da su robustni, imaju dobre performanse na podacima sa velikim šumom.

Kako brzina algoritma često zavisi od proizvoljnog broja (hiperparametar), korisnik može da bira da li hoće bržu aproksimaciju ili teorijski tačniju (približniju) aproksimaciju. Prikazaćemo kako ove aproksimacije mogu da se kombinuju sa klasičnim metodama da bi dobili vrlo efikasne algoritme, koji rešavaju probleme spektralnog klasterovanja i kernel regresije.

Uvedimo sada neke osnovne pojmove o matricama.

Definicija 1.1.1. Data je matrica $A \in \mathbf{R}^{n \times n}$. Sopstvena vrednost ove matrice je realan broj λ ako postoji netrivialno rešenje matrice jednačine

$$Ax = \lambda x$$

gde se vektor $x \in \mathbf{R}^n$ zove sopstveni vektor koji odgovara sopstvenoj vrednosti λ .

Teorema 1.1.2. Data je simetrična matrica $A \in \mathbf{R}^{n \times n}$. Ova matrica ima samo realne sopstvene vrednosti:

$$\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n.$$

Matrica A se može napisati u sledećem obliku:

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

gde je Λ dijagonalna matrica sa sopstvenim vrednostima na dijagonali, a U je matrica čije su kolone ortonormirani sopstveni vektori matrice A i za nju važi $U^T U = I$.

1.1. SPEKTRALNI ALGORITMI I ALGORITMI ZA APROKSIMACIJU MATRICE

U slučaju da matrica nije simetrična ili nije kvadratna može da se faktoriše na sledeći način:

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

gde je Σ dijagonalna matrica sa elementima na dijagonali koji se zovu singularne vrednosti, a za matrice U i V važi $UU^T = I$ i $VV^T = I$ i njihove kolone se zovu levi singularni vektori i desni singularni vektori, respektivno. Primetimo da su kvadrati singularnih vrednosti sopstvene vrednosti matrice AA^T i $A^T A$ i da su levi (desni) singularni vektori sopstveni vektori matrice AA^T ($A^T A$).

Definicija 1.1.3. Data je simetrična matrica $A \in \mathbf{R}^{n \times n}$. Kažemo da je ona pozitivno semidefinitna ako za svaki netrivialni vektor $x \in \mathbf{R}^n$ važi:

$$x^T A x \geq 0.$$

Teorema 1.1.4. Data je simetrična i pozitivno semidefinitna matrica $A \in \mathbf{R}^{n \times n}$. Ova matrica ima samo realne nenegativne sopstvene vrednosti.

Teorema 1.1.5 (Rejljev odnos). Data je simetrična matrica $A \in \mathbf{R}^{n \times n}$, sa sledećih $k \leq n$ realnih sopstvenih vrednosti

$$\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_k.$$

Neka su vektori x_1, x_2, \dots, x_k sopstveni vektori koji odgovaraju ovim sopstvenim vrednostima respektivno. Tada je za svaki vektor y koji je linearna kombinacija ovih vektora sledeći izraz tačan:

$$\lambda_1 \leq \frac{y^T A y}{y^T y} \leq \lambda_k.$$

Jednakost važi kada je y prvi ili poslednji sopstveni vektor.

Definicija 1.1.6. Date su simetrične i pozitivno semidefinitne matrice $A, B \in \mathbf{R}^{n \times n}$. Kažemo da je $A \preceq B$ ako i samo ako za svaki netrivialni vektor $x \in \mathbf{R}^n$ važi

$$x^T A x \leq x^T B x.$$

Teorema 1.1.7. Date su simetrične i pozitivno semidefinitne matrice $A, B \in \mathbf{R}^{n \times n}$ i proizvoljna matrica C . Ako važi $A \preceq B$ onda važi $C^T A C \preceq C^T B C$.

1.1.1 Metode aproksimacije

Sada ćemo ukratko opisati metode koje ćemo koristiti za aproksimaciju matrice dekompozicija i samim tim za aproksimaciju same matrice. Uglavnom pratimo [Halko et al., 2011]. U našem slučaju su sve matrice simetrične i često pozitivno semidefinitne i zato nas interesuju samo metode koji se primenjuju na takvim matricama. Razmatraćemo metode koje nam omogućavaju brzo i tačno izračunavanje aproksimativne dekompozicije matrice. Analiziraćemo jednu vrstu takvih metoda, a to su nasumične aproksimacije (na engleskom 'randomized approximations'). One su našle svoju primenu u algoritmima mašinskog učenja, a popularizovane su sledećim radovima [Halko et al., 2011, Rahimi and Recht, 2008, Bach, 2013]. Dekompozicije na kojima ćemo primenjivati ove metode su SVD

i spektralna dekompozicija. Sada ćemo opisati intuiciju ispred nasumične aproksimacije i početni korak aproksimacije (nalaženje matrice Q) kao u radu [Halko et al., 2011].

Neka je A matrica koju želimo da aproksimiramo. Problem računanja aproksimativne dekompozicije se može podeliti na dva dela. Prvi deo je nalaženje matrice Q čije kolone su ortonormirane, i te kolone matrice Q aproksimiraju bazu prostora kolona matrice A . To jest prostor kolona matrice Q (prostor koji predstavlja raspon skupa vektora kolona od Q) aproksimira prostor kolona matrice A . Tražimo matricu Q takvu da važi

$$A \approx QQ^T A.$$

To znači kada projektujemo matricu na prostor $\mathcal{R}(Q)$ mi dobijamo približno istu matricu. Da bi algoritmi faktorizacije bili što brži matrica Q treba da ima što manje kolona. Nalaženje matrice Q može se obaviti vrlo efikasno nasumičnim (randomizovanim) metodama u kojima ili uzorkujemo neke slučajne podmatrice date matrice ili uzorkujemo nove matrice određenih dimenzija.

Drugi korak je primena matrice Q za računanje same aproksimacije, koja će se računati deterministički, a ne probabilistički.

Prvo ćemo opisati kako nalazimo matricu $Q \in \mathbf{R}^{n \times r}$. Tražimo onu matricu koja minimizuje:

$$\|A - QQ^T A\|.$$

Kako su kolone matrice Q ortonormirane one predstavljaju bazu prostora koji aproksimira prostor $\mathcal{R}(A)$. Sada ćemo dati dve intuicije ispred (randomizovanog) algoritma sa nasumičnim uzorkovanjem za formiranje matrice Q , kako su opisani u [Halko et al., 2011].

Intuicija 1 [Halko et al., 2011]. Pretpostavimo da je rang matrice A jednak r i da tražimo bazu prostora $\mathcal{R}(A)$. Uzorkujemo r slučajnih vektora $\Omega = \{\omega_1, \omega_2, \dots, \omega_r\}$ i izračunamo sledeće vektore $y_i = A\omega_i$. Zbog slučajnosti u formiranju skupa Ω sledi da je velika verovatnoća da je skup vektora Ω linearno nezavistan i da nijedan njegov element ne pripada nultom prostoru matrice A . Prema tome i skup vektora $\{A\omega_1, \dots, A\omega_r\}$ je linearno nezavistan, pa je onda njegov lineal upravo prostor $\mathcal{R}(A)$. Sledi da je Ω baza prostora $\mathcal{R}(A)$.

Pretpostavimo da nemamo pristup matrici A , ali imamo pristup matrici $B = A + E$, gde je E mala greška. Ponovo uzorkujemo skup Ω veličine $r + l$ (gde je l mali prirodan broj koji zovemo parametar prekomernog uzorkovanja), pa izračunamo skup vektora y :

$$\Gamma = \{A\omega_1 + E\omega_1, \dots, A\omega_{r+l} + E\omega_{r+l}\}.$$

Što je l veće, veća je i šansa da prostor koji dobijamo kao lineal vektora iz skupa Γ obuhvata prostor $\mathcal{R}(A)$. U praksi se pokazalo da je vrlo često dovoljno dobro da je $l = 5$.

Intuicija 2 [Halko et al., 2011]. Rešenje sledećeg problema minimizacije

$$\min_{\text{rank}(X) \leq r} \|A - X\|$$

je matrica oblika $X = UU^T A$ gde je matrica U sastavljena od r najvećih singularnih vektora matrice A .

Iz ove dve ideje vidimo da se proces formiranja slučajne matrice $Q \in \mathbf{R}^{n \times r}$ (čije kolone su baza prostora koji aproksimira $\mathcal{R}(A)$) može sumirati na sledeći način kao u [Halko et al., 2011]:

- Generisati Gausovu slučajnu matricu

$$\Omega \in \mathbf{R}^{n \times (r+l)},$$

gde je l mali parametar prekomernog uzorkovanja (obično 5).

- Konstruisati matricu

$$Y = A\Omega \in \mathbf{R}^{n \times (r+l)}.$$

Parametar prekomernog uzorkovanja l je izabran jer povećava šanse da matrica Y obuhvati r -dimenzionalni potprostor prostora kolona $\mathcal{R}(A)$.

- Izvršiti QR dekompoziciju matrice Y . Matrica Q nam daje sledeću aproksimaciju $A \approx QQ^T A$.

1.1.2 Ulazna matrica je proizvoljna kvadratna matrica

Objasnimo sada na jednom primeru (metod iz [Halko et al., 2011]) kako bi se računala aproksimacija SVD-a kada znamo matricu Q . Pretpostavimo da je matrica A dimenzije $n \times n$ (ne mora da bude simetrična) i aproksimiramo je matricom malog ranga r . Prema tome aproksimacija dekompozicije indukuje matricu ranga r . Zato prvo izračunamo matricu Q dimenzije $n \times r$. Zatim, izračunamo matricu $B = Q^T A$. Sledi da matrica QB aproksimira matricu A jer

$$QB = QQ^T A \approx A.$$

Primetimo da je dimenzija matrice B jednaka $r \times n$. Vremenska složenost računanja SVD dekompozicije matrice B je zbog toga dosta manja nego složenost računanja dekompozicije matrice A , pa izračunamo tu dekompoziciju:

$$B = U\Lambda V^T.$$

Ako pomnožimo ovaj izraz matricom Q sa leve strane dobijamo matricu koja je približna matrici A :

$$A \approx QU\Lambda V^T.$$

Samim tim dobijeni izraz je i aproksimativna dekompozicija matrice A .

Pretpostavimo da sada računamo spektralnu dekompoziciju matrice A , koja je simetrična i pozitivno semidefinitna. Matrica Q je takva da njen prostor kolona aproksimira $\mathcal{R}(A)$ (izračunata kao u prethodnom koraku). Primetimo sledeće:

$$\begin{aligned} A &\approx QQ^T A, \\ A^T &\approx A^T QQ^T, \\ A &\approx AQQ^T, \\ A &\approx QQ^T AQQ^T, \\ A &\approx QBQ^T, \end{aligned}$$

za matricu B koja je izračunata kao $B = Q^T A Q$. Dimenzija matrice B je $r \times r$, i ta matrica je isto simetrična i pozitivno semidefinitna. Izračunamo spektralnu dekompoziciju ove matrice:

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

Odatle sledi da je

$$A \approx QU \Lambda U^T Q^T$$

aproksimativna spektralna dekompozicija matrice A .

1.1.3 Ulazna matrica je simetrična i pozitivno semidefinitna

U prethodnom poglavlju smo uveli metode za aproksimativnu faktorizaciju kvadratne matrice koja može, a i ne mora biti simetrična. Sada ćemo posmatrati slučaj kada je matrica simetrična i pozitivno semidefinitna $A \in \mathbf{R}^{n \times n}$. Podsetimo se da je matrica pozitivno semidefinitna ako i samo ako važi $x^T A x \geq 0$ za svaki netrivialni n -dimenzionalni vektor x . U ovom slučaju Nistromov metod se može iskoristiti da poboljša kvalitet aproksimativne dekompozicije. Sam Nistromov metod idejno dolazi iz analize integralnih jednačina [Baker, 1977], ali je uveden u oblast mašinskog učenja radom [Williams and Seeger, 2001]. Našao je dosta primena u drugim algoritmima:

- Gausovi procesi [Williams and Seeger, 2001]
- polu nadgledano učenje [McWilliams et al., 2013, Sinha and Belkin, 2009]
- spektralno klasterovanje [Fowlkes et al., 2004, Choromanska et al., 2013]

Nistromov metod bazira svoju aproksimaciju na osnovu uzorkovanja malog broja kolona matrice. U praksi se pokazalo kao najbolje rešenje uniformno nasumično biranje kolona bez zamene. Naravno predložene su i druge metode uzorkovanja [Kumar et al., 2009], kao i razne ekstenzije samog algoritma [Li et al., 2015b]. Ovaj metod ćemo primenjivati za aproksimaciju kernel (Gramove) matrice i za aproksimaciju Laplasove matrice radi razvijanja brzih algoritama.

Opišimo sada Nistromov metod ukratko kao u [Williams and Seeger, 2001, Choromanska et al., 2013]. Neka je \mathcal{L} skup nasumično izabranih indeksa kolona. Definišemo

maticu $C = A(\mathcal{L}, :)$ i njena dimenzija je $l \times n$. Matrica Q je izvedena iz C izborom l redova na sledeći način $Q = C(:, \mathcal{L})$. Nistromov metod kaže da sledeća matrica aproksimira matricu A :

$$\tilde{A} = CQ^+C^T.$$

Kako je matrica A simetrična i pozitivno semidefinitna sledi da je i matrica Q simetrična i pozitivno semidefinitna, ali mnogo manje dimenzije od matrice A , pa možemo izračunati njenu spektralnu dekompoziciju

$$Q = U_Q \Sigma_Q U_Q^T,$$

u maloj vremenskoj složenosti. Odavde vidimo da Nistromova metoda generiše i aproksimaciju dekompozicije matrice A sa sledećim sopstvenim vrednostima i sopstvenim vektorima:

$$A \approx \left(\sqrt{\frac{l}{n}} C U_Q \Sigma_Q^+\right) \left(\frac{n}{l} \Sigma_Q\right) \left(\sqrt{\frac{l}{n}} C U_Q \Sigma_Q^+\right)^T. \quad (1.1)$$

Primena ovih algoritama na probleme koje ćemo rešavati se nalazi u glavama 2, 3 i 4. Nistromova metoda i njene primene analizirane su u [Halko et al., 2011, Mahoney, 2011, Bach, 2013, Choromanska et al., 2013, Rahimi and Recht, 2008, Kumar et al., 2009, Williams and Seeger, 2001].

1.2 Klasterovanje

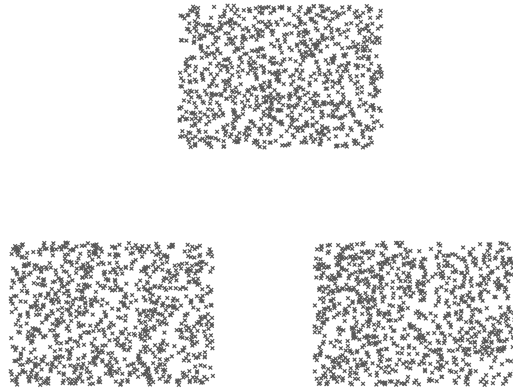
U ovom poglavlju ćemo prikazati neke poznate algoritme za klasterovanje iz literature [Law, 2006, Höppner et al., 2000, Aggarwal and Wang, 2010, Von Luxburg, 2007, Ng et al., 2002].

Cilj klasterizacije podataka, ili analize klastera, je da pronade najbolji način grupisanja skupa podataka. Način grupisanja zavisi od aplikacije do aplikacije. Sama ocena kvaliteta klastera takođe zavisi od aplikacije i ne postoji univerzalan način ocene kvaliteta. Na slici 1.1 vidimo primer podataka za klasterovanje, a na slici 1.2 sam rezultat klasterovanja.

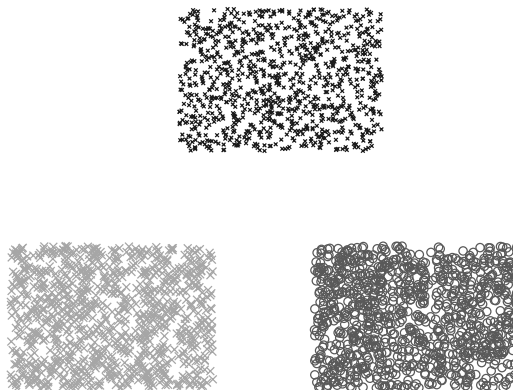
Analiza klastera se primenjuje u svakoj oblasti u kojoj analiziramo podatke. Kako primena ima mnogo ovde ćemo navesti samo neke:

- segmentacija slike, može se rešavati kao klasterovanje;
- klasterizacija dokumenata;
- klasterizacija genetskih podataka u biologiji.

Vratimo se na sliku 1.2. Primetimo da su tu klasteri istog oblika i lako se detektuju (recimo najpoznatiji i najjednostavniji algoritam klasterovanja, algoritam k -srednjih vrednosti, će naći korektne klasterne). Sa druge strane, netrivialni primer je prikazan na slici 1.3 i ovde se klasteri ne detektuju lako. Sada imamo 7 klastera koji nisu svi istog oblika: neki su u obliku kružnice, neki u obliku spirale, neki u obliku pravougaonika



Slika 1.1: Slika podataka za klasterovanje.

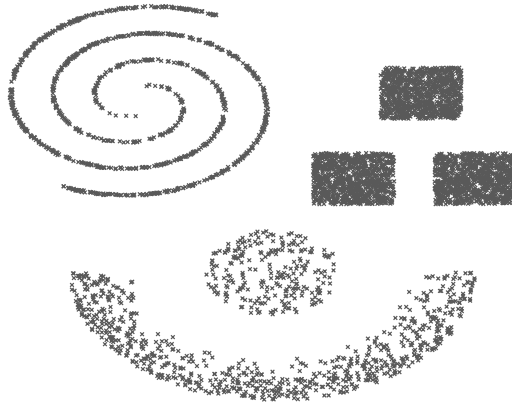


Slika 1.2: Slika u kojima su podaci sa slike 1.1 klasterovani u tri klastera.

itd. Ovaj primer ilustrira fundamentalni problem klasterizacije po [Law, 2006]: Različiti "dobri" klasteri u različitim scenarijima dovode do nemogućnosti definisanja univerzalno "dobrog" klastera.

Cilj analize klastera je da izvrši particiju skupa podataka u klasterne. Ovi klasteri treba da zadovoljavaju sledeće osobine (kao što je definisano u [Höppner et al., 2000]):

- homogenost u klasteru \rightarrow podaci u istom klasteru treba da budu što je moguće sličniji;



Slika 1.3: Slika podataka gde klasterizacija nije jednostavna zato što klasteri imaju različite oblike.

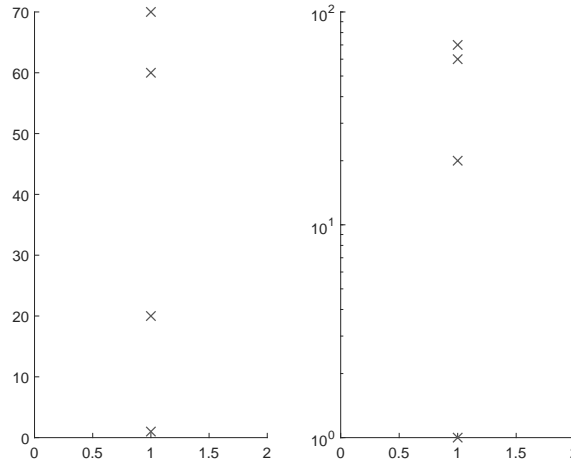
- heterogenost između klastera → podaci koji pripadaju različitim klasterima treba da budu što je moguće različitiji.

Mera sličnosti treba da se definiše u zavisnosti od podataka. Posmatrajući repozitorijum [Lichman, 2013] primećujemo da su podaci u najvećem slučaju realni višedimenzionalni vektori, pa se kao mera sličnosti može uzeti rastojanje između tih vektora. Ako koristimo Euklidsko rastojanje svi elementi (koordinate) vektora će imati istu važnost. Kod nekih skupova podataka se može desiti da različite koordinate imaju različitu važnost, pa korišćenje mere sličnosti koja ovo ne uzima u obzir nećemo dobiti korektne klasterne. Na slici 1.4 ovaj primer je lepo ilustrovan. U prvom delu očigledno vidimo dva klastera $\{x_1, x_2\}$ i $\{x_3, x_4\}$. Ako koordinate skaliramo logaritamskom skalom onda dobijamo drugačiji izgled podataka, kao na drugom delu slike. I ovde takođe vidimo dva klastera, ali su oni drugačiji $\{x_1\}$ i $\{x_2, x_3, x_4\}$. Prema tome izbor mere sličnosti utiče na rezultata klasterovanja, kvalitet rezultata klasterovanja i treba se dobro odabrati u zavisnosti od ulaznog skupa podataka.

Podaci koji se klasteruju mogu biti proizvoljnog tipa dok god je mera sličnosti definisana na njima. U sledećem poglavlju ćemo opisati najpoznatiji i najčešće korišćeni algoritam za klasterovanje, zvan algoritam k -srednjih vrednosti.

1.2.1 Algoritam k -srednjih vrednosti

Dobro poznati algoritam u analizi klastera je algoritam k -srednjih vrednosti (eng. " *k-means algorithm* "). Ovde ga opisujemo prateći [Law, 2006]. Neka svi podaci $\{x_1, x_2, \dots, x_n\}$ pripadaju vektorskom prostoru \mathbb{R}^d . Svaki klaster je predstavljen jednim njegovim elemen-



Slika 1.4: Slike podataka sa različitim skaliranjem koordinata. Oznake podataka i na jednoj i na drugoj slici su x_1, x_2, x_3 i x_4 označeno odozdo na gore.

tom, prototipom, $\mu_j \in \mathbb{R}^d$ gde je $j = \overline{1, k}$. Imamo i indikatore z_i koji označavaju kom klasteru pripada element i . Izlaz ovog algoritma je minimum sledeće funkcije cene:

$$\mathcal{J}(z, \mu) = \sum_{i=1}^n \|x_i - \mu_{z_i}\|^2$$

$$\mathcal{J}(z, \mu) = \sum_{i=1}^n \sum_{j=1}^k I(z_i = j) \|x_i - \mu_j\|^2$$

gde je $I(z_i = j)$ indikatorska funkcija. Sada tražimo optimum funkcije. Prvo računamo $\{z_1, z_2, \dots, z_n\}$ ako znamo $\{\mu_1, \mu_2, \dots, \mu_k\}$ i to na sledeći način:

$$z_i = \arg \min_j \|x_i - \mu_j\|^2$$

Zatim računamo prototipe klastera $\{\mu_1, \mu_2, \dots, \mu_k\}$ ako znamo $\{z_1, z_2, \dots, z_n\}$ tako da oni minimiziraju funkciju cene. To ćemo odraditi na sledeći način:

$$\begin{aligned} \frac{\partial \mathcal{J}(z, \mu)}{\partial \mu_j} &= 2 \sum_{i=1}^n I(z_i = j) (x_i - \mu_j) \\ 2 \sum_{i=1}^n I(z_i = j) (x_i - \mu_j) &= 0 \\ 2 \sum_{i=1}^n I(z_i = j) x_i - 2 \sum_{i=1}^n I(z_i = j) \mu_j &= 0 \end{aligned}$$

Algoritam 1 Algoritam k -srednjih vrednosti.

```

procedure KMEANS( $X, k, \epsilon$ )
  Input: skup podataka  $X$ ;
  Input: broj željenih klastera  $k$ ;
  Input: greška  $\epsilon$ ;
  Inicijalizujemo  $(\mu_1, \mu_2, \dots, \mu_k)$ ;
   $\mathcal{J} := \infty$ ;
  while true do
    for all  $i$  do
       $z_i := \arg \min_j \|x_i - \mu_j\|^2$ ;
    end for
    for all  $j$  do
       $\mu_j := \frac{\sum_{i=1}^n I(z_i=j) x_i}{\sum_{i=1}^n I(z_i=j)}$ ;
    end for
     $\mathcal{J}_{new} := \sum_{i=1}^n \|x_i - \mu_{z_i}\|^2$ ;
    if  $\|\mathcal{J} - \mathcal{J}_{new}\| \leq \epsilon$  then
      break;
    end if
     $\mathcal{J} := \mathcal{J}_{new}$ ;
  end while
  Output: Indikator klastera  $\mu$ ;
end procedure

```

$$\sum_{i=1}^n I(z_i = j) x_i = \mu_j \sum_{i=1}^n I(z_i = j)$$

$$\mu_j = \frac{\sum_{i=1}^n I(z_i = j) x_i}{\sum_{i=1}^n I(z_i = j)}$$

Startujući od nekih inicijalnih prototipa algoritam iterira sve dok ne dođe u lokalni minimum. Rezultujući z i μ definišu rešenje. U praksi algoritam se može zaustaviti kada je razlika između dvaju uzastopnih vrednosti manja od nekog praga.

Prednosti ovog algoritma su jednostavnost i lakoća implementacije. Međutim on ima i mane. Ako klasteri nisu konveksni (čak i sfernog oblika) algoritam neće naći dobro rešenje [Law, 2006]. Takođe nailazi na probleme kada različiti klasteri imaju veoma različit broj elemenata [Law, 2006]. Algoritam spektralnog klasterovanja rešava ove probleme i opisaćemo ga u narednom poglavlju.

1.2.2 Grafovski algoritam klasterovanja: spektralno klasterovanje

Spektralno klasterovanje [Ng et al., 2002] pripada klasi grafovskih algoritama i opisujemo ga kao u [Von Luxburg, 2007]. Na ulazu imamo neusmereni težinski graf (često

nazvan graf sličnosti), $G = (V, W)$ sa skupom od n čvorova $V = \{v_1, v_2, \dots, v_n\}$ i matricom susedstva W . Konkretno, W je simetrična matrica gde je pozitivan realan broj w_{ij} težina ivice (v_i, v_j) . Ako je $W_{ij} = 0$, tada nema ivice između v_i i v_j . Pored grafa na ulazu imamo i željeni broj klastera. Cilj ovog algoritma je grupisanje čvorova grafa tako da važi (po [Von Luxburg, 2007]):

- suma grana čiji krajnji čvorovi pripadaju istoj grupi (klasteru) treba da bude što veća,
- suma grana koje povezuju čvorove koji pripadaju različitim klasterima treba da bude što manja.

Ovaj algoritam pripada klasi spektralnih algoritama zato što je metod baziran na analizi spektra Laplasove matrice grafa. Može se primeniti i na slučaj kada na ulazu imamo skup vektora, meru sličnosti (najčešće kernel funkcija) i broj željenih klastera. Naravno na ulazu ne moramo da imamo skup vektora, možemo da imamo bilo koji skup predmeta (instanci) na kome je definisana mera sličnosti. Kako algoritam spektralnog klasterovanja na ulazu zahteva graf, prvi korak će biti izvođenje grafa iz ulaznog skupa. Taj graf se zove graf sličnosti. U narednom poglavlju ćemo navesti neke načine za formiranje grafova sličnosti.

Uvedimo neke osnovne pojmove [Von Luxburg, 2007, Ng et al., 2002, Shi and Malik, 2000] iz teorije grafova koje ćemo koristiti u tezi:

- Matrica stepena je dijagonalna matrica takva da se stepen čvora v_i nalazi na i -om mestu dijagonale:

$$D_{ii} = \sum_{j=1}^n W_{ij}.$$

- Obim podskupa temena A je zbir stepena čvorova u A :

$$\text{vol}(A) = \sum_i D_{ii}.$$

- Presek (rez) između dva podskupa čvorova A i B definiše se kao zbir ivica koje ih povezuju:

$$\text{cut}(A, B) = \sum_{i:v_i \in A} \sum_{j:v_j \in B} w_{ij}.$$

- Presek (rez) podskupa čvorova A je presek između A i ostalih čvorova grafa:

$$\text{cut}(A, V \setminus A) = \sum_{i:v_i \in A} \sum_{j:v_j \notin A} w_{ij},$$

$$\text{cut}(A) = \sum_{i:v_i \in A} \sum_{j:v_j \notin A} w_{ij}.$$

- Za skup particija od k particija (A_1, A_2, \dots, A_k) iz skupa čvorova V , mera reza definisana je kao suma rezova svakog skupa:

$$Cut(A_{1:k}) = \sum_{i=1}^k cut(A_i, V \setminus A_i),$$

$$Cut(A_{1:k}) = \sum_{i=1}^k cut(A_i).$$

- Za skup particija od k particija (A_1, A_2, \dots, A_k) iz skupa čvorova V , normalizovana mera reza definisana je kao normalizovana suma rezova svakog skupa:

$$NCut(A_{1:k}) = \sum_{i=1}^k \frac{cut(A_i, V \setminus A_i)}{vol(A_i)},$$

$$NCut(A_{1:k}) = \sum_{i=1}^k \frac{cut(A_i)}{vol(A_i)}.$$

1.2.3 Graf sličnosti

Neka je dato n ulaznih primera $\{x_1, x_2, \dots, x_n\}$ i mera sličnosti s na njima. Od svakog ulaznog primera formiramo čvor. Čvoru v_i odgovara podatak x_i . Prema tome skup čvorova grafa sličnosti je:

$$V = \{v_1, v_2, \dots, v_n\}.$$

Sada formiramo skup grana. Prevodimo sličnosti u težine grafa, pa računamo w_{ij} (težina grane koja povezuje čvor v_i i čvor v_j) koja modelira sličnost s_{ij} (sličnost između podataka i i podataka j). Postoji više metoda za formiranje skupa grana. Ovde ćemo opisati neke od njih kao što su opisani u [Von Luxburg, 2007]:

1. k -najbližih suseda,
2. ϵ -okolina,
3. kompletan graf gde težine predstavljaju sličnost.

Objasnimo sada svaki od ovih metoda malo detaljnije.

Algoritam k -najbližih suseda [Von Luxburg, 2007]. Ovde povezujemo čvor v_i sa čvorom v_j ako je čvor v_j među k najbližih suseda čvora v_i . Kada kažemo najbliži mislimo najbliži. Primetimo da u ovom slučaju graf postaje usmeren, zato što ako je čvor v_j među k najbližih suseda čvora v_i ne mora da važi i obratno. Postoje dva načina za rešavanje ovog problema:

1. Ako je čvor v_j među k najbližih suseda čvora v_i dodamo neusmerenu granu koja povezuje čvorove v_i i v_j . Imajmo na umu da u ovom slučaju možemo da imamo čvorove čiji je stepen veći od k .
2. Ako je čvor v_i među k najbližih suseda čvora v_j i ako je čvor v_j među k najbližih suseda čvora v_i dodamo neusmerenu granu koja povezuje čvorove v_i i v_j . U suprotnom ne dodajemo nijednu granu. Imajmo na umu da u ovom slučaju možemo da imamo čvorove čiji je stepen manji od k .

Kod ovog algoritma težine grana mogu biti odabrane na sledeći način:

1. $w_{ij} = 1$
2. $w_{ij} = s_{ij}$

Algoritam ϵ -okolina [Von Luxburg, 2007]. Ovde povezujemo čvor v_i sa čvorom v_j ako je sličnost $s_{ij} < \epsilon$. Imajmo na umu da je mera sličnosti simetrična pa prema tome ako povezujemo čvor v_i sa čvorom v_j onda povezujemo i čvor v_j sa čvorom v_i . Kako su rastojanja između čvorova koji su povezani otprilike ista, najviše ϵ u ovom slučaju, graf koji formiramo je najčešće bestežinski. Zato stavimo da je $w_{ij} = 1$ ako postoji grana između v_i i v_j .

Algoritam kompletan graf [Von Luxburg, 2007]. U ovom slučaju povežemo sve čvorove grafa jedan sa drugim. Naravno kako je graf težinski moramo da odaberemo i težine i zato stavimo da je težina grane koja povezuje čvorove v_i i v_j jednaka

$$w_{ij} = s_{ij}.$$

Najčešće se koristi Gausova mera sličnosti (tj. Gausov kernel) u slučaju da su ulazni podaci vektori:

$$s(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right).$$

Svi ovi algoritmi se koriste u algoritmu spektralnog klasterovanja [Von Luxburg, 2007]. Ne posotoji način da se odredi koji je algoritam od njih najbolji [Von Luxburg, 2007]. U ovoj tezi ćemo za empirijsku evaluaciju najčešće koristiti algoritam tri u kome se formira kompletan graf.

1.2.4 Laplasova matrica grafa

Glavni alat koji se koristi u spektralnoj teoriji grafova je grafovska Laplasova matrica ili grafovski Laplasijan [Von Luxburg, 2007]. Imajmo na umu da ne postoji univerzalna definicija Laplasove matrice. U literaturi može naći više verzija. U ovom i narednom poglavlju ćemo definisati dva tipa Laplasovih matrica koje ćemo koristiti u daljem tekstu. Opisujemo same matrice i njihovu primenu na problemu klasterovanja kao u [Von Luxburg, 2007].

Naravno kako ćemo klasterovanje primenjivati na neusmerenim grafovima pretpostavimo da je graf koji ćemo analizirati i na kome ćemo definisati Laplasovu matrici neusmeren. Takođe pretpostavimo da je graf težinski. Znači w_{ij} predstavlja težinu grane koja povezuje čvor v_i i čvor v_j , ako ta grana postoji a grana postoji ako je $w_{ij} > 0$.

Pretpostavimo sledeće:

- Kada kažemo sopstveni vektor mislimo na normalizovan vektor. Prema tome kada kažemo skup svih sopstvenih vektora mislimo na ortonormiranu bazu.
- Kada kažemo vektor jedinica mislimo na vektor čiji su svi elementi jedinice $\mathbf{1}$.
- Kada kažemo prvih k sopstvenih vektora mislimo na one sopstvene vektore koji odgovaraju najmanjim sopstvenim vrednostima.

Sada ćemo definisati dva tipa Laplasove matrice i opisati neke njihove osnovne osobine.

1.2.5 Nenormalizovana Laplasova matrica

Laplasova matrica se definiše na sledeći način:

$$L = D - W.$$

Osobine 1. [Von Luxburg, 2007] Za grafov Laplasijan L sledeći iskazi su tačni:

- Za svaki vektor $f \in \mathbf{R}^n$ važi

$$f^T L f = \frac{1}{2} \sum_{i,j} w_{ij} (f_i - f_j)^2.$$

- L je simetrična i pozitivno semidefinitna matrica,
- L ima n nenegativnih realnih sopstvenih vrednosti:

$$0 = \lambda_1 \leq \dots \leq \lambda_n,$$

- 0 je sopstvena vrednost matrice L kojoj odgovara sopstveni vektor $\mathbf{1}$.

Dokaz. Sada ćemo pokazati ove osobine:

- Iz definicije i sređivanjem izraza dobijamo:

$$\begin{aligned} f^T L f &= f^T D f - f^T W f \\ f^T L f &= \sum_{i,j} D_{ij} f_i f_j - \sum_{i,j} w_{ij} f_i f_j \\ f^T L f &= \sum_i d_i f_i^2 - \sum_{i,j} w_{ij} f_i f_j \end{aligned}$$

$$f^T L f = \frac{1}{2} \left(\sum_i d_i f_i^2 - 2 \sum_{i,j} w_{ij} f_i f_j + \sum_j d_j f_j^2 \right)$$

$$f^T L f = \frac{1}{2} \left(\sum_{ij} w_{ij} f_i^2 - 2 \sum_{i,j} w_{ij} f_i f_j + \sum_{ij} w_{ij} f_j^2 \right)$$

$$f^T L f = \frac{1}{2} \sum_{i,j} w_{ij} (f_i - f_j)^2.$$

- Simetričnost matrice sledi iz simetričnosti matrice susedstva i iz simetričnosti dijagonalne matrice stepena. Iz prve osobine tvrđenja sledi da za svaki vektor f važi

$$f^T L f \geq 0$$

pa sledi da je matrica pozitivno semidefinitna.

- Primitimo da je

$$L\mathbf{1} = (D - W)\mathbf{1} = \sum_{ij} D_{ij}\mathbf{1}_j - \sum_{ij} w_{ij}\mathbf{1}_j$$

$$L\mathbf{1} = \sum_{ij} D_{ij} - \sum_{ij} w_{ij} = 0$$

odakle sledi tvrđenje.

- Iz činjenice da je matrica simetrična sledi da su njene sopstvene vrednosti realne, a iz činjenice da je matrica pozitivno semidefinitna sledi da su one nenegativne.

□

Primitimo da grane koje povezuju čvor sam sa sobom ne utiču na Laplasovu matricu [Von Luxburg, 2007]. Takođe multiplikativnost sopstvene vrednosti 0 predstavlja broj povezanih komponenti datog grafa [Mohar, 1997]. Laplasova matrica je analizirana u radovima [Mohar et al., 1991, Mohar, 1997].

1.2.6 Normalizovana Laplasova matrica

Postoji više matrica u literaturi koje se zovu normalizovana Laplasova matrica. U ovoj tezi normalizovanom Laplasovom matricom ćemo označiti sledeću matricu:

$$L = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}.$$

Iako je spektralno klasterovanje uvedeno još 70-ih godina u [Fiedler, 1973, Donath and Hoffman, 2003], sam algoritam je popularizovan primenama u sledećim radovima [Fowlkes et al., 2004, Ng et al., 2002]. Ovu definiciju normalizovane Laplasove matrice ćemo koristiti zato što je to matrica koja je korišćena prilikom spektralnog klasterovanja u radovima [Shi and Malik, 2000, Ng et al., 2002]. Primitimo da su autori [Shi and Malik, 2000] primenili ovaj algoritam na segmentaciju slike, dok je recimo rad [Ng et al., 2002] jedan od retkih koji daje teorijsku analizu algoritma.

Osobine 2. [Von Luxburg, 2007] Za normalizovanu Laplasovu matricu L sledeći iskazi su tačni:

- Za svaki vektor $f \in \mathbf{R}^n$ važi

$$f^T L f = \frac{1}{2} \sum_{i,j} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2.$$

- L je simetrična i pozitivna semidefinitna matrica,
- L ima n nenegativnih realnih sopstvenih vrednosti:

$$0 = \lambda_1 \leq \dots \leq \lambda_n,$$

- 0 je sopstvena vrednost matrice L kojoj odgovara sopstveni vektor $D^{\frac{1}{2}} \mathbf{1}$.

Dokaz. Sada ćemo pokazati ove osobine:

- Iz definicije i sređivanjem izraza dobijamo:

$$f^T L f = f^T D^{-\frac{1}{2}} (D - W) D^{-\frac{1}{2}} f = f^T f - f^T D^{-\frac{1}{2}} W D^{-\frac{1}{2}} f$$

$$f^T L f = \sum_i f_i^2 - \sum_{i,j} (D^{-\frac{1}{2}} f)_i (D^{-\frac{1}{2}} f)_j w_{ij}$$

$$f^T L f = \sum_i f_i^2 \frac{d_i}{d_i} - \sum_{i,j} \frac{f_i f_j}{d_i d_j} w_{ij}$$

$$f^T L f = \sum_i f_i^2 \frac{1}{d_i} \sum_j w_{ij} - \sum_{i,j} \frac{f_i f_j}{d_i d_j} w_{ij}$$

$$f^T L f = \frac{1}{2} \sum_{i,j} \frac{f_i^2}{d_i} w_{ij} + \frac{1}{2} \sum_{i,j} \frac{f_j^2}{d_j} w_{ij} - \frac{1}{2} \sum_{i,j} 2 \frac{f_i f_j}{d_i d_j} w_{ij}$$

$$f^T L f = \frac{1}{2} \sum_{i,j} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2.$$

- Simetričnost matrice sledi iz simetričnosti matrice susedstva i iz simetričnosti dijagonalne matrice stepena. Iz prve osobine tvrđenja sledi da za svaki vektor f važi

$$f^T L f \geq 0$$

pa sledi da je matrica pozitivno semidefinitna.

- Iz činjenice da je matrica simetrična sledi da su njene sopstvene vrednosti realne, a iz činjenice da je matrica pozitivno semidefinitna sledi da su one nenegativne.

- Primitimo da je

$$LD^{\frac{1}{2}}\mathbf{1} = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}D^{\frac{1}{2}}\mathbf{1} = D^{-\frac{1}{2}}(D - W)\mathbf{1} = 0,$$

odakle sledi tvrđenje.

□

Kao i kod obične Laplasove matrice i kod normalizovane Laplasove matrice multiplikativnost sopstvene vrednosti 0 predstavlja broj povezanih komponenti datog grafa. Ova osobine i ostale osobine simetrične Laplasove matrice se mogu naći u [Chung and Graham, 1997].

1.2.7 Intuitivni prikaz spektralnog klasterovanja: relaksacija problema nalaženja minimalnog reza datog grafa

Sada ćemo opisati intuiciju ispred spektralnog klasterovanja kao što je opisano u [Von Luxburg, 2007], a koja idejno dolazi iz rada [Shi and Malik, 2000]. Šta je rezultat algoritma klasterovanja? Rezultat su grupe podataka tako da su podaci u grupi slični, a podaci koji pripadaju različitim grupama različiti. Kada od podataka formiramo graf sličnosti ovu ideju možemo preformulisati na sledeći način: formiramo grupe čvorova tako da je suma grana koje povezuju čvorove iz istog klastera što veća, a suma grana koje povezuju čvorove iz različitih klastera što manja. Sledi da klasterovanje na grafu možemo definisati kao problem minimalnog reza

$$\min_{A_{1:k}} cut(A_{1:k}).$$

U slučaju da je $k = 2$, imamo samo dva klastera. Ovaj problem se može rešiti efikasno [Stoer and Wagner, 1997]. Problem klasterovanja grafa kada imamo samo dva klastera se na drugi način zove bisekcija ili bipartcija grafa. U opštem slučaju, kada je k proizvoljan broj, rešenje ovog problema minimizacije ne daje uvek zadovoljavajuće rešenje u praksi. Vrlo često se može desiti da algoritam napravi klustere koji sadrže samo jedan čvor, što sa druge strane nije ono što želimo jer klasteri obično ne treba da budu tako mali [Von Luxburg, 2007]. Zato se treba definisati neka funkcija cene koja će ograničiti veličinu samih klastera. U literaturi postoji više funkcija cene koje se koriste i više načina za merenje veličine klastera, videti [Hagen and Kahng, 1992, Shi and Malik, 2000]. Mi ćemo koristiti normalizovani rez grafa za funkciju cene kao u radovima [Shi and Malik, 2000, Ng et al., 2002]. Nju koristimo zato što se ona pokazala u praksi kao vrlo dobra i zato što daje algoritam koji ima jednostavnu implementaciju. Problem klasterovanja se definiše kao problem minimiziranja normalizovanog reza:

$$\min_{A_{1:k}} NCut(A_{1:k}).$$

Rešenje ove optimizacije će biti skup klastera koji su relativno veliki. Idejno (po [Von Luxburg, 2007]) to je zato što ako posmatramo funkciju

$$\sum_{i=1}^k \frac{1}{vol(A_i)}$$

vidimo da ona dostiže minimum kada svaki skup A_i ima isti obim. Prema tome rešenje sledeće $\min_{A_{1:k}} NCut(A_{1:k})$ klasterizacije su balansirani klasteri. Nažalost kada se doda obim u funkciju cene dobijamo problem koji je NP težak [Wagner and Wagner, 1993]. Zato se ova minimizacija relaksira i kao proces relaksacije dobijamo algoritam spektralnog klasterovanja. Sada ćemo to objasniti malo detaljnije. Prvo ćemo posmatrati problem bisekcije grafa kao u radu [Von Luxburg, 2007], tj. slučaj kada je broj željenih klastera jednak dva.

Posmatramo bisekciju grafa koju dobijamo kada minimiziramo normalizovani rez grafa. Imamo dve particije A_1 i A_2 za koje važi da je $A_1 \cup A_2 = V$, jer se svi čvorovi klasteruju u dve klase. Prema tome ako znamo jednu particiju onda znamo i obe particije. Minimizacija normalizovanog reza rešava sledeći problem:

$$\min_{A_1, A_2} NCut(A_{1:2}).$$

Definišimo vektor f na sledeći način:

- Ako čvor v_i pripada particiji A_1 onda stavimo

$$f_i = \sqrt{\frac{vol(V \setminus A_1)}{vol(A_1)}} d_i = \sqrt{\frac{vol(A_2)}{vol(A_1)}} d_i$$

odakle je očigledno $f_i > 0$.

- Ako čvor v_i pripada particiji A_2 onda stavimo

$$f_i = -\sqrt{\frac{vol(A_1)}{vol(V \setminus A_1)}} d_i = -\sqrt{\frac{vol(A_1)}{vol(A_2)}} d_i$$

odakle je očigledno $f_i < 0$.

Prvo izračunamo koliko je $f^T L f$. Iz teoreme 2 znamo da važi:

$$f^T L f = \frac{1}{2} \sum_{i,j} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2$$

$$f^T L f = \frac{1}{2} \sum_{i,j \in A_1} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2 + \frac{1}{2} \sum_{i,j \in A_2} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2 + \sum_{i \in A_1, j \in A_2} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2$$

Sada zamenimo vrednosti elemenata vektora f i dobijamo

$$f^T L f = \sum_{i \in A_1, j \in A_2} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2$$

$$f^T L f = \sum_{i \in A_1, j \in A_2} w_{ij} \left(\frac{\sqrt{vol(A_2)}}{\sqrt{vol(A_1)}} - \frac{\sqrt{vol(A_1)}}{\sqrt{vol(A_2)}} \right)^2$$

$$f^T L f = \sum_{i \in A_1, j \in A_2} w_{ij} (vol(A_1) + vol(A_2)) \left(\frac{1}{vol(A_1)} + \frac{1}{vol(A_2)} \right)$$

$$f^T L f = \text{cut}(A_1, A_2)(\text{vol}(A_1) + \text{vol}(A_2))\left(\frac{1}{\text{vol}(A_1)} + \frac{1}{\text{vol}(A_2)}\right)$$

$$f^T L f = \text{vol}(V) \text{NCut}(A_1, A_2)$$

Izračunajmo sada normu vektora f :

$$f^T f = \sum_i f_i^2$$

$$f^T f = \sum_{i \in A_1} \frac{\text{vol}(A_2) d_i}{\text{vol}(A_1)} + \sum_{i \in A_2} \frac{\text{vol}(A_1) d_i}{\text{vol}(A_2)}$$

$$f^T f = \text{vol}(V)$$

Jednostavno možemo pokazati i da je vektor f ortogonalan na vektor $D^{\frac{1}{2}} \mathbf{1}$. Prema tome minimiziranje normalizovanog reza se može napisati na sledeći način:

$$\min_{f \in \mathbf{R}^n} f^T L f$$

tako da važi

$$f^T f = \text{vol}(V),$$

$$f \perp D^{\frac{1}{2}} \mathbf{1},$$

f ima odgovarajući oblik

Ovaj problem je NP težak i zato ga relaksiramo. To uradimo tako što ignorišemo oblik vektora f , tj. ne mora da važi $f_i = \sqrt{\frac{\text{vol}(A_2)}{\text{vol}(A_1)}} d_i$ ako v_i pripada klasteru A_1 . Relaksacijom dobijamo sledeći optimizacioni problem:

$$\min_{f \in \mathbf{R}^n} f^T L f$$

tako da važi

$$f^T f = \text{vol}(V),$$

$$f \perp D^{\frac{1}{2}} \mathbf{1}.$$

(1.2)

Iz Rejljeve teoreme znamo da je rešenje problema 1.2, sopstveni vektor koji odgovara drugoj sopstvenoj vrednosti. Izračunamo taj vektor, zvani indikatorski vektor, i iz njega izvedemo particije na sledeći način:

- kažemo da čvor v_i pripada klasteru A_1 ako je f_i pozitivno,
- kažemo da čvor v_i pripada klasteru A_2 ako je f_i negativno.

Algoritam 2 Algoritam spektralne bisekcije grafa.

procedure BISEKCIJA(G)

Input: graf G ;

 Izračunamo normalizovanu Laplasovu matricu $L = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$;

 Nađemo drugi sopstveni vektor f matrice L ;

 $A_1 = \emptyset$;

 $A_2 = \emptyset$;

for all i **do**

 if $f_i > 0$ **then**

 $A_1 = A_1 \cup \{v_i\}$;

 end if

 if $f_i < 0$ **then**

 $A_2 = A_2 \cup \{v_i\}$;

 end if
end for
Output: Particije $\{A_1, A_2\}$;

end procedure

Ovo nije jedini način relaksacije problema minimiziranja normalizovanog reza, [Bie and Cristianini, 2006]. Način koji je prikazan u ovoj tezi se koristi najčešće zato što tako dobijamo jednostavno analitičko rešenje, sa jednostavnom implementacijom, koju možemo videti u algoritmu 2.

Posmatrajmo sada opšti slučaj kada je k proizvoljan broj veći od dva. Sada se uvodi indikatorska matrica sledećeg tipa:

$$H_{ij} = \begin{cases} \frac{\sqrt{d_i}}{\sqrt{\text{vol}(A_j)}} & \text{ako } v_i \in A_j, \\ 0 & \text{u suprotnom.} \end{cases} \quad (1.3)$$

Može se pokazati da je problem minimizacije normalizovanog reza ekvivalentan sledećem problemu:

$$\min_{H \in \mathbf{R}^{n \times n}} \text{tr}(H^T L H)$$

tako da važi

$$H^T H = I,$$

 H je definisano kao u (1.3).

Izbacivanjem poslednjeg uslova dobijamo relaksiran problem koji ima jednostavno rešenje:

$$\min_{H \in \mathbf{R}^{n \times n}} \text{tr}(H^T L H)$$

tako da važi

$$H^T H = I.$$

Rešenje ovog problema (po Rejljevoj teoremi) je matrica H čije su kolone prvih k sopstvenih vektora matrice L . Kada se dobije indikatorska matrica H potrebno je izračunati klustere. Oni se računaju primenom algoritma k -srednjih vrednosti na vrste matrice H kao što je predloženo u radu [Shi and Malik, 2000]. Ovaj algoritam se zove algoritam spektralnog klasterovanja (videti pseudokod 3). U radu [Ng et al., 2002] u je spektralno klasterovanje teorijski analizirano, i to je jedan od retkih radova koji daje ovakvu analizu. Algoritam spektralnog klasterovanja nije jedini način relaksacije problema nalaženja minimalnog normalizovanog reza ali je najčešće korišćen zbog njegove jednostavnosti.

Algoritam 3 Spektralno klasterovanje.

[Ng et al., 2002]

Input: W i k ;

for all i, j **do**

if $i = j$ **then**

$D_{ii} := \sum_l w_{il}$;

else

$D_{ij} := 0$;

end if

end for

$L = L = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$.

Izračunati H čije su kolone prvih k sopstvenih vektora matrice L ;

Normalizovati vrste matrice H ;

Klasterovati vrste matrice H primenom algoritma k -srednjih vrednosti 1;

Output: Klasteri C_1, \dots, C_k ;

Primetimo da algoritam spektralnog klasterovanja može da napravi klustere proizvoljnog oblika što nije slučaj sa algoritmom k -srednjih vrednosti (vidi sliku 1.5).

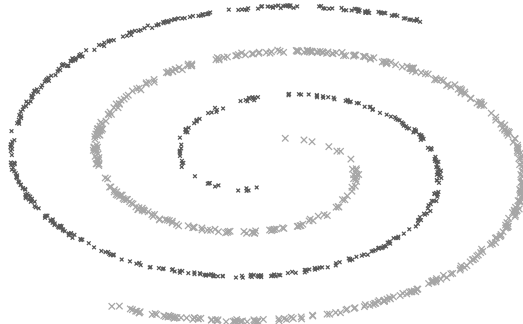
Da zaključimo, pozitivna strana algoritma spektralnog klasterovanja je da on nudi nekoliko prednosti u poređenju sa ostalim algoritmima klasterovanja:

- jednostavna funkcija cene sa analitičkim rešenjem,
- jednostavna implementacija,
- sposobnost izvođenja klastera proizvoljnih oblika.

Međutim, sa negativne strane, spektralno grupisanje ima sledeće probleme:

- činjenica da se u velikoj meri i isključivo oslanja na kvalitet mere sličnosti,
- vremenska složenost po broju primera.

Ovi problemi spektralnog klasterovanja su razmatrani u narednim glavama. Dodavanje ograničenja (koja nam kažu da li dva čvora treba staviti u isti klaster ili ne) je način kojim se redukuje oslanjanje na meru sličnosti. Zato smo u ovoj tezi rešavali problem spektralnog klasterovanja sa ograničenjima. Rešili smo ga implementacijom algoritma u kome smo prvo formirali višeslojni graf time što smo ograničenja kodirali novim grafovima, a zatim smo klasterovali njegove čvorove. Ovaj algoritam ima kvadratnu vremensku



Slika 1.5: Slike podataka koji se mogu klasterovati algoritmom spektralnog klasterovanja a ne algoritmom k -srednjih vrednosti.

složenost koja je smanjena primenom Nistromove metode. Dakle u glavi 2 smo implementirali algoritme za spektralno klasterovanje sa ograničenjima linearne složenosti.

Algoritmi sa ograničenjima zavise od kvaliteta samih ograničenja. Zbog toga smo u glavi 3 razvijali algoritam koji može sam da poboljša skup ograničenja. Kao rešenje ovog problema smo implementirali algoritam spektralnog klasterovanja sa samoučenjem linearne složenosti.

1.3 Kernel regresija

U ovom poglavlju ćemo uvesti problem regresije koji ćemo razmatrati u glavi 4. Ovde pratimo literaturu i opisujemo neke algoritme, metode i zaključke iz [Bach, 2013, Hastie et al., 2009, Schölkopf et al., 2002a]. Pretpostavimo da imamo sledeći trening skup na ulazu $\mathcal{D} = \{x_{1:n}, y_{1:n}\}$, gde su vektori x_i trening primeri (instance, podaci), a skalar y_i je oznaka (labela) odgovarajućeg primera. Cilj problema regresije je da nauči (estimira) funkciju predikcije (prediktor, regresor) f koja treba da modelira zavisnost između ulaznih vektora i izlaznih oznaka. Prema tome treba da važi:

$$f(x_i) \approx y_i.$$

Funkcija se trenira na ulaznom trening skupu, ali treba dobro da generalizuje zavisnost tj. da na neviđenom skupu dobro označava neviđene vektore. Algoritam učenja se može definisati na sledeći način kao optimizacioni problem:

$$\min_f \frac{1}{n} \sum_{i=1, n} \ell(f(x_i), y_i) + \frac{\lambda}{2} \Omega(f)$$

gde je ℓ funkcija cene ili greška predikcije jednog primera, a funkcija Ω je regularizaciona funkcija. Takođe ulazne primere možemo preslikati u neki drugi prostor i rešavati problem u njemu. Recimo da je to preslikavanje $\phi(x_i)$. U tom slučaju se definiše funkcija predikcije čiji je ulazni parametar $\phi(x_i)$, pa prema tome za nju treba da važi $f(\phi(x_i)) \approx y_i$. Mi ćemo posmatrati dva slučaja ovog problema:

- linearna regresija i
- kernel regresije.

Uvedimo pojam kernela sledećom definicijom.

Definicija 1.3.1. Kernel je pozitivno semidefinitna funkcija $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{R}$ ako i samo ako za svaki skup $\{x_1, x_2, \dots, x_n\} \subset \mathcal{X}$ važi da je matrica (zvana Gramova) $K \in \mathbf{R}^{n \times n}$ definisana na sledeći način

$$(\forall i, j) \quad K_{ij} = k(x_i, x_j)$$

pozitivno semidefinitna.

U oba slučaja funkcija cene je kvadratna greška. Kod linearne regresije funkcija predikcije je $f(x_i) = w^T x_i$ gde je w vektor parametara koji se estimira kao

$$w = (X^T X + n\lambda I)^{-1} X^T y.$$

Kod kernel regresije pretpostavljamo da imamo pristup kernelu $k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$, pa je funkcija predikcije $f(x) = \sum_i \alpha_i k(x, x_i)$ gde je

$$\alpha = (K + n\lambda I)^{-1} y.$$

Primetimo da je vremenska složenost rešenja kernel regresije $O(n^3)$, a memorijska složenost $O(n^2)$. Ovo je nerešivo za velike ulazne skupove. Zato se primenjuju metode aproksimacije radi linearizacije vremenske i memorijske složenosti. U radu [Bach, 2013] je primenjena Nistromova metoda na kernel regresiju i analizirana je greška koju ova regresija pravi. Mi smo u glavi 4 razmatrali problem aproksimacije kernel regresije ekstenzijom Nistromove metode. Rešavanjem ovog problema razvili smo algoritam koji daje bolje empirijske rezultate nego algoritam iz [Bach, 2013], a teorijski ima sličnu grešku. Ovi rezultati su prikazani u glavi 4.

Glava 2

Klasterovanje sa ograničenjima

2.1 Uvod

Kod problema klasterizacije na ulazu nam je dat skup podataka (koji je u najvećem broju slučajeva skup vektora (tačaka) u d -dimenzionalnom realnom prostoru) na kome je definisana mera sličnosti između parova podataka. Na izlazu dobijamo ulazni skup podeljen u grupe zvane klasteri, tako da su elementi koji pripadaju istoj grupi slični, a elementi koji pripadaju različitim grupama su različiti.

Algoritmi klasterizacije se bave nalaženjem najboljeg načina za particiju jednog skupa podataka. Mnoge metode su bazirane na eksplicitnoj ili implicitnoj pretpostavci da klasteri formiraju konveksne regione. Međutim mi ćemo ovde analizirati drugi pristup, koji se zove spektralna klasterizacija [Ng et al., 2002, Von Luxburg, 2007, White and Smyth, 2005]. Ona pripada klasi algoritama klasterizacije na grafovima. Graf je korišćen za modeliranje ulaznog skupa tako da čvorovi modeliraju ulazne vektore a grane modeliraju meru sličnosti. Takav graf se zove graf sličnosti (poglavlje 1.2). Spektralno klasterovanje koristi spektar (sopstvene vrednosti i sopstvene vektore) matrice sličnosti grafa, čiji elementi predstavljaju sličnost između ulaznih vektora, da bi izvršilo klasterovanje grafa sličnosti. Za razliku od algoritama klasterisanja zasnovanih na Euklidskom prostoru poput algoritma k -srednjih vrednosti, algoritmi spektralnog klasterisanja mogu da formiraju klasterne proizvoljnog oblika.

Međutim, algoritmi spektralnog klasterisanja nisu naširoko prihvaćeni kao konkurenti hijerarhijskom klasterisanju ili algoritmu k -srednjih vrednosti za analiziranje velikih skupova podataka. Očigledan razlog je taj što algoritmi spektralnog klasterisanja izračunavaju sopstvene vektore matrice sličnosti i zbog toga ovi algoritmi ima kubnu vremensku složenost po broju ulaznih podataka (čvorova grafa). Još jedno pitanje koje treba razmotriti je kako omogućiti korisniku da vodi proces klasterisanja i iskoristiti (ugraditi) prethodno znanje iz domena kome pripada ulazni skup podataka (domena kome pripada problem koji se rešava) kako bi se poboljšala tačnost izlaznog skupa klastera. Algoritmi klasterisanja sa ograničenjima [Basu et al., 2008] uključuju znanje iz domena u obliku ograničenja koja nam daju informacije o tome da li neka instanca može pripadati

klasteru ili ne. Algoritmi najčešće koriste dva tipa ograničenja: ML (iz engleskog must-link constraints) ograničenja i CL (iz engleskog cannot-link constraints) ograničenja. ML ograničenja određuju skupove instanci koje bi trebalo povezati, znači koje pripadaju istom klasteru. Sa druge strane CL ograničenja su parovi instanci koje ne bi trebalo povezati, znači koje pripadaju različitim klasterima.

Algoritmi klasterisanja sa ograničenjima iz prethodnih radova se mogu podeliti u tri kategorije. U prvoj kategoriji, ograničenja se uvode promenom matrice susedstva kao u [Kamvar et al., 2003, Xu et al., 2005a, Lu and Carreira-Perpinan, 2008]. U drugoj kategoriji, ograničenja se uvode ograničavanjem prostora rešenja kao u [Coleman et al., 2008, De Bie et al., 2004, Li et al., 2009]. U trećoj kategoriji su algoritmi koji modifikuju problem u problem optimizacije sa ograničenjima kao u [Wang et al., 2014a].

Pristup, predložen u ovoj glavi, se sastoji iz dva glavna koraka. Konvertujemo ulazni skup sa ograničenjima u višeslojni graf. Višeslojni graf je struktura koja predstavlja skup grafova sa zajedničkim skupom čvorova. Svaki graf u strukturi se zove sloj. Prvi sloj je graf sličnosti, a naredni slojevi su grafovi ograničenja formirani iz podskupa skupa ograničenja. Drugi korak je spektralno klasterisanje bez ograničenja na novoformiranom višeslojnom grafu. Za spektralno klasterisanje na višeslojnom grafu koristimo algoritam dizajniran u [Dong et al., 2014]. Autori iz [Dong et al., 2014] su koristili algoritam koji pripada klasi metoda baziranih na analizi vektorskih potprostora [Turk and Pentland, 1991], zato što algoritam ugrađuje slojeve grafa u Grasmanovu mnogostrukost. Prethodni rezultati koji koriste analizu potprostora na Grasmanovim mnogostrukostima mogu se naći u [Hamm and Lee, 2008, 2009, Harandi et al., 2011].

Vremenska složenost ovih algoritama spektralnog klasterovanja sa ograničenjima kao i složenost običnog algoritma spektralnog klasterovanja bez ograničenja je $O(n^2 + t_{km})$ gde t_{km} predstavlja broj iteracija algoritma k-srednjih vrednosti i n predstavlja broj instanci u ulaznom skupu. Da bismo postigli bolju vremensku složenost, primenili smo Nistromovu aproksimaciju. U literaturi postoji nekoliko pristupa [Choromanska et al., 2013, Fowlkes et al., 2004, Li et al., 2011] koji uključuju Nistromovu metodu u spektralno klasterovanje. Međutim, samo su autori u [Choromanska et al., 2013] prikazali garancije performansi za svoj algoritam. Uključujući ovaj algoritam u naš algoritam spektralnog klasterovanja sa ograničenjima dobijamo algoritam složenosti $O(nl^2 + t_{km})$ gde je $l \ll n$ broj izabranih ulaznih instanci.

Autori u [Li et al., 2015a] su predložili algoritam za brzo spektralno klasterovanje sa ograničenjima, sa linearnom vremenskom složenošću po n . Međutim, oni pretpostavljaju da je broj ograničenja c mali $c \ll n$, što u principu može biti veliko kao n^2 . Algoritam iz [Li et al., 2015a] ima ili kvadratnu vremensku složenost po c ili kvadratnu memorijsku složenost po n , u zavisnosti od implementacije, i on je neprimenljiv za veliko n i c . Sa druge strane, naš algoritam ima linearnu vremensku i linearnu memorijsku složenost po n bez obzira na broj ograničenja. Prema našem znanju, to je jedini algoritam spektralnog klasterovanja sa ograničenjima koji koristi i tvrda i meka ograničenja i postiže ovu složenost bez obzira na veličinu skupa ograničenja.

Prilažemo sledeće rezultate:

-
- Pokazali smo da algoritam spektralnog klasterovanja sa ograničenjima može da se primeni kao algoritam spektralnog klasterovanja bez ograničenja na višeslojnom grafu gde su ograničenja predstavljena slojevima tog grafa. Vremenska složenost ovog algoritma je n na kvadrat.
 - Da bismo rešili problem vremenske složenosti, u odeljku 2.4 predstavljamo algoritam zasnovan na Nistromovoj metodi sa linearnom vremenskom i memorijskom složenošću.

2.2 Spektralno klasterovanje

U ovom odeljku ćemo ukratko objasniti algoritam spektralnog klasterovanja na grafu G . Više reči o tome je bilo u poglavlju 1.2. Neka je $G = (V, W)$ neusmereni težinski graf, sa skupom od n temena $V = \{v_1, v_2, \dots, v_n\}$ i matricom susedstva W . Konkretno, W je simetrična matrica gde je pozitivan realan broj W_{ij} težina ivice (v_i, v_j) . Ako je $W_{ij} = 0$, tada nema ivice između v_i i v_j . Dijagonalna matrica D takva da je

$$D_{ii} = \sum_{j=1}^n W_{ij}$$

se zove matrica stepena, a D_{ii} stepen čvor v_i .

U ovom algoritmu želimo da čvorove grafa podelimo u k grupa, za dati neusmereni graf sličnosti $G = (V, W)$ i broj željenih klastera k . Konkretno, cilj algoritma klasterizacije je da pronađe particiju koja minimizira normalizovani rez. Konkretno za dati podskup čvorova $A \subset V$, $cut(A, V \setminus A)$ je zbir težina ivica W_{ij} gde su $v_i \in V$ i $v_j \in V \setminus A$. Pored toga, $vol(A)$ predstavlja zbir stepena čvorova u A . Prema tome, normalizovani rez klastera (particije) skupa (A_1, A_2, \dots, A_k) je zbir

$$NCut(A_{1:k}) = \sum_{i=1}^k \frac{cut(A_i, V \setminus A_i)}{vol(A_i)}.$$

Skup klastera $A_{1:k}$, takav da $NCut(A_{1:k})$ dostiže minimum, zovemo rešenje problema $minNCut$, tj. minimiziranje reza. Prema [Von Luxburg, 2007] regularizacija ovog problema daje:

$$\begin{aligned} \min_{U \in \mathbf{R}^{n \times k}} \quad & \text{tr}(U^T L U) \\ \text{s. t.} \quad & U^T U = I. \end{aligned} \tag{2.1}$$

Matrica L se zove normalizovana Laplasova matrica ili normalizovani grafov Laplasijan i njena vrednost je:

$$L = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}.$$

Osobine 3. [Von Luxburg, 2007] Za normalizovani grafov Laplasijan L sledeći iskazi su tačni:

- L je pozitivna semidefinitna matrica,
- L ima n nenegativnih realnih sopstvenih vrednosti:

$$0 = \lambda_1 \leq \dots \leq \lambda_n,$$

- 0 je sopstvena vrednost matrice L čiji je sopstveni vektor $D^{\frac{1}{2}}\mathbf{1}$.

Rešenje problema (2.1) je matrica U , nazvana *spektralna matrica ugradnje*, čije su kolone prvih k sopstvenih vektora Laplasijana L . Pripadnosti klasterima izvedene su kao rešenje klasterovanja korišćenjem algoritma k -srednjih vrednosti gde ulaz čine vrste matrice U .

Algoritam 4 Spektralno klasterovanje.

[Ng et al., 2002]

Input: W i k ;

for all i, j **do**

if $i = j$ **then**

$D_{ii} := \sum_l w_{il}$;

else

$D_{ij} := 0$;

end if

end for

$L = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$.

Izračunati U čije su kolone prvih k sopstvenih vektora matrice L ;

Normalizovati vrste matrice U ;

Klasterovati vrste matrice U primenom algoritma k -srednjih vrednosti 1;

Output: Klasteri C_1, \dots, C_k ;

Dalje, prostor kolona matrice U koristiće se za predstavljanje grafa potprostorom prilikom klasterovanje na višeslojnom grafu kao u [Dong et al., 2014]. Prema tome graf će biti tačka u Grasmanovoj mnogostrukosti. Sada ćemo ukratko opisati Grasmanovu mnogostrukost i njene osnovne osobine.

2.2.1 Grasmanova mnogostrukost

Definicija 2.2.1. [Hamm and Lee, 2008] Grasmanova mnogostrukost $\mathcal{G}(k, n)$ je skup k -dimenzionalnih linearnih vektorskih potprostora prostora \mathbf{R}^n .

Ortonormirana matrica $Y \in \mathbf{R}^{n \times k}$ se može koristiti za predstavljanje tačke, prostora kolona $\text{span}(Y)$, u $\mathcal{G}(k, n)$. Ova vrsta predstavljanja nije jedinstvena, jer $\text{span}(Y_1) = \text{span}(Y_2)$ može biti tačno za dve različite matrice Y_1, Y_2 . Prema tome, kada kažemo da matrica Y pripada $\mathcal{G}(k, n)$, mislimo da njen prostor kolona pripada $\mathcal{G}(k, n)$. Udaljenost između dva elementa $\mathcal{G}(k, n)$ može se definisati pomoću glavnih uglova [Golub and Van Loan, 1996].

Definicija 2.2.2. Neka su Y_1 i Y_2 dve ortonormirane matrice iz $\mathbf{R}^{n \times k}$. Glavni uglovi $0 \leq \theta_1 \leq \dots \leq \theta_k \leq \frac{\pi}{2}$ između dva potprostora $\text{span}(Y_1)$ i $\text{span}(Y_2)$, definisani su

$$\cos \theta_i = \max_{u_i \in \text{span}(Y_1)} \max_{v_i \in \text{span}(Y_2)} u_i^T v_i$$

tako da važi

$$u_i^T u_i = 1,$$

$$v_i^T v_i = 1,$$

$$u_j^T u_i = 0,$$

$$v_j^T v_i = 0,$$

za svako $j \in \{1, \dots, i-1\}$.

Primetimo da θ_1 predstavlja najmanji ugao između dva jedinična vektora uzeta iz $\text{span}(Y_1)$ i $\text{span}(Y_2)$ respektivno. Ugao θ_2 je na sličan način izabran kao najmanji ugao između dva jedinična vektora uzeta iz $\text{span}(Y_1)$ i $\text{span}(Y_2)$ respektivno, s tim da je dodat jedan uslov da su ti vektori ortogonalni prethodno odabranim vektorima. Preostali uglovi se biraju slično. Glavni uglovi se mogu izračunati iz dekompozicije po singularnim vrednostima (SVD) matrice

$$Y_1^T Y_2 = U \cos \Theta V^T,$$

gde je

$$U = [u_1 \dots u_k],$$

$$V = [v_1 \dots v_k],$$

i $\cos \Theta$ je dijagonalna matrica

$$\cos \Theta = \text{diag}(\cos \theta_1, \dots, \cos \theta_k).$$

Prema [Hamm and Lee, 2008] rastojanje projekcije se definiše kao

$$d_P^2(Y_1, Y_2) = \left(\sum_{i=1}^k \sin^2 \theta_i \right),$$

što je prema [Dong et al., 2014] ekvivalentno sa

$$\begin{aligned} d_P^2(Y_1, Y_2) &= \left(\frac{\text{tr}(Y_1^T Y_1 + Y_2^T Y_2 - 2Y_1 Y_1^T Y_2 Y_2^T)}{2} \right) \\ &= \frac{1}{2} \|Y_1 Y_1^T - Y_2 Y_2^T\|_F^2. \end{aligned} \quad (2.2)$$

Algoritam 5 Spektralno klasterovanje višeslojnog grafa.

[Dong et al., 2014]

Input: m, W_1, \dots, W_m i k ;

for all $i = 1, k$ **do**

Izračunati matricu stepena D_i iz matrice W_i ;

$L_i := D_i^{-\frac{1}{2}}(D_i - W_i)D_i^{-\frac{1}{2}}$;

Izračunati U_i čije su kolone prvih k sopstvenih vektora matrice L_i ;

end for

$L_{mod} := \sum_{i=1}^m L_i - \alpha \sum_{i=1}^m U_i U_i^T$;

Izračunati U čije su kolone prvih k sopstvenih vektora matrice L_{mod} ;

Normalizovati vrste matrice U ;

Klasterovati vrste matrice U primenom algoritma k -srednjih vrednosti 1;

Output: Klasteri C_1, \dots, C_k

2.2.2 Spektralno klasterovanje višeslojnog grafa

Sada ćemo ukratko objasniti algoritam spektralnog klasterovanja (grupisanja) na višeslojnim grafovima razvijenim u [Dong et al., 2014]. Ovaj algoritam će se koristiti u našem rešenju zadatog problema spektralnog klasterovanja sa ograničenjima. Višeslojni graf koji se sastoji od m slojeva je skup od m pojedinačnih grafova $M_G = \{(G_i = (V, E_i, W_i))_{i \leq m}\}$ sa istim skupom čvorova. V je zajednički skup čvorova, E_i je skup ivica i W_i je matrica susedstva sloja i .

Za svaki sloj izračunavaju se graf Laplasijan L_i i matrica spektralnog ugrađivanja U_i . Autori iz [Dong et al., 2014] su predložili da se pronade matrica U takva da je tačka $\text{span}(U)$ što bliže moguće tačkama iz skupa $\{\text{span}(U_i)\}_{1 \leq i \leq m}$, i to tako da U čuva lokalna rastojanja između slojeva. Na kraju se izvrši algoritam klasterovanja k -srednjih vrednosti na vrstama matrice U . Da bi izračunali matricu U , predložili su sledeći problem optimizacije za klasterovanje na višeslojnom grafu:

$$\min_{\substack{U \in \mathbf{R}^{n \times k} \\ U^T U = I}} \sum_{i=1}^m \text{tr}(U^T L_i U) + \alpha [km - \sum_{i=1}^m \text{tr}(U U^T U_i U_i^T)], \quad (2.3)$$

gde je α parametar regularizacije. Ovaj problem je ekvivalentan sledećem problemu:

$$\min_{\substack{U \in \mathbf{R}^{n \times k} \\ U^T U = I}} \text{tr}(U^T (\sum_{i=1}^m L_i - \alpha \sum_{i=1}^m U_i U_i^T) U). \quad (2.4)$$

Stoga su kolone matrice rešenja U prvih k sopstvenih vektora modifikovane Laplasove matrice:

$$L_{mod} = \sum_{i=1}^m L_i - \alpha \sum_{i=1}^m U_i U_i^T. \quad (2.5)$$

Imajmo na umu da je redosled slojeva irelevantan za izvođenje matrice L_{mod} i da je zbog toga redosled slojeva nebitan za klasterisanje.

2.3 Spektralno klasterovanje sa ograničenjima

U ovom odeljku ćemo opisati naš algoritam spektralnog klasterovanja sa ograničenjima. Dat nam je graf sličnosti $G = (V, W)$, izveden iz ulaznih vektora karakteristika, i skup ograničenja \mathcal{PC} . Elementi skupa \mathcal{PC} imaju sledeći oblik

$$e = \{v_i, v_j, t, type\} :$$

1. Ograničenje e se odnosi na čvorove v_i i v_j ,
2. $type \in \{ML, CL\}$ se odnosi na tip ograničenja (dva čvora treba staviti u isti klaster ili dva čvora ne treba staviti u isti klaster),
3. Broj t je težina ograničenja. Ako za težinu t svakog ograničenja iz skupa \mathcal{PC} važi $t \in \{0, 1\}$, tada se ograničenja nazivaju *tvrda* (*hard*), ali ako se za neka ograničenja koristi težina $t \in (0, 1)$, onda se ograničenja nazivaju *meka* (*soft*).

U literaturi postoji nekoliko metoda za rešavanje ovog problema. Ovde ćemo ukratko spomenuti dve koje daju najbolje rezultate prema [Wang et al., 2014a]. U [Kamvar et al., 2003] autori su razvili algoritam koji se može primeniti samo na skup tvrdih ograničenja, koji se naziva algoritam spektralnog učenja, baziran na promeni matrice susedstva

1. Ako je $\{v_i, v_j, 1, ML\} \in \mathcal{PC}$, tada je težina ivice $W(i, j)$ postavljena na 1,
2. Ako je $\{v_i, v_j, 1, CL\} \in \mathcal{PC}$, tada je težina ivice $W(i, j)$ postavljena na 0.

Nakon toga se primenjuje spektralno klasterisanje. Kreira se matrica, čije su kolone najvećih k sopstvenih vektora matrice $(W + d_{max}I - D)/d_{max}$ gde je D dijagonalna matrica stepena i d_{max} najveći stepen u grafu, i na redovima ove matrice se izvodi algoritam klasterovanja k -srednjih vrednosti.

Autori u [Wang et al., 2014a] imaju drugačiji pristup i njihov algoritam se može primeniti i na skup podataka sa tvrdim ograničenjima i na skup podataka sa mekim ograničenjima. Njihova ideja je bila da pronađu vektor indikatora klastera kao rešenje problema optimizacije sa ograničenjima. Prvo ćemo objasniti njihov pristup na slučaju dva klastera.

Prvo se kreira matrica ograničenja Q na sledeći način:

$$Q(i, j) = \begin{cases} 1, & \text{ako } \{v_i, v_j, 1, ML\} \in \mathcal{PC} \\ -1, & \text{ako } \{v_i, v_j, 1, CL\} \in \mathcal{PC} \\ 0, & \text{u suprotnom.} \end{cases} \quad (2.6)$$

U slučaju mekih ograničenja umesto vrednosti $\{-1, 1\}$ se u matricu ograničenja stavljaju težine $\{-t, t\}$. Matrica ograničenja normalizovana je na isti način kao i graf Laplasijan $Q \leftarrow D^{-\frac{1}{2}} Q D^{-\frac{1}{2}}$. Autori iz [Wang et al., 2014a] definišu problem spektralnog klasterovanja sa ograničenjima za 2 particije kao sledeći problem optimizacije, čije je rešenje vektor indikatora klastera v :

$$\begin{aligned}
& \min_{v \in \mathbf{R}^n} v^T L v \\
& \text{tako da važi} \\
& v^T Q v \geq \alpha, \\
& \|v\|^2 = vol, \\
& v \neq D^{\frac{1}{2}} \mathbf{1},
\end{aligned} \tag{2.7}$$

gde je obim grafa vol zbir stepena njegovih čvorova

$$vol = \sum_{i=1}^n D_{ii}.$$

Za izabrano $\beta < \alpha$ može se dokazati da je rešenje v problema spektralnog klasterovanja sa ograničenjima (2.7) rešenje sledećeg generalizovanog problema sopstvenih vrednosti:

$$Lv = \lambda \left(Q - \frac{\beta}{vol} I \right) v. \tag{2.8}$$

Pored toga, prema [Wang et al., 2014a] samo generalizovani sopstveni vektori koji odgovaraju pozitivnim sopstvenim vrednostima mogu biti rešenje problema (2.7). Stoga se algoritam iz [Wang et al., 2014a] može sažeti na sledeći način:

1. Rešavanje generalizovanog problema sopstvenih vrednosti (2.8)
2. Kreiranje izvedivog skupa od generalizovanih sopstvenih vektora v koji odgovaraju pozitivnim sopstvenim vrednostima i normalizovanje tih vektora tako da važi $v^T v = vol$
3. Iz izvedivog skupa odabere se vektor $v^* = \arg \min_v v^T L v$
4. Iz vektora indikatora klastera $u^* = D^{-\frac{1}{2}} v^*$ konstruišu se klasteri.

U slučaju da imamo k klastera, a ne dva, u koraku (3) matrica

$$V^* = \arg \min_{V \in \mathbf{R}^{n \times (k-1)}} \text{tr}(V^T L V)$$

je konstruisana tako da se njene kolone biraju iz izvedivog skupa. U koraku (4) algoritam klasterovanja k -srednjih vrednosti se izvršava na redovima matrice $D^{-\frac{1}{2}} V^*$.

Mi predlažemo da transformišemo ulazni graf sličnosti $G = (W, E)$ i skup ograničenja \mathcal{PC} u višeslojni graf M_G . U svetlu ove transformacije primenjujemo algoritam spektralnog klasterovanja sa ograničenjima na ulaznom grafu sličnosti i skupu ograničenja kao algoritam spektralnog klasterovanja bez ograničenja na višeslojnom grafu M_G . Prvo ćemo konstruisati višeslojni graf koji će odgovarati našem problemu sa ograničenjima. Svaki

sloj ovog višeslojnog grafa imaće isti skup čvorova V . Očigledno koristimo graf sličnosti ulaza kao prvi sloj, stoga je njegova matrica susedstva jednaka matrici susedstva ulaznog grafa $W_1 = W$. Sledeći koraci su ugrađivanje ograničenja u naredne slojeve. Predlažemo da se dva ugrađena sloja koriste kao ugrađena ograničenja, jedan sloj za ograničenja koja se odnose na čvorove koji se moraju povezati i jedan sloj za ograničenja koja se odnose na čvorove koja se ne mogu povezati. Zbog toga konstruišemo troslojni graf u kojem je prvi sloj ulazni graf, a drugi i treći sloj predstavljaju ograničenja.

ML ograničenja ćemo kodirati sledećom matricom susedstva W_2 drugog sloja

$$W_2(i, j) = \begin{cases} 1, & \text{ako } \{v_i, v_j, 1, ML\} \in \mathcal{PC} \\ 0, & \text{u suprotnom} \end{cases} \quad (2.9)$$

a CL ograničenja sledećom matricom susedstva W_3 trećeg sloja

$$W_3(i, j) = \begin{cases} 0, & \text{ako } \{v_i, v_j, 1, CL\} \in \mathcal{PC} \\ 1, & \text{u suprotnom} \end{cases} \quad (2.10)$$

U poslednjem koraku grupišemo čvorove ovog višeslojnog grafa koristeći algoritam (5). Naš algoritam je sažet u algoritmu (6). Imajmo na umu da redosled slojeva nije važan za algoritam.

Možemo da izvučemo nekoliko zaključaka o ovom algoritmu:

1. Ovaj algoritam nije zavistan od tvrdih ograničenja. Na primer, neka je

$$c = \{v_i, v_j, t_c, type\} \in \mathcal{PC}$$

ograničenje sa ulaza. Težina ograničenja je t_c i zato kažemo da je težina ivice između čvorova v_i i v_j u sloju ograničenja jednaka t_c .

2. Budući da su ograničenja kodirana koristeći dva sloja višeslojnog grafa koji ima n čvorova, vremenska i memorijska složenost ovog algoritma zavisice samo od n , a ne i od broja ograničenja. Stoga možemo dodati onoliko ograničenja koliko želimo i to neće uticati na brzinu algoritma.
3. Kako dodajemo još ograničenja, algoritam će konvergirati prema istinskoj particiji. Kada dodamo sva ograničenja, drugi i treći sloj će predstavljati tačnu particiju i oni će imati veliki uticaj na klasterovanje.

Algoritam 6 Algoritam spektralnog klasterovanja sa ograničenjima kao spektralno klasterovanje višeslojnog grafa.

Input: W , k i \mathcal{PC} ;
 $W_1 := W$;
 $W_2 := 0$;
 $W_3 := 1$;
for all $\{v_i, v_j, 1, type\} \in \mathcal{PC}$ **do**
 if $type = ML$ **then**
 $W_2(i, j) := 1$;
 else
 $W_3(i, j) := 0$;
 end if
end for
for all $i = \overline{1, 3}$ **do**
 Izračunati matricu stepena D_i iz matrice W_i ;
 $L_i := D_i^{-\frac{1}{2}}(D_i - W_i)D_i^{-\frac{1}{2}}$;
 Izračunati U_i čije su kolone prvih k sopstvenih vektora matrice L_i ;
end for
 $L_{mod} := \sum_{i=1}^m L_i - \alpha \sum_{i=1}^m U_i U_i^T$;
 Izračunati U čije su kolone prvih k sopstvenih vektora matrice L_{mod} ;
 Normalizovati vrste matrice U ;
 Klasterovati vrste matrice U primenom algoritma k-srednjih vrednosti 1;
Output: Klasteri C_1, \dots, C_k

2.4 Novi algoritam spektralnog klasterovanja sa ograničenjima linearne vremenske i memorijske složenosti

Složenost algoritma (6) je $O(kn^2 + t_{km})$. Stoga će njegova primena na velikim grafovima biti spora. Da bi se rešio ovaj problem, treba smanjiti vremensku složenost spektralnog ugrađivanja. Ovde ćemo primeniti sličnu ideju kao u brzom spektralnog klasterisanju metodom Nistroma, implementiranoj u [Choromanska et al., 2013], kako bismo smanjili složenost našeg algoritma spektralnog klasterisanja. Uzorkovaćemo skup l kolona \mathcal{L} , zajednički za matricu susedstva svakog sloja, i upotrebiti ga za aproksimaciju grafovske Laplasove matrice C_i i iz nje izračunati aproksimativno spektralno ugrađivanje U_i za svaki sloj. Dalje, aproksimiraćemo modifikovani graf Laplasijan sa

$$\tilde{C}_{mod} = \sum_{i=1}^m C_i + \alpha \sum_{i=1}^m U_i U_i^T(:, \mathcal{L})$$

i koristiti najmanjih k sopstvenih vektora ove matrice kao ulaz algoritma klasterovanja k-srednjih vrednosti.

Sada ćemo ukratko opisati Nistromovu metodu za aproksimaciju simetrične i pozitivno semidefinitne matrice $X \in \mathbf{R}^{n \times n}$. Više reči o ovoj metodi je bilo u poglavlju 1.1.

Matrica C izvedena je iz matrice X uzorkovanjem njenih kolona. Konkretno, neka \mathcal{L} bude skup nasumično izabranih indeksa kolona. Otuda je $C = X(\mathcal{L}, :)$ i njena dimenzija je $l \times n$. Matrica Q je izvedena iz C izborom l redova na sledeći način $Q = C(:, \mathcal{L})$. Tako možemo izračunati dekompoziciju po sopstvenim vrednostima matrice

$$Q = U_Q \Sigma_Q U_Q^T,$$

jer je to simetrična matrica. Nistromova metoda generiše aproksimaciju \tilde{X} matrice X sa sledećom dekompozicijom po sopstvenim vrednostima:

$$\tilde{X} = \left(\sqrt{\frac{l}{n}} C U_Q \Sigma_Q^+ \right) \left(\frac{n}{l} \Sigma_Q \right) \left(\sqrt{\frac{l}{n}} C U_Q \Sigma_Q^+ \right)^T. \quad (2.11)$$

Složenost ove metode je $\mathcal{O}(nl^2)$. Sledeća teorema iz [Kumar et al., 2009] daje gornju granicu greške Nistromove metode:

Teorema 2.4.1. [Kumar et al., 2009] *Neka je $K \in \mathbf{R}^{n \times n}$ simetrična i pozitivno semidefinitna matrica. Pretpostavimo da se l kolona matrice K uzorkuju nasumično uniformno bez zamena, da je \tilde{K}_r Nistromova aproksimacija ranga r i da je K_r najbolja aproksimacija ranga r matrice K . Neka je*

$$\delta \in (0, 1), \quad \epsilon > 0$$

$$l \geq \frac{64r}{\epsilon^4},$$

$$\eta = \sqrt{\frac{\log(\frac{2}{\delta}) \xi(l, n-l)}{l}}$$

gde je

$$\xi(a, b) = \frac{ab}{a+b-1/2} \frac{1}{1 - \frac{1}{2 \max(a,b)}}.$$

Ako je $\gamma = \sum_{i \in D(l)} K_{ii}$ gde je $D(l)$ skup indeksa l najvećih dijagonalnih elemenata matrice K i ako je $\theta = \sqrt{n \sum_{i=1}^n K_{ii}^2}$ onda je sa verovatnoćom $1 - \delta$ sledeći izraz tačan

$$\|K - \tilde{K}_r\|_F \leq \|K - K_r\|_F + \epsilon \left[\left(\frac{n}{l} \gamma \right) \left(\theta + \eta \max_{i=1, n} K_{ii} \right) \right]^{\frac{1}{2}}. \quad (2.12)$$

Sada ćemo opisati algoritam za brzo spektralno klasterovanje grafa $G = (V, W)$ pomoću Nistromove metode iz [Choromanska et al., 2013]. Neka je \mathcal{L} skup indeksa l uzorkovanih kolona matrice susedstva W . Matrica $\hat{W} = W(:, \mathcal{L})$ je matrica susedstva sa uzorkovanim kolonama. Sada se izračunavaju dve (retke) dijagonalne matrice stepena čvorova:

$$D \in \mathbf{R}^{n \times n} \quad \text{takva da je} \quad D_{ii} = \frac{1}{\sqrt{\sum_{j=1}^l \hat{W}_{ij}}}, \quad (2.13)$$

$$\Delta \in \mathbf{R}^{l \times l} \quad \text{takva da je} \quad \Delta_{jj} = \frac{1}{\sqrt{\sum_{i=1}^n \hat{W}_{ij}}}. \quad (2.14)$$

Sledeća matrica je aproksimacija uzorkovane grafovske Laplasove matrice

$$C = I(:, \mathcal{L}) - \sqrt{\frac{l}{n}} DW \Delta.$$

Sada se matrica Q računa kao $\text{sym}(C(\mathcal{L}, :))$. U sledećem koraku se računa dekompozicija po sopstvenim vrednostima matrice Q

$$Q = U_Q \Sigma_Q U_Q^T.$$

Dijagonalna matrica sopstvenih vrednosti aproksimativne grafovske Laplasove matrice je $\tilde{\Sigma} = \frac{n}{l} \Sigma_Q$ a aproksimativna matrica sopstvenih vektora je

$$\tilde{U} = \sqrt{\frac{l}{n}} C U_Q \Sigma_Q^+.$$

Iz toga sledi da je aproksimativni grafovski Laplasijan jednak

$$\tilde{L} = \tilde{U} \tilde{\Sigma} \tilde{U}^T.$$

Matrica spektralnog ugrađivanja U računa se iz prvih (oni koji odgovaraju najmanjim sopstvenim vrednostima) k sopstvenih vektora matrice \tilde{U} i redovi matrice U se normalizuju. Kao i u uobičajenom algoritmu spektralnog klasterisanja, algoritam klasterovanja k -srednjih vrednosti se primenjuje na redove matrice U .

Opremljeni brzim algoritmom spektralnog klasterovanja, modifikovaćemo algoritam spektralnog klasterovanja za m -slojni graf $M_G = \{(G_i = (V, E_i, W_i))_{i \leq m}\}$ tako da njena vremenska složenost bude linearna po n . Prvi korak je izračunavanje aproksimacija uzorkovanih Laplasovih matrica C_i za svaki sloj i izračunavanje aproksimativnog spektralnog ugrađivanja U_i kao u [Choromanska et al., 2013]. Sledeći korak je izračunavanje aproksimacije modifikovane Laplasove matrice (2.5) na osnovu \mathcal{L} . Imajmo na umu da je skup uzorkovanih indeksa kolona \mathcal{L} jednak za sve slojeve grafa:

$$\tilde{C}_{mod} = \sum_{i=1}^m C_i + \alpha \sum_{i=1}^m U_i U_i^T(:, \mathcal{L}). \quad (2.15)$$

Budući da ne želimo da složenost našeg algoritma bude kvadratna po n , ne možemo prvo izračunati $U_i U_i^T$ a zatim uzorkovati njegove kolone, ali možemo izračunati uzorkovanu matricu odmah:

$$U_i U_i^T(:, j) = \sum_{t=1}^k U_i(:, t) U_i(\mathcal{L}_j, t). \quad (2.16)$$

Koristeći teoremu (2.4.1) izračunaćemo gornju granicu za grešku aproksimacije \tilde{L}_{mod} . Kao u [Choromanska et al., 2013] pretpostavimo da za svaki sloj grafovski Laplasijan ima vrednost 1 u svakom polju na glavnoj dijagonali i dobijamo sledeću ganicu:

Teorema 2.4.2. *Neka je $\{L_i \in \mathbf{R}^{n \times n}\}_{i=1, \dots, m}$ skup idealnih grafovskih Laplasijana svakog sloja. Pretpostavimo da se l kolona matrice L_i uzorkuju nasumično i uniformno bez zamena i da je \tilde{L}_{mod} najbolja Nistromova aproksimacija ranga r matrice L_{mod} . Neka važi*

$$\epsilon > 0,$$

$$l \geq \frac{64r}{\epsilon^4},$$

$$\eta = \sqrt{\frac{\log(\frac{2}{\delta})\xi(l, n-l)}{l}}$$

gde je $\delta \in (0, 1)$ i

$$\xi(a, b) = \frac{ab}{a+b-1/2} \frac{1}{1 - \frac{1}{2\max(a,b)}}.$$

Sa verovatnoćom $1 - \delta$ sledeći izraz je tačan:

$$\|L_{mod} - \tilde{L}_{mod}\|_F \leq \epsilon \left[\left(1 + \frac{\alpha mk}{l}\right) (\eta + \alpha \eta k + m + \frac{\alpha mk}{n}) \right] \quad (2.17)$$

Dokaz. Iz teoreme (2.4.1) imamo nejednakost:

$$\|L_{mod} - \tilde{L}_{mod}\|_F \leq \epsilon \left[\left(\frac{n}{l}\gamma\right) (\theta + \eta \max_{i=1, \dots, n} n(L_{mod})_{ii}) \right]^{\frac{1}{2}} \quad (2.18)$$

gde $\gamma = \sum_{i \in D(l)} (L_{mod})_{ii}$, $D(l)$ je skup indeksa l najvećih dijagonalnih elemenata matrice L_{mod} , i $\theta = \sqrt{n \sum_{i=1}^n (L_{mod})_{ii}^2}$.

Prvo ćemo izračunati gornje granice za γ , θ i $\max_{i=1, \dots, n} n(L_{mod})_{ii}$:

$$\begin{aligned} \gamma &= \sum_{j \in D(l)} (L_{mod})_{jj} \\ &= \sum_{j \in D(l)} \left(\sum_{i=1}^m (L_i)_{jj} + \alpha \sum_{i=1}^m (U_i U_i^T)_{jj} \right) \\ &\leq ml + \alpha \sum_{i=1}^m \text{tr}(U_i U_i^T) \leq ml + \alpha mk \end{aligned} \quad (2.19)$$

$$\begin{aligned}
 \theta &= \sqrt{n \sum_{j=1}^n (L_{mod})_{jj}^2} \leq \sum_{j=1}^n (L_{mod})_{jj} \\
 &= \sum_{j=1}^n \left(\sum_{i=1}^m (L_i)_{jj} + \alpha \sum_{i=1}^m (U_i U_i^T)_{jj} \right) \\
 &= mn + \alpha \sum_{i=1}^m \text{tr}(U_i U_i^T) \\
 &= mn + \alpha mk \tag{2.20}
 \end{aligned}$$

$$\begin{aligned}
 \max_{j=1, n} n(L_{mod})_{jj} &= n \max_{j=1, n} \sum_{i=1}^m (L_i)_{jj} + \alpha \sum_{i=1}^m (U_i U_i^T)_{jj} \\
 &\leq n + n\alpha k \tag{2.21}
 \end{aligned}$$

Primenom (2.19), (2.20) i (2.21) u (2.18) dobijamo:

$$\|L_{mod} - \tilde{L}_{mod}\|_F \leq n\epsilon \left[\left(1 + \frac{\alpha mk}{l}\right) (\eta + \alpha\eta k + m + \frac{\alpha mk}{n}) \right]^{\frac{1}{2}}. \tag{2.22}$$

□

Koristeći istu ideju u pogledu predstavljanja ograničenja slojevima grafa kao u prethodnom poglavlju, implementiramo algoritam (7) za brzo spektralno klasterovanje sa ograničenjima. Algoritam (7) prvo kodira ograničenja slojevima grafa, a zatim izvodi brzo spektralno klasterisanje višeslojnog grafa na osnovu Nistromove metode. Složenost ovog algoritma je $\mathcal{O}(nl^2 + t_{km})$ dok je složenost algoritma (6) jednaka $\mathcal{O}(n^2k + t_{km})$.

Algoritam 7 Brzi algoritam spektralnog klasterovanja sa ograničenjima.

Input: W , l , k i \mathcal{PC} ;

Uzorkovati indekse kolona \mathcal{L} ;

Izračunati $W_{1:3}$ iz ulaza;

for $i = \overline{1, 3}$ **do**

$\hat{W} = W_i(:, \mathcal{L})$;

$D := 0$;

$\Delta := 0$;

for all i **do**

$$D_{ii} := \frac{1}{\sqrt{\sum_{j=1}^l \hat{W}_{ij}}}$$

end for

for all j **do**

$$\Delta_{jj} = \frac{1}{\sqrt{\sum_{i=1}^n \hat{W}_{ij}}}$$

end for

$$C = I(:, \mathcal{L}) - \sqrt{\frac{l}{n}} DW \Delta;$$

$$Q = \text{sym}(C(\mathcal{L}, :));$$

$$(U_Q, \Sigma_Q) = \text{eig}(Q);$$

$$U_i := (\sqrt{\frac{l}{n}} C U_Q \Sigma_Q^+)(:, 1 : k);$$

$$C_i := C;$$

end for

$$C_{mod} = \sum_{i=1}^3 C_i + \alpha \sum_{i=1}^m U_i U_i^T(:, \mathcal{L});$$

Izračunati U iz C_{mod} , normalizovati pa klasterovati njene vrste;

Output: Klasteri C_1, \dots, C_k

2.5 Empirijska evaluacija algoritama spektralnog klasterovanja sa ograničenjima

U ovom odeljku procenjujemo performanse naših algoritama:

- algoritam spektralnog klasterovanja sa ograničenjima korišćenjem višeslojnog grafa (**ConstrainedSPectral-MultiLayer** (6)) koji ćemo označavati **CSP-ML**,
- i brzi algoritam spektralnog klasterovanja sa ograničenjima korišćenjem višeslojnog grafa (**FAST-ConstrainedSPectral-MultiLayer** (7)), koji ćemo označavati **FAST-CSP-ML**.

Upoređujemo ove algoritme sa sledećim algoritmima:

- algoritmom spektralnog klasterovanja sa ograničenjima (**CSP**) iz [Wang et al., 2014a],
- i algoritmom spektralnog učenja (**SL**) iz [Kamvar et al., 2003].

U Matlabu smo implementirali naše algoritme **CSP-ML** i **FAST-CSP-ML**, a kod za **CSP** algoritam je preuzet sa sajta jednog od autora [Wang, 2014]. Program smo izvršavali na mašini sa Intel Core i7-4790 CPU-om 3.60GHz, koji ima 4 jezgra i 32GB memorije, a izlaz programa su slike i tabele koje ćemo analizirati na kraja ovog poglavlja.

Tabela 2.1: Skupovi podataka na kojima ćemo evaluirati algoritme klasterovanja sa ograničenjima. Prikazujemo naziv skupa, broj primera u skupu i koliko atributa ima svaki skup.

Skup podataka	Instance	Atributi
Iris	100	4
Wine	130	13
Glass	214	9
Seeds	210	7
Ionosphere	351	34
Sylva	145 252	216
Orange	50 000	43
Sdd	58 509	49
Postures	78 095	38
Covtype	581 012	54

Koristimo 10 skupova podataka iz stvarnog sveta i svi su iz [Lichman, 2013]: Wine (poglavlje 2.5.1), Iris (poglavlje 2.5.1), Seeds (poglavlje 2.5.1 - 2.5.5), Glass (poglavlje 2.5.1 - 2.5.5), Ionosphere (poglavlje 2.5.1 - 2.5.5), Sensorless drive diagnosis data set (*Sdd*, poglavlje 2.5.6), Postures (poglavlje 2.5.6) i Covtype (poglavlje 2.5.6) osim skupa Orange¹(koristimo samo attribute koji nemaju podatke koji fale) skup podataka i Sylva¹

skup podataka. Matricu susedstva formiramo koristeći RBF kernel

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2}\right).$$

Ograničenja su izvedena iz tačnih oznaka. Na primer, biramo podskup A vektora ulaznih karakteristika i pretpostavljamo da znamo njihove oznake. Ako čvorovi $v_i, v_j \in A$ pripadaju skupu A , onda

1. ako pripadaju istoj klasi, onda dodajemo $\{v_i, v_j, t, ML\}$ skupu \mathcal{PC}
2. ako pripadaju različitim klasama, onda dodajemo $\{v_i, v_j, t, CL\}$ skupu \mathcal{PC}

Za informacije o skupovima podataka pogledajte tabelu (3.1). Rezultati su procenjeni pomoću Randovog indeksa kako je definisan u [Manning et al., 2008]:

$$RI = \frac{tp + tn}{tp + fp + tn + fn}$$

gde tp, fp, tn, fn predstavlja *true positive*, *false positive*, *true negative* i *false negative* respektivno.

U svim figurama broj na x -koordinati predstavlja poznate instance i y -koordinata predstavlja Randov indeks, osim kod eksperimenata kod koga su ograničenja meke ivice gde broj na x -koordinati predstavlja broj poznatih ivica.

2.5.1 Polu nadgledano testiranje

Počinjemo testiranje koristeći 30 primeraka sa poznatim oznakama i u svakom koraku povećavamo poznate oznake za 15. U svakom koraku izvodimo 30 testova (na svakom testu nasumično biramo poznate oznake) i prijavljujemo srednji Randov indeks. Objasnimo rezultate na slikama (2.1) i (2.2):

- Naši algoritmi **CSP-ML** (6) i **FAST-CSP-ML** (7) imaju dobar učinak na svim skupovima podataka. Oni efikasno koriste mali i veliki broj ograničenja. Sa malim brojem zadatih ograničenja algoritmi mogu da daju dobre rezultate i dosledno daju bolje rezultate kako se dodaje više ograničenja. Dalje, skup klastera izveden iz našeg algoritma konvergira u pravi skup klastera.
- Naš algoritam **CSP-ML** (6) može da nadmaši ili proizvede uporedive rezultate sa konkurentima algoritmom spektralnog klasterovanja sa ograničenjima [Wang et al., 2014a] i algoritmom spektralnog učenja [Kamvar et al., 2003] na svim skupovima podataka. Zaključujemo da je najveća razlika između našeg algoritma **CSP-ML**

¹<http://www.causality.inf.ethz.ch/activelearning.php>

i njegovih konkurenata postignuta kada je poznat veliki broj ograničenja. S druge strane, naš algoritam **FAST-CSP-ML** (7) dobija bolje rezultate od algoritma spektralnog klasterovanja sa ograničenjima samo na Glass skupu podataka i bolje rezultate od spektralnog učenja samo na skupovima podataka Glass, Wine i Ionosphere. Međutim, najveća prednost algoritma **FAST-CSP-ML** je njegova složenost $\mathcal{O}(nl^2 + t_{km})$ koja je bolja u poređenju sa složenošću ostalih algoritama (kvadratna po broju instanci).

Pored toga, u tabeli navodimo prosečni Randov indeks, njegovu standardnu devijaciju kao i prosečnu normalizovanu međusobnu informaciju (normalized mutual information), kao i standardnu devijaciju, tabele (2.2) i (2.3). Normalizovana združena količina informacija (nmi) između slučajnih promenljivih X i Y izračunava se na sledeći način

$$NMI(X, Y) = \frac{I(X; Y)}{\sqrt{H(X)H(Y)}}$$

gde je $I(X; Y)$ združena količina informacija i $H(X)$ je entropija. Zbog činjenice da je standardna devijacija mala i slična u svim preostalim eksperimentima, njihove rezultate predstavljamo samo na slikama.

Tabela 2.2: Poređenje performansi različitih algoritama spektralnog klasterovanja sa ograničenjima na sledećim skupovima podataka iz stvarnog sveta: *glass*, *seeds* i *ionosphere*. Prikazujemo prosečni Randov indeks i njegovu standardnu devijaciju .

Poznate oznake algoritam	Mera poređenja RI			
	CSP	CSP-ML	FAST-CSP-ML	SL
<i>skup podataka</i>	<i>glass</i>			
30	0.68 ± 0.12	0.55 ± 0.13	0.69 ± 0.12	0.31 ± 0.06
75	0.72 ± 0.13	0.69 ± 0.13	0.71 ± 0.13	0.31 ± 0.07
120	0.75 ± 0.14	0.80 ± 0.14	0.76 ± 0.14	0.41 ± 0.14
165	0.78 ± 0.14	0.89 ± 0.16	0.85 ± 0.15	0.58 ± 0.14
<i>skup podataka</i>	<i>seeds</i>			
30	0.87 ± 0.16	0.91 ± 0.16	0.82 ± 0.15	0.88 ± 0.16
75	0.91 ± 0.16	0.94 ± 0.17	0.83 ± 0.16	0.91 ± 0.16
120	0.93 ± 0.17	0.96 ± 0.17	0.85 ± 0.17	0.95 ± 0.17
165	0.93 ± 0.17	0.98 ± 0.18	0.91 ± 0.17	0.98 ± 0.18
<i>skup podataka</i>	<i>ionosphere</i>			
30	0.76 ± 0.15	0.72 ± 0.15	0.54 ± 0.10	0.55 ± 0.10
75	0.84 ± 0.15	0.79 ± 0.19	0.55 ± 0.11	0.55 ± 0.10
120	0.88 ± 0.16	0.90 ± 0.18	0.59 ± 0.12	0.55 ± 0.10
165	0.90 ± 0.16	0.94 ± 0.17	0.64 ± 0.14	0.55 ± 0.10

Tabela 2.3: Poređenje performansi različitih algoritama spektralnog klasterovanja sa ograničenjima na sledećim skupovima podataka iz stvarnog sveta: *glass*, *seeds* i *ionosphere*. Prikazujemo prosečnu normalizovanu združenu količinu informacija i standardnu devijaciju.

Poznate oznake algoritam	Mera poređenja NMI			
	CSP	CSP-ML	FAST-CSP-ML	SL
<i>skup podataka</i>	<i>glass</i>			
30	0.36 ± 0.08	0.35 ± 0.08	0.28 ± 0.06	0.12 ± 0.02
75	0.39 ± 0.10	0.49 ± 0.10	0.33 ± 0.08	0.12 ± 0.02
120	0.44 ± 0.11	0.61 ± 0.11	0.46 ± 0.09	0.32 ± 0.20
165	0.53 ± 0.11	0.76 ± 0.14	0.62 ± 0.12	0.55 ± 0.14
<i>skup podataka</i>	<i>seeds</i>			
30	0.71 ± 0.15	0.76 ± 0.14	0.53 ± 0.14	0.71 ± 0.13
75	0.75 ± 0.16	0.81 ± 0.15	0.57 ± 0.14	0.78 ± 0.14
120	0.81 ± 0.15	0.87 ± 0.16	0.65 ± 0.15	0.85 ± 0.16
165	0.83 ± 0.15	0.92 ± 0.17	0.75 ± 0.14	0.92 ± 0.17
<i>skup podataka</i>	<i>ionosphere</i>			
30	0.46 ± 0.11	0.36 ± 0.18	0.18 ± 0.04	0.09 ± 0.02
75	0.58 ± 0.12	0.52 ± 0.22	0.19 ± 0.07	0.08 ± 0.02
120	0.64 ± 0.12	0.68 ± 0.18	0.21 ± 0.08	0.07 ± 0.02
165	0.71 ± 0.13	0.80 ± 0.15	0.31 ± 0.09	0.06 ± 0.02

2.5.2 Nedosledna ograničenja

Ako ne postoji dodeljivanje klastera koje zadovoljava čitav skup ograničenja \mathcal{PC} , tada skup ograničenja \mathcal{PC} sadrži nedosledna (kontradiktorna) ograničenja. Na primer: nedoslednost se može pojaviti zbog grešaka u skupu oznaka. U ovoj tezi razmotrićemo sledeću vrstu nedoslednih (kontradiktornih) ograničenja. Neka su a, b i c instance iz ulaznog skupa podataka. Sledeći skup ograničenja je nedosledan:

$$\{ML(a, b), ML(b, c), CL(a, c)\}.$$

Tri skupa podataka (Glass, Seeds i Ionosphere) koriste se za testiranje u ovom poglavlju. Počinjemo sa 30 primeraka sa poznatim oznakama i u svakom koraku povećavamo poznate oznake za 15. U svakom koraku (broj na x osi označava korak) izvodimo 30 testova (u svakom testu nasumično biramo x poznatih oznaka i nasumično biramo 0,08 x neusaglašenih ograničenja) i prijavljujemo srednji Randov indeks. Objasnimo rezultate sa slike (2.3):

- U sva tri skupa podataka naši algoritmi **CSP-ML** i **FAST-CSP-ML** su sposobni da efikasno koriste kontradiktorna ograničenja. U stanju su da koriste mali broj

ograničenja, a dodavanje ograničenja poboljšava rezultate. **CSP-ML** je u stanju da konvergira ka pravoj particiji klastera.

- Algoritam **CSP-ML** ili nadmašuje konkurente ili daje nešto slabije rezultate. S druge strane, **FAST-CSP-ML** daje najbolje rezultate na skupu podataka Glass, lošije na Seeds skupu podataka i bolje rezultate nego algoritam spektralnog učenja, ali gore od algoritma spektralnog klasterovanja sa ograničenjima **CSP** na Ionosphere skupu podataka.

2.5.3 Skup ograničenja sa 30% šuma

U ovom slučaju pretpostavljamo da skup ograničenja ima 30% šuma. Ograničenje $CL(a, b)$ je šum ako instance a i b pripadaju istom klasteru. Koristimo tri skupa podataka (Glass, Seeds i Ionosphere) u ovom slučaju.

Počinjemo sa 30 primeraka sa poznatim oznakama i u svakom koraku povećavamo poznate oznake za 15. U svakom koraku (broj na x osi označava korak) izvodimo 30 testova (u svakom testu nasumično biramo x poznatih oznaka od kojih kreiramo 70% tačnih ograničenja i 30% netačnih ograničenja) i prijavljujemo srednji Randov indeks. Objasnimo rezultate sa slike (2.4):

- U sva tri skupa podataka naš algoritam **CSP-ML** je u stanju da efikasno koristi ograničenja sa 30% šuma i konvergira ka pravoj particiji. S druge strane, **FAST-CSP-ML** obično daje iste rezultate bez obzira na broj ograničenja.
- Algoritam **CSP-ML** nadmašuje konkurente u sva tri skupa podataka. Primetili smo da na skupu podataka Ionosphere (redak graf) **CSP-ML** nadmašuje konkurente i algoritam **FAST-CSP-ML** sa velikom razlikom. S druge strane, **FAST-CSP-ML** daje bolje rezultate od spektralnog algoritma učenja, ali lošije od algoritma spektralnog klasterovanja sa ograničenjima na sva tri skupa podataka.

2.5.4 Meka ograničenja

Do sada smo izvodili testove samo na skupovima sa tvrdim ograničenjima, ali ovde ćemo raditi eksperimente na primerima sa mekim ograničenjima. Koristimo tri skupa podataka (Glass, Seeds i Ionosphere) u ovom slučaju. Budući da algoritam spektralnog učenja ne može da koristi meka ograničenja, upoređujemo naše algoritme **CSP-ML** i **FAST-CSP-ML** samo sa algoritmom spektralnog klasterovanja sa ograničenjima [Wang et al., 2014a].

Počinjemo sa 30 primeraka sa poznatim oznakama i u svakom koraku povećavamo poznate oznake za 15. U svakom koraku (broj na x osi označava korak) izvodimo 30 testova (u svakom testu nasumično biramo x poznatih oznaka od kojih kreiramo ograničenja sa stepenom tačnosti između 0,7 i 1,0) i prijavljujemo srednji Randov indeks. Sa slike (2.5) zaključujemo da:

- U sva tri skupa podataka naši algoritmi **CSP-ML** i **FAST-CSP-ML** su u stanju da efikasno koriste meka ograničenja. U stanju su da koriste mali broj ograničenja, a dodavanje ograničenja poboljšava rezultate. Dodatno, **CSP-ML** je u stanju da konvergira ka pravoj particiji klastera.
- Algoritam **CSP-ML** daje nešto bolje rezultate od svojih konkurenata u svim skupovima podataka, izuzev kod malog broja poznatih oznaka na skupovima podataka *Glass* i *Ionosphere*. S druge strane, **FAST-CSP-ML** daje slabije rezultate od svojih konkurenata na *Seeds* i *Ionosphere* skupovima podataka i slične rezultate najboljem konkurentu na *Glass* skupu.

2.5.5 Meka ograničenja na granama

Do sada smo testirali na skupovima sa ograničenjima indukovanim iz skupa poznatih oznaka na čvorovima. Kada smo saznali oznaku čvora onda smo izračunali za svaku granu tip ograničenje: ML ili CL. Međutim, u ovom slučaju ćemo nasumično odabrati skup ograničenja iz skupa svih ograničenja (skupa svih grana). Pored toga, sva ograničenja će biti meka. Zbog toga, s obzirom da algoritam spektralnog učenja ne može da koristi meka ograničenja, upoređujemo naše algoritme **CSP-ML** i **FAST-CSP-ML** samo sa algoritmom spektralnog klasterovanja sa ograničenjima [Wang et al., 2014a].

Svako ograničenje će biti upareno sa slučajno izabranim stepenom između 0,7 i 1,0. X -koordinata će predstavljati broj poznatih ograničenja. Y -koordinata će predstavljati Randov indeks. Pogledajmo rezultate sa slike (2.6):

- U sva tri skupa podataka naš algoritam **CSP-ML** je u stanju da efikasno koristi meka ograničenja na ivicama. Ovaj algoritam efikasno koristi mali broj ograničenja, a dodavanje ograničenja poboljšava rezultate. Dodatno, **CSP-ML** je u stanju da konvergira ka pravom skupu klastera.
- Algoritam **CSP-ML** nadmašuje svoje konkurente ili daje uporedivo slične rezultate. Konkretno na *Ionosphere* skupu podataka nadmašuje konkurente sa velikom razlikom. S druge strane, **FAST-CSP-ML** daje bolje rezultate od algoritma spektralnog klasterovanja sa ograničenjima na *Seeds* skupu podataka i slične rezultate na *Glass* i *Ionosphere* skupovima podataka.

2.5.6 Veliki ulazni skup

U ovom odeljku testiramo svoje algoritme na velikim skupovima podataka. Prvo, empirijski pokazujemo prednosti koje naši algoritmi pokazuju na *covtype* [Lichman, 2013] skupu podataka. *Covtype* je veliki skup podataka koji sadrži 581 012 instanci sa 54 atributa koji pripadaju jednoj od sedam različitih klasa. U našim eksperimentima koristili smo samo 10 kvantitativnih atributa sa najvećom varijansom u ulaznom skupu. Ovaj skup podataka ćemo nazvati *covtype10*.

Tabela 2.4: Performanse algoritma **FAST-CSP-ML** na velikim skupovima podataka sa dve različite vrste ograničenja (meko bez šuma i meko sa 30 % šuma). Izveštavamo o prosečnom randovom indeksu i prosečnom vremenu izvršenja programa u sekundama izračunatom iz rezultata 20 izvedenih testova.

Poznate oznake (%)	Ograničenja sa šumom			
	No		Yes	
	RI	Time (s)	RI	Time (s)
<i>skup podataka</i>	<i>covtype</i>			
10%	0.48 ± 0.01	20.23	0.46 ± 0.01	22.32
30%	0.47 ± 0.02	23.42	0.49 ± 0.01	21.11
50%	0.52 ± 0.05	21.34	0.51 ± 0.04	23.19
70%	0.52 ± 0.05	22.11	0.53 ± 0.05	23.09
90%	0.65 ± 0.06	21.51	0.54 ± 0.03	21.78
<i>skup podataka</i>	<i>sdd</i>			
10%	0.83 ± 0.01	2.41	0.84 ± 0.01	2.15
30%	0.84 ± 0.01	2.39	0.84 ± 0.01	2.41
50%	0.87 ± 0.01	2.23	0.84 ± 0.01	3.74
70%	0.90 ± 0.01	2.20	0.84 ± 0.01	5.75
90%	0.96 ± 0.01	1.86	0.85 ± 0.01	7.43
<i>skup podataka</i>	<i>postures</i>			
10%	0.70 ± 0.01	2.33	0.68 ± 0.01	2.15
30%	0.70 ± 0.01	2.36	0.69 ± 0.01	2.20
50%	0.74 ± 0.02	2.19	0.69 ± 0.01	2.85
70%	0.80 ± 0.03	2.05	0.70 ± 0.01	3.49
90%	0.87 ± 0.03	1.79	0.72 ± 0.01	4.47

Zbog memorijske složenosti od $O(n^2)$ implementacije algoritama **CSP-ML**, **CSP** i **SL** zahtevaju najmanje 2TB memorije. S druge strane, algoritam **FAST-CSP-ML** ne bi imao takvih problema jer je njegova memorijska složenost jednaka $O(kn)$. Zbog toga smo, da bismo uporedili ove algoritme, koristili podskup skupa *covtype10* od 2100 instanci (tačno 300 pripadaju svakoj od sedam različitih klasa) koji su nasumično izabrani od ukupno 581 012 instanci. Dalje, testove smo vršili sa nedoslednim (poglavlje 2.5.2), mekim (poglavlje 2.5.4) i mekim po granama (poglavlje 2.5.5) ograničenjima i uporedili smo performanse algoritama i njihovo vreme izvršenja. Pogledajmo rezultate sa slika (2.7, 2.8 i 2.9):

- Na skupu sa nedoslednim ograničenjima oba naša algoritma **CSP-ML** i **FAST-CSP-ML** su dobila najbolje rezultate bez obzira na veličinu skupa ograničenja.
- Na skupu sa mekim ograničenjima sva tri algoritma (**SL** algoritam nije primenljiv na skup sa mekim ograničenjima) su dobila slične rezultate, mada je **CSP** algoritam bio nešto bolji od svojih konkurenata.
- Na skupu sa mekim ograničenjima po granama **FAST-CSP-ML** i **CSP** algoritmi dobili su slične rezultate dok je **CSP-ML** algoritam dobio najgore rezultate na

malom skupu ograničenja i najbolje rezultate na velikom skupu ograničenja.

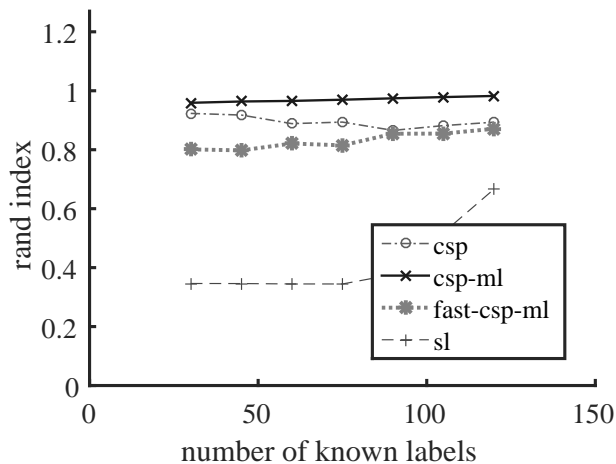
- U svakom testu i sa bilo kojom od tri vrste ograničenja **CSP** je mnogo sporiji u poređenju sa ostalim algoritmima.
- Prednost linearne vremenske i memorijske složenosti algoritma **FAST-CSP-ML** postaje očigledna kada se grupiše celokupan skup podataka *covtype*.

Izvršili smo testove na celokupnim *covtype* i *covtype10* skupovima podataka kao i na drugim velikim skupovima podataka: *sdd*, *postures*, *orange* i *sylva*. Kao što je već pomenuto samo **FAST-CSP-ML** algoritam je primenljiv na tako velikim skupovima podataka. Autori u [Li et al., 2015a] su razvili aproksimativnu verziju **CSP** algoritma zasnovanu na retkom kodiranju (eng. *sparse coding*). Međutim, njihov algoritam zahteva matricu ograničenja veličine $O(n^2)$ što je nerešivo kada je broj ograničenja veliki. S druge strane, vremenska složenost i memorijska složenost našeg algoritma su linearne po veličini skupa podataka bez obzira na broj ograničenja koji može biti n^2 .

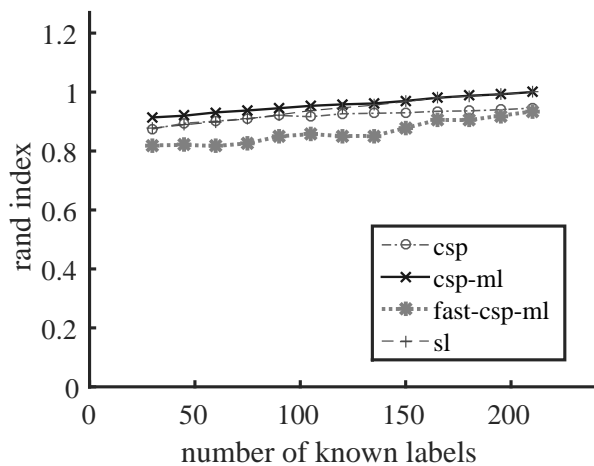
Izvršili smo testove na skupovima podataka sa mekim ograničenjima bez šuma i sa 30% šuma i na skupu sa mekim ograničenjima po granama. **FAST-CSP-ML** algoritam je primenjen, a rezultati su prikazani u sledećim tabelama (2.4 i 2.5). Primećujemo da je algoritam u stanju da efikasno koristi ograničenja i kako dodajemo ograničenja, rezultati se poboljšavaju.

Tabela 2.5: Performanse algoritma **FAST-CSP-ML** na velikim skupovima podataka sa dve različite vrste ograničenja (meko bez šuma i meko sa 30 % šuma). Izveštavamo o prosečnom randovom indeksu i prosečnom vremenu izvršenja programa u sekundama izračunatom iz rezultata 20 izvedenih testova.

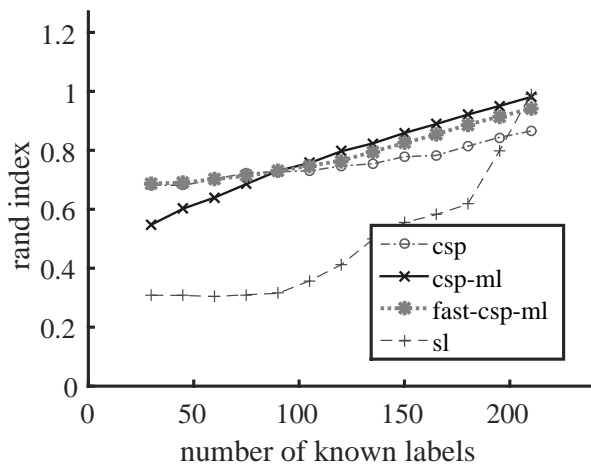
Poznate oznake (%)	Ograničenja sa šumom			
	Ne		Da	
	RI	Time (s)	RI	Time (s)
<i>skup podataka</i>	<i>covtype10</i>			
10%	0.59 ± 0.01	27.93	0.58 ± 0.01	26.27
30%	0.59 ± 0.01	27.03	0.59 ± 0.01	30.85
50%	0.62 ± 0.03	29.39	0.59 ± 0.01	33.11
70%	0.68 ± 0.04	34.03	0.59 ± 0.01	29.49
90%	0.70 ± 0.04	27.90	0.58 ± 0.01	32.65
<i>skup podataka</i>	<i>orange</i>			
10%	0.50 ± 0.01	1.06	0.50 ± 0.01	0.89
30%	0.54 ± 0.01	0.97	0.51 ± 0.01	1.02
50%	0.62 ± 0.01	0.97	0.56 ± 0.02	1.31
70%	0.74 ± 0.01	0.93	0.66 ± 0.02	1.54
90%	0.89 ± 0.01	0.91	0.74 ± 0.02	2.55
<i>skup podataka</i>	<i>sylva</i>			
10%	0.51 ± 0.01	1.13	0.50 ± 0.01	1.12
30%	0.54 ± 0.03	1.11	0.50 ± 0.01	1.26
50%	0.64 ± 0.02	1.16	0.52 ± 0.01	1.29
70%	0.73 ± 0.01	1.18	0.58 ± 0.03	1.39
90%	0.91 ± 0.01	1.18	0.67 ± 0.04	1.52



(a) Wine

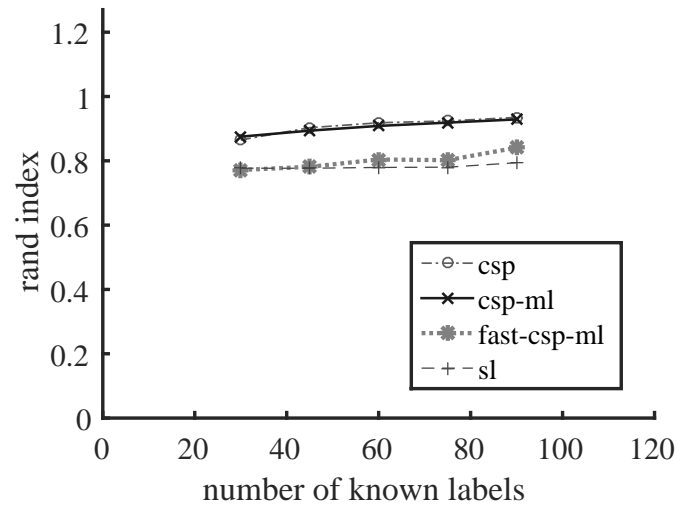


(b) Seeds

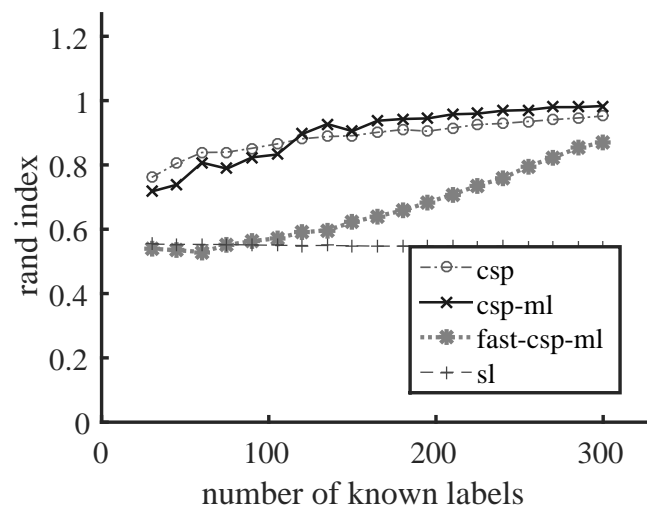


(c) Glass

Slika 2.1: Poređenje performansi različitih algoritama spektralnog klasterovanja sa ograničenjima na skupovima podataka iz stvarnog sveta za različit broj poznatih labela koji je dat na x -osi.

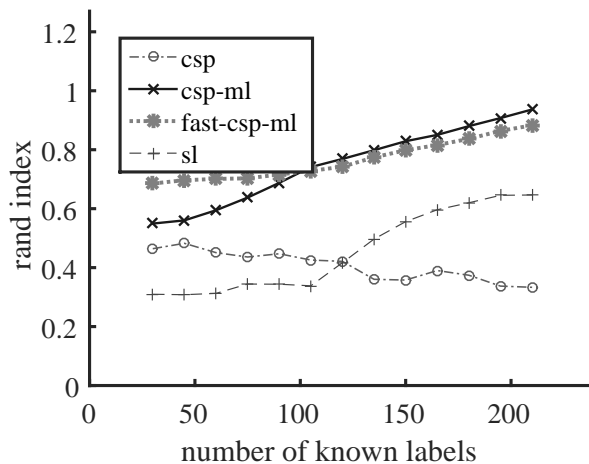


(a) Iris

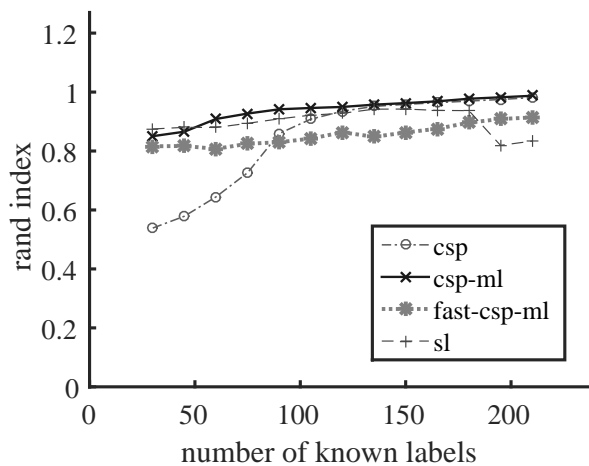


(b) Ionosphere

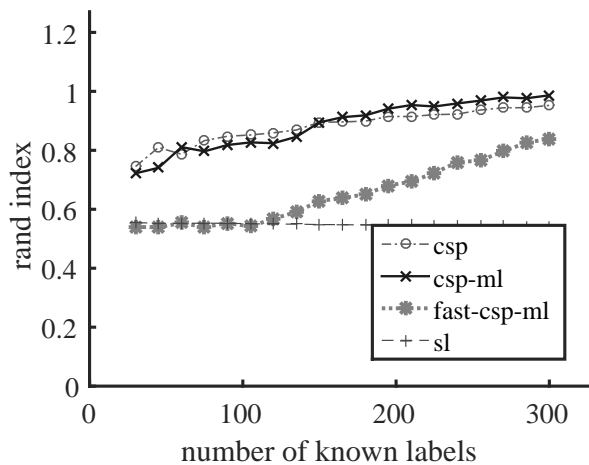
Slika 2.2: Poređenje performansi različitih algoritama spektralnog klasterovanja sa ograničenjima na skupovima podataka iz stvarnog sveta za različit broj poznatih labela koji je dat na x -osi.



(a) Glass

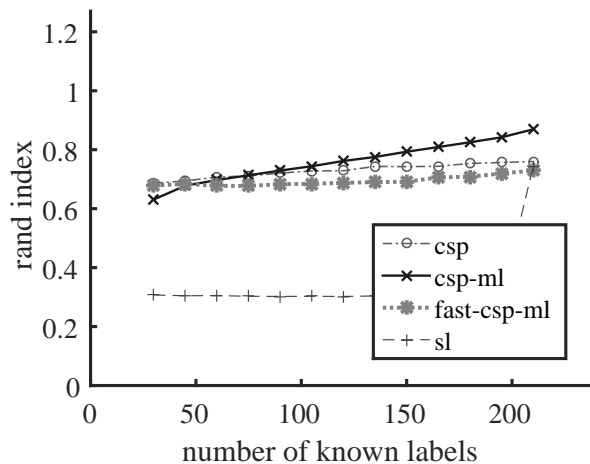


(b) Seeds

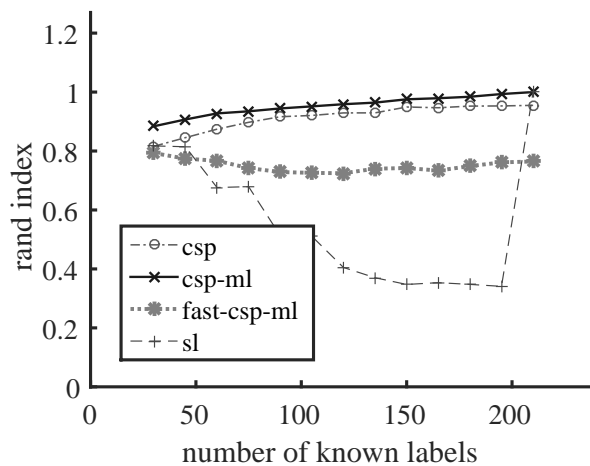


(c) Ionosphere

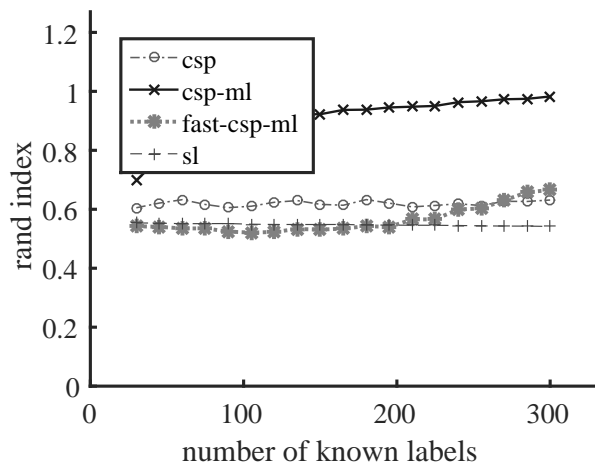
Slika 2.3: Poređenje performansi različitih algoritama spektralnog klasterovanja sa ograničenjima na skupovima podataka iz stvarnog sveta sa nedoslednim (kontradiktornim) ograničenjima za različit broj poznatih labela koji je dat na x -osi.



(a) Glass

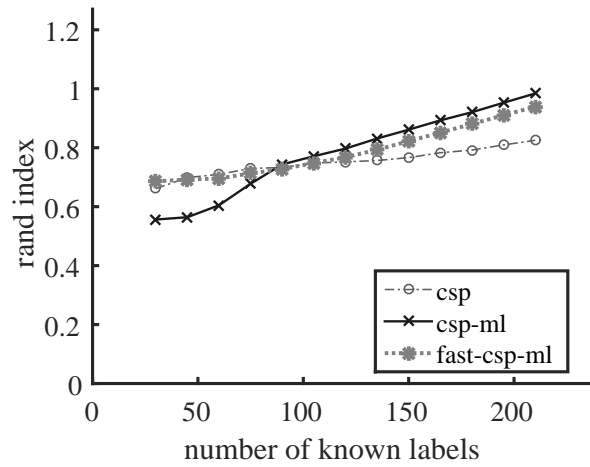


(b) Seeds

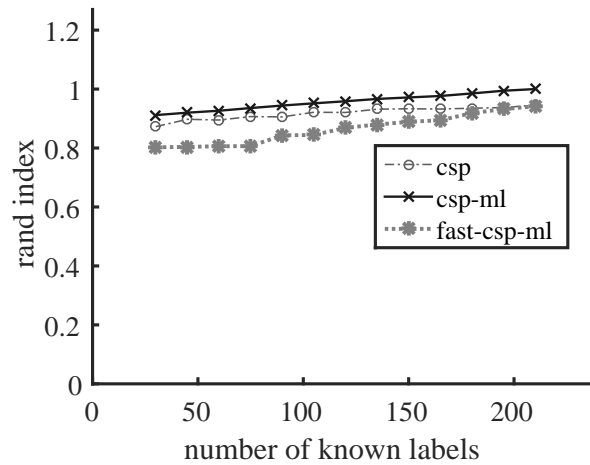


(c) Ionosphere

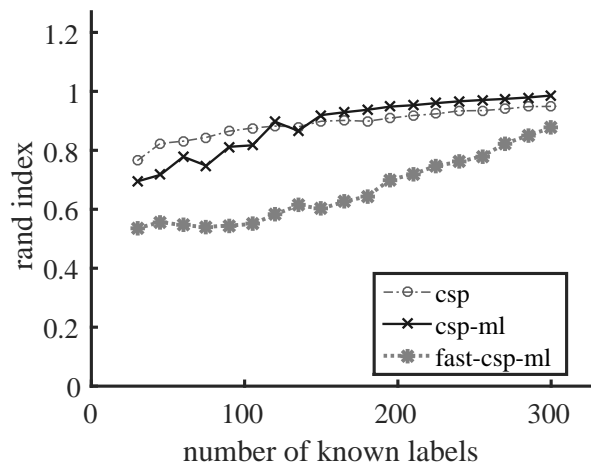
Slika 2.4: Poređenje performansi različitih algoritama spektralnog klasterovanja sa ograničenjima na skupovima podataka iz stvarnog sveta, gde skup ograničenja sadrži šum, za različit broj poznatih labela koji je dat na x -osi.



(a) Glass

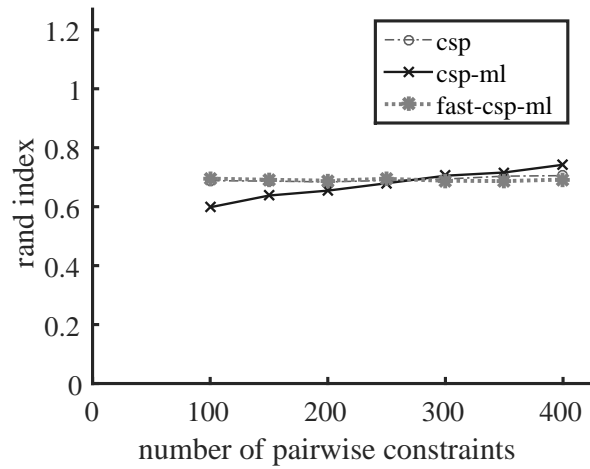


(b) Seeds

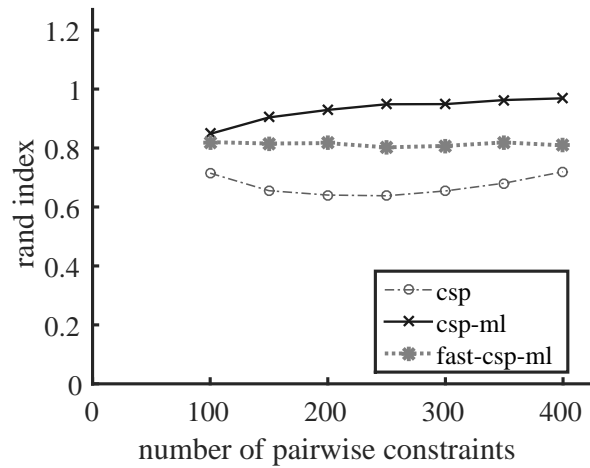


(c) Ionosphere

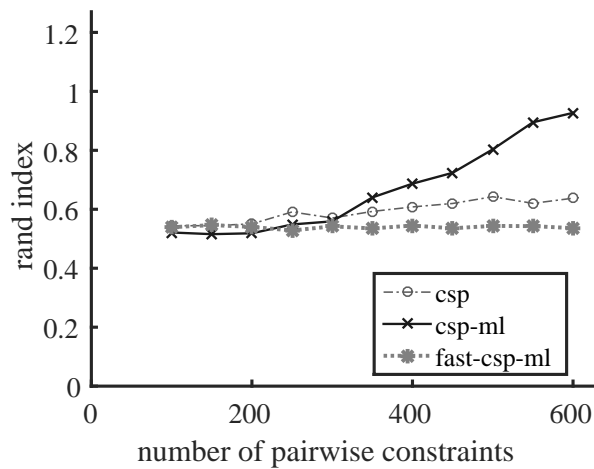
Slika 2.5: Poređenje performansi različitih algoritama spektralnog klasterovanja sa ograničenjima na skupovima podataka iz stvarnog sveta, gde skup ograničenja sadrži meka ograničenja za različit broj poznatih labela koji je dat na x -osi.



(a) Glass

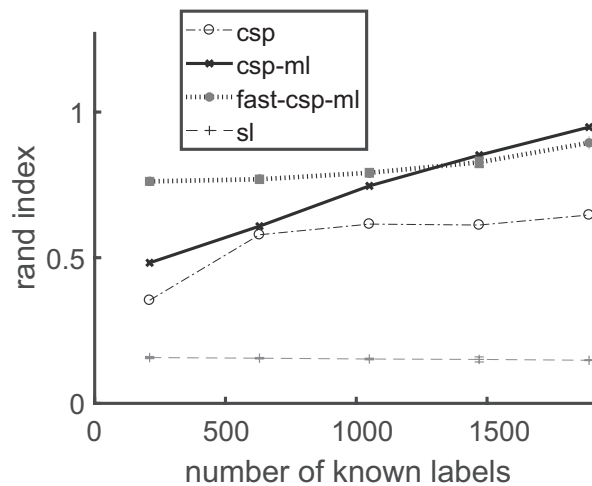


(b) Seeds

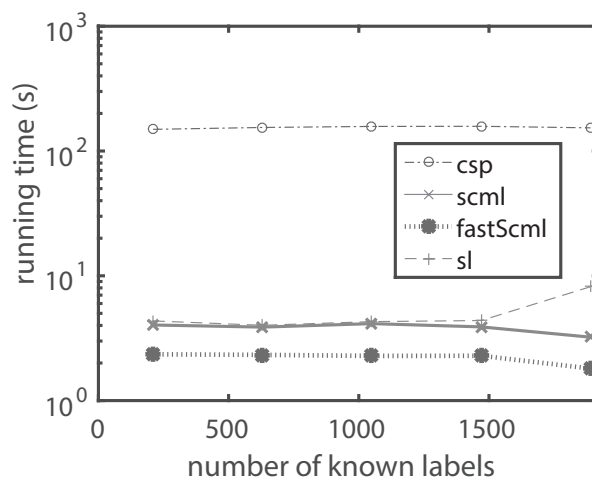


(c) Ionosphere

Slika 2.6: Poređenje performansi različitih algoritama spektralnog klasterovanja sa ograničenjima na skupovima podataka iz stvarnog sveta, gde skup ograničenja sadrži meka ograničenja po ivicama za različit broj parova ograničenja koji je dat na x -osi.

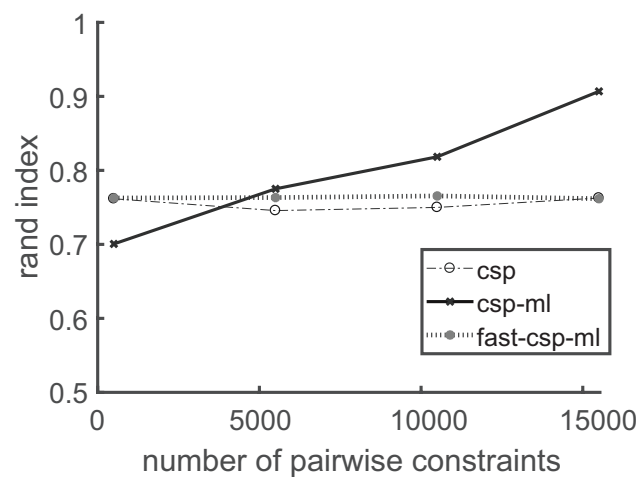


(a) Inconsistent

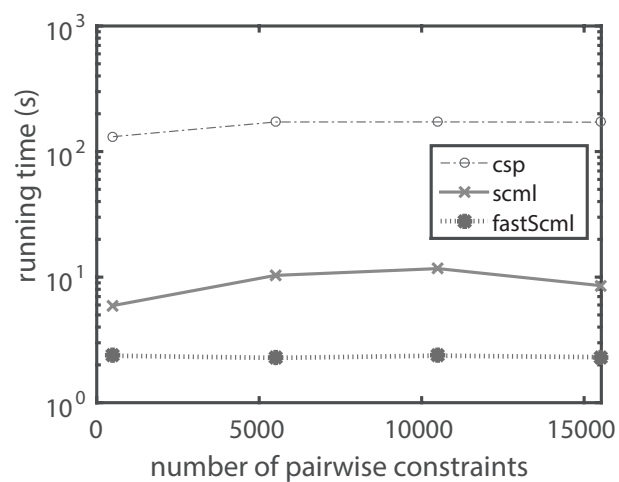


(b) Inconsistent

Slika 2.7: Poređenje performansi i vremena izvršenja različitih algoritama spektralnog klasterovanja sa ograničenjima na podskupu skupa podataka covtype sa tri različita tipa ograničenja (nedosledna, meka i meka po granama). Vreme rada je predstavljeno sa logaritamskom skalom. Broj poznatih labela je dat na x -osi.

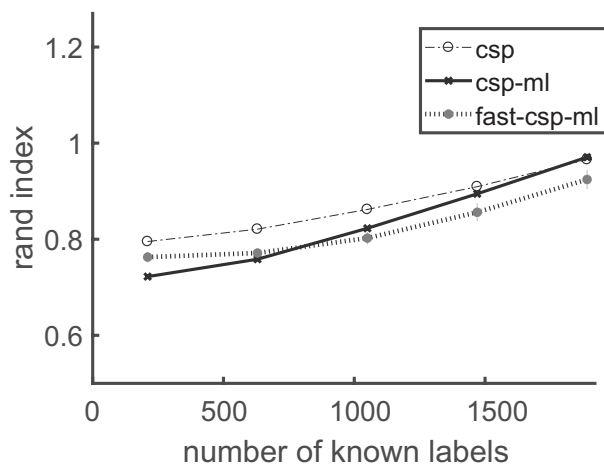


(a) Soft Edge

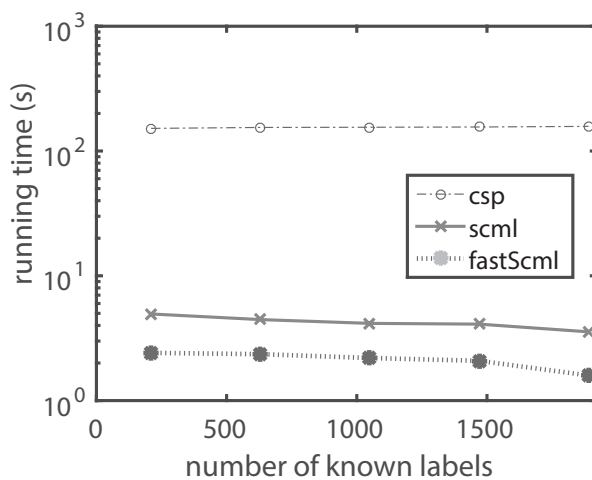


(b) Soft Edge

Slika 2.8: Poređenje performansi i vremena izvršenja različitih algoritama spektralnog klasterovanja sa ograničenjima na podskupu skupa podataka covtype sa tri različita tipa ograničenja (nedosledna, meka i meka po granama). Vreme rada je predstavljeno sa logaritamskom skalom. Broj poznatih labela je dat na x -osi.



(a) Soft



(b) Soft

Slika 2.9: Poređenje performansi i vremena izvršenja različitih algoritama spektralnog klasterovanja sa ograničenjima na podskupu skupa podataka covtype sa tri različita tipa ograničenja (nedosledna, meka i meka po granama). Vreme rada je predstavljeno sa logaritamskom skalom. Broj poznatih labela je dat na x -osi.

Glava 3

Brzo spektralno klasterovanje pomoću samoučenja

3.1 Uvod i prethodni rezultati

Ponovimo ukratko prednosti i mane spektralnog klasterovanja (poglavlje 1.2). Sa pozitivne strane algoritam spektralnog klasterisanja nudi nekoliko prednosti u poređenju sa ostalim algoritmima klasterisanja:

- jednostavna funkcija cene sa analitičkim rešenjem,
- laka implementacija,
- sposobnost izvođenja klastera proizvoljnih oblika.

Međutim, sa negativne strane, spektralno klasterovanje ima sledeća ograničenja:

- vremenska složenost po broju primera,
- činjenica da se u velikoj meri i isključivo oslanja na kvalitet mere sličnosti.

Predloženo je nekoliko pristupa za rešavanje tih problema i neka od njih smo već analizirali u prethodnoj glavi 2. U cilju smanjenja vremenske složenosti primena metode Nistroma [Kumar et al., 2009] je predložena. Nistromova metoda je tehnika aproksimacije matrice koja može ubrzati izračunavanje spektra matrice, što smo i pokazali u prethodnom poglavlju. Uključivanje znanja o pripadnosti klasterima (poput korisničkog vodiča - eng. *user guide* koje zovemo ekspertsko znanje - eng. *Oracle knowledge*) može smanjiti oslanjanje na meru sličnosti. Ovo je proučavano u literaturi i u prethodnoj glavi pod definicijom algoritam spektralnog klasterovanja sa ograničenjima. Ograničenja se mogu kodirati kao (ML) indikator pripadnosti istim klasterima ili (CL) indikator pripadnosti različitim klasterima. Imajmo na umu da ugrađivanjem ograničenja u algoritam spektralnog klasterovanja on više ne pripada klasi algoritama učenja bez nadzora, već pripada klasi polunadgledanih algoritama učenja.

Međutim, pristup dobrom ekspertskom znanju je obično redak. Zbog toga se razvijaju metode aproksimacije tog znanja, i takve metode će se koristiti za učenje dodatnih ograničenja. One pripadaju klasi algoritama za klasterovanje korišćenjem samoučenja. U ovoj tezi razmotrićemo brzo spektralno klasterovanje sa samoučenjem sa malom vremenskom složenošću. Naš algoritam iterativno uči nova ograničenja i koristi ih za poboljšanje klasterovanja. Prema našim saznanjima, ovo je prvi algoritam spektralnog klasterisanja sa samoučenjem s vremenskom složenošću linearnom po broju predmeta bez obzira na broj ograničenja.

Spektralno klasterovanje sa ograničenjima ima dobre performanse ako imamo pristup dobrim ograničenjima. Međutim, ponekad nemamo pristup ograničenjima, ali imamo pristup ekspertskom znanju (korisnički vodič). Algoritam može odabrati parove primera za koje je najmanje pouzdan i pitati korisnika da li ti primeri pripadaju istoj klasi ili ne. Ova vrsta algoritma naziva se aktivno spektralno klasterovanje [Wang and Davidson, 2010, Wauthier et al., 2012, Xu et al., 2005b]. Problem sa algoritmom aktivnog klasterovanja je pristup korisniku ili nekoj bazi korektnih informacija, jer određene informacije nisu uvek lako dostupne. Dakle, pitanje je da li bi sam algoritam mogao da predstavlja neku aproksimaciju korisnika i da li bi algoritam mogao da nauči dodatna ograničenja. Ovaj problem se naziva samoučenje [Dai et al., 2008, Raina et al., 2007], a pojam je nastao kod istraživanja algoritama transfera znanja (posebne klase algoritama mašinskog učenja) [Pan and Yang, 2009]. Algoritam u [Raina et al., 2007] uvećava ulazni skup velikim brojem neobebeženih primera radi poboljšanja klasifikacije. Razlikuje se od polunadgledanog učenja time što ti neobebeženi primeri nisu nužno uzorkovani iz iste slučajne raspodele. Algoritam u [Dai et al., 2008] sledi iste principe, sa fokusom na klasterovanje. U [Wang et al., 2014b] spektralno klasterovanje koje se samostalno uči definiše se kao kombinacija algoritma spektralnog klasterovanja sa ograničenjima iz [Wang et al., 2014a], i učenja novih ograničenja iz starih ograničenja i matrice susedstva putem problema upotpunjavanja matrice. To je izvodivo jer su autori u [Wang et al., 2014b] pretpostavili da matrica ograničenja ima mali rang. Ovaj algoritam pripada klasi algoritama za samoučenje, jer koristi znanje o matrici susedstva i znanje o prethodnim ograničenjima da bi naučio nova ograničenja. Stoga algoritam samoobučava skup ograničenja, koji se zauzvrat koristi za izvođenje spektralnog klasterovanja. Međutim, vremenska i memorijska složenost ovog algoritma je kvadratna po broju primera i stoga je nerešiva za velike skupove podataka. Mi ćemo definisati algoritam spektralnog klasterovanja koji se samostalno uči i koji je primenljiv na velikim skupovima podataka.

U ovoj tezi razmotrićemo problem spektralnog klasterovanja sa samoučenjem i naš algoritam biće nastavak algoritma opisanog u prethodnoj glavi 2. Odabrali smo ovaj algoritam zbog nekoliko prednosti koje ima u poređenju sa konkurentima:

- Rešava problem spektralnog klasterovanja sa ograničenjima, s vremenskom i memorijskom složenošću koja je linearna u n , bez obzira na broj ograničenja.
- Algoritam jednostavno ugrađuje skup ulaza i ograničenja u troslojni graf. Međutim, ovo se lako može proširiti na graf sa više od tri sloja, što može kodirati dodatne informacije. To je ideja koju ćemo koristiti.

- Funkcija cene je jednostavna, ima analitičko rešenje i lako i brzo se implementira.

Naš algoritam se može primeniti u dve različite postavke:

1. Skup ulaza sadrži samo skup uzoraka i funkciju sličnosti (standardno nenadgledano podešavanje).
2. Ulaz sadrži skup od n primera, meru sličnosti i skup ograničenja.

Bez obzira na skup ograničenja, naš algoritam iterativno uči nova ograničenja i koristi ih za poboljšanje rezultujućeg skupa klastera. Ta nova ograničenja su zapravo izvedena iz rezultata klasterovanja u prethodnoj iteraciji. Ukratko, spektralno klasterovanje sa samoučenjem izvodiće se na sledeći način:

1. Primeniti algoritam spektralnog klasterovanja sa ograničenjima ili algoritam običnog spektralnog klasterovanja na ulaznom skupu.
2. Povećati ulazni skup novim skupom ograničenja, gde su ML i CL ograničenja izvedena iz rezultata prethodnog koraka.
3. Izvršiti prva dva koraka t puta, gde je broj t hiper-parametar algoritma koji je izabran na početku.

Glavne prednosti našeg algoritma i samim tim glavni doprinos ove glave su sledeće:

1. Brz, lako primenljiv algoritam spektralnog klasterovanja sa samoučenjem koji je primenljiv na velikim skupovima podataka sa vremenskom složenošću od $O(t(nk^2 + t_{km}))$. Prema našim saznanjima, ovo je jedini algoritam spektralnog klasterovanja sa samoučenjem linearne složenosti.
2. U stanju je da uključi rešenja algoritma klasterovanja (iz prethodnog koraka) kao ograničenja u sledećem koraku spektralnog klasterovanja.

3.2 Algoritam

U ovom odeljku ćemo objasniti našu formulaciju spektralnog klasterisanja koji se samostalno uči i prikazati naš algoritam koji rešava ovaj problem. Naš algoritam je proširenje algoritma (7) iz prethodne glave. Prvo, diskutujmo o ulazu našeg algoritma.

Postoje dva slučaja koja istražujemo, slučaj spektralnog klasterovanja bez ograničenja i spektralnog klasterovanja sa ograničenjima. U prvom slučaju, ulaz je graf $G = (V, W)$. U drugom slučaju, ulaz je graf i skup ograničenja, a pošto koristimo proširenje algoritma (7) ulaz je troslojni graf

$$M_G = \{G_1 = (V, W_1), G_2 = (V, W_2), G_3 = (V, W_3)\},$$

gde su slojevi definisani kao u prethodnoj glavi. Imajmo na umu da je prvi slučaj zapravo

podskup drugog slučaja. Hajde da analiziramo slučaj u kome nemamo ograničenja:

- Prvi sloj u grafu M_G je isti kao i graf G .
- Prema tome, Laplasova matrica L_1 je ista kao L .
- Budući da nemamo ograničenja, drugi i treći sloj su prazni i stoga su L_2 i L_3 nula matrice, što takođe važi i za ugrađivanje U_2 i U_3 , respektivno.
- Dakle, kada pogledamo modifikovanu Laplasovu matricu

$$L_{mod} = \sum_{i=1}^m L_i - \alpha \sum_{i=1}^m U_i U_i^T$$

i Laplasijan L vidimo da je ugrađivanje izvedeno iz obe ove matrice isto.

- Zaključak, oba ova grafa indukuju isto grupisanje.

Prema tome, u ostatku ovog poglavlja, kada kažemo ulazni graf, mislićemo na višeslojni graf

$$M_G = \{G_1 = (V, W_1), G_2 = (V, W_2), G_3 = (V, W_3)\}.$$

Jednostavno formulisan, algoritam samoučenja uči skup dodatnih informacija (poput ograničenja) koje poboljšavaju učenje ciljnog problema. Ovaj termin je nastao u polju mašinskog učenja transferom znanja [Dai et al., 2008, Raina et al., 2007]. Algoritam u [Raina et al., 2007] uvećava ulazni skup velikim brojem neobebeženih primera radi poboljšanja klasifikacije. Razlika od polunadgledanog algoritma je u tome što ti neobebeženi primeri nisu nužno uzorkovani iz iste slučajne raspodele. Algoritam u [Dai et al., 2008] sledi iste principe, ali problem koji se rešava je klasterovanje. Zajednička ideja između tih algoritama je da oba zapravo nauče novo predstavljanje ulaznih instanci (nauče funkciju koja preslika ulazne instance u ulazne vektore karakteristika), a nakon toga uče model. U [Wang et al., 2014b] spektralno klasterovanje koje se samostalno uči definiše se kao kombinacija algoritma spektralnog klasterovanja sa ograničenjima iz [Wang et al., 2014a] i učenja novih ograničenja putem problema upotpunjavanja matrice. Ovo je računski izvodivo samo zbog pretpostavke koju su autori izneli u [Wang et al., 2014b], da je matrica ograničenja malog ranga. Ovaj algoritam pripada klasi algoritama za samoučenje, jer koristi znanje o matrici susedstva i znanje o ograničenjima da bi naučio nova ograničenja. Stoga se samoobučava skup ograničenja, koji se zauzvrat koristi za izvođenje spektralnog klasterovanja. Međutim, vremenska i memorijska složenost ovog algoritma je kvadratna po broju primera i stoga nerešiva za velike skupove. Sada ćemo definisati algoritam spektralnog klasterovanja koji se samostalno uči i koji se može primeniti na velike skupove podataka.

Rezultate klasterovanja ćemo koristiti kao dodatna ograničenja, a naš algoritam će biti iterativni. Za razliku od algoritma iz [Wang et al., 2014b] koja koristi rezultate klasterisanja da bi pronašla matricu ograničenja niskog ranga, naš algoritam će koristiti rezultate klasterisanja kao dodatne slojeve u višeslojnom grafu bez ikakvih dodatnih

pretpostavki. Objasnimo detaljnije. Prvo, algoritmi u glavi 2 urade klasterovanje na osnovu aproksimacije modifikovane Laplasove matrice (2.5):

$$C_{mod} = \sum_{i=1}^m C_i - \alpha \sum_{i=1}^m U_i U_i^T(:, \mathcal{L}),$$

U našem slučaju prvi sloj je izveden iz ulaznog skupa, drugi i treći sloj su izvedeni iz skupa ograničenja, a dodatni slojevi će se sami naučiti i izvesti iz rezultata klasterisanja. Stoga bi prva tri sloja trebala biti pouzdanija od sledećih. Dakle, malo ćemo promeniti modifikovanu Laplasove matrice:

$$C_{mod} = \sum_{i=1}^m \alpha_i (C_i - \alpha U_i U_i^T(:, \mathcal{L})), \quad (3.1)$$

gde su $\alpha_i \in (0, 1)$ dodatne težine koje će odrediti pouzdanost sloja.

Hajde da razgovaramo o slojevima grafa. Na početku imamo 3 sloja: ulazni graf, sloj gde su označeni primeri koji trebaju biti u istom klasteru (ML sloj) i sloj gde su označeni primeri koji ne trebaju biti u istom klasteru (CL sloj). Zatim izvodimo klasterovanje i dobijamo particiju čvorova $\{A_1, A_2, \dots, A_k\}$. Imajmo na umu da s obzirom na to da je ovo celokupno znanje o ograničenjima, ne bi nam trebao ML sloj i CL sloj, jer bi oba bila ista. Prema tome, četvrti sloj je sledeći:

$$W_4(i, j) = \begin{cases} 1, & \text{ako } (\exists l) v_i, v_j \in A_l \\ 0, & \text{u suprotnom.} \end{cases} \quad (3.2)$$

Sada se spektralno klasterovanje ponovo izvodi na grafu koji sadrži samo prva četiri sloja i rezultati se slično dodaju u višeslojni graf kao peti sloj, i tako dalje. Ovaj proces se iterativno izvodi t puta, a konačno klasterovanje je rešenje. Algoritam se može naći u *algoritmu (8)*. Vremenska složenost ovog algoritma je $O(t(nk^2 + t_{km}))$, a memorijska složenost je $O(tnl)$. Prema našim saznanjima, ovo je jedino spektralno klasterovanje sa samoučenjem koje ima linearnu vremensku složenost.

3.3 Empirijska evaluacija

Procenjujemo performanse našeg algoritma na skupovima podataka iz stvarnog sveta: Postures, Covtype i Orange. Svi su iz [Lichman, 2013] osim Orange¹(samo atributi bez

Algoritam 8 Brzo spektralno klasterovanje koje se samouči.

procedure FAST-SLC(M_G, k, l, t)

Input: troslojni graf M_G ;

Input: broj kolona l , broj klastera k i broj iteracija t ;

Input: hiperparametri $\alpha, \alpha_{1:t}$;

for $j = 4 \dots t + 1$ **do**

 Uzorkovati indekse kolona \mathcal{L} ;

 $W_1 = W_1(:, \mathcal{L});$
 $W_2 = W_2(:, \mathcal{L});$
 $W_3 = W_3(:, \mathcal{L});$

 Izračunati C_i za sva 3 sloja;

 Izračunati U_i za sva 3 sloja;

if $j \leq t$ **then**
 $m = 3$: zato što za učenje ograničenja koristimo prva 3 sloja;

 Izračunati C_{mod} ;

else
 $m = t$: za klasterovanje koristimo sva ograničenja;

 Izračunati C_{mod} ;

end if
 $Q = sym(C_{mod}(\mathcal{L}, :));$
 $(U_Q, \Sigma_Q) = eig(Q);$
 $\tilde{\Sigma} = \frac{n}{l} \Sigma_Q;$
 $U := (\sqrt{\frac{l}{n}} C U_Q \Sigma_Q^+)(:, 1 : k);$

 Normalizovati vrste matrice U ;

 Klasterovati vrste od U u klastera A_1, \dots, A_k koristeći algoritam KMEANS 1;

 $G_j := (V, W_j)$ kao u (3.2) ako je $j \leq t$;

end for
Print Klasteri A_1, \dots, A_k ;

end procedure

Tabela 3.1: Skupovi podataka na kojima ćemo evaluirati algoritme spektralnog klasterovanja sa samoučenjem. Prikazujemo naziv skupa, broj primera u skupu i koliko atributa ima svaki skup.

Skup podataka	Instance	Atributi
Orange	50 000	43
Postures	78 095	38
Covtype	581 012	54

podataka koji nedostaju) skupa podataka. Pogledajte tabelu (3.1) za više informacija.

¹<http://www.causality.inf.ethz.ch/activelearning.php>

Matrica susedstva se formira pomoću RBF kernela

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2}\right).$$

Ograničenja su izvedena iz slučajnog podskupa A ulaznih vektora karakteristika. Ako važi $v_i, v_j \in A$, onda

1. ako pripadaju istoj klasi, onda dodajemo $\{v_i, v_j, t, type = ML\}$ skupu \mathcal{PC}
2. ako pripadaju različitim klasama, onda dodajemo $\{v_i, v_j, t, type = CL\}$ skupu \mathcal{PC}

Rezultate smo procenili pomoću Randovog indeksa koji je definisan kao u [Manning et al., 2008]:

$$RI = \frac{tp + tn}{tp + fp + tn + fn}$$

gde tp, fp, tn, fn su *true positive, false positive, true negative* i *false negative* respektivno.

Upoređujemo naš algoritam brzog spektralnog klasterovanja sa samoučenjem (**F**ast **S**elf **L**earning - **S**pectral **C**lustering (8)) sa algoritmom spektralnog klasterovanja sa ograničenjima pomoću višeslojnog grafa (**C**onstrained **S**pectral - **M**ulti**L**ayer (7)) i algoritmom za spektralno učenje (**S**pectral learning algoritam iz [Wang et al., 2014b], koji je ekstenzija algoritma iz [Kamvar et al., 2003]) i rezultat predstavimo u sledećoj tabeli. Imajmo na umu da je glavni algoritam iz [Wang et al., 2014b] neprimenljiv za ove skupove podataka zbog velike potrebe za memorijom. Pokazujemo da samoučenje novih ograničenja poboljšava rezultate. Nova ograničenja u algoritmu **fsl-sc** se uče u 10 iteracija. Rezultati su predstavljeni u tabeli (3.2).

Prvo bismo želeli da primetimo da su performanse algoritma poboljšane kada dodajemo još ograničenja, što je jedan od zahteva klasterovanja sa ograničenjima. Pored toga, naš algoritam je nadmašio konkurente u svim skupovima podataka. Zaključujemo da se najveća margina između našeg algoritma **fsl-sc** i njegovih konkurenata postiže na ulaznim skupovima sa većim brojem ograničenja. To znači da je naš algoritam u stanju da bolje koristi ograničenja u odnosu na druge algoritme. Dalje, ako gledamo slučaj bez ograničenja, primetićemo da kada dodamo ograničenja samoučenjem poboljšavamo rezultate. Konačno, naš algoritam takođe ima manju varijansu.

Tabela 3.2: Poređenje performansi algoritama **csp-ml**, **fsl-sc** i **sl** na sledećim skupovima podataka iz stvarnog sveta: Postures, Orange i Covtype. Prikazujemo prosečan Randov indeks i njegovu varijansu. Ograničenja predstavljaju procenat ulaznih vektora za koje znamo oznaku.

Skup podataka	Ograničenja	csp-ml	sl
Postures	0	0.6907 ± 0.0095	0.6907 ± 0.0095
	6	0.6920 ± 0.0127	0.2565 ± 0.0019
	10	0.7004 ± 0.0106	0.3530 ± 0.002
	15	0.7099 ± 0.0120	0.4250 ± 0.0038
Orange	0	0.4940 ± 0.0018	0.4940 ± 0.0018
	6	0.4962 ± 0.0033	0.4950 ± 0.0081
	10	0.5073 ± 0.0059	0.5061 ± 0.0092
	15	0.5202 ± 0.0065	0.5294 ± 0.0154
Covtype	0	0.4807 ± 0.0137	0.4807 ± 0.0137
	6	0.4968 ± 0.0029	0.4565 ± 0.0644
	10	0.5137 ± 0.0178	0.4909 ± 0.0519
	15	0.5367 ± 0.0199	0.5351 ± 0.0521
Skup podataka	Ograničenja	fsl-sc	
Postures	0	0.6927 ± 0.003	
	6	0.6969 ± 0.0113	
	10	0.7130 ± 0.0079	
	15	0.7298 ± 0.0007	
Orange	0	0.4959 ± 0.0002	
	6	0.4990 ± 0.0030	
	10	0.5158 ± 0.0030	
	15	0.5322 ± 0.0013	
Covtype	0	0.4983 ± 0.0082	
	6	0.5002 ± 0.0031	
	10	0.5311 ± 0.0211	
	15	0.5672 ± 0.0191	

Glava 4

O očekivanoj grešci aproksimacije kernel regresije metodom Nistroma i njenom ekstenzijom

Kernel metode predstavljaju klasu algoritama mašinskog učenja koja uče i otkrivaju obrasce u velikom (moguće beskonačnom) prostoru dimenzija karakteristika dobijenih često nelinearnim (moguće beskonačnim) preslikavanjem ulaznog prostora. Glavni problem kernel metoda je njihova vremenska složenost. Za skup podataka od n ulaznih vektora vremenska složenost kernel metode je $O(n^3)$, što je neizvodivo za veliki skup podataka. Metoda zasnovana na slučajnim Nistrom karakteristikama je metoda aproksimacije koja je u stanju da smanji vremensku složenost na $O(np^2 + p^3)$ gde je p broj slučajno izabranih ulaznih primera. Vremenska složenost $O(p^3)$ dolazi iz činjenice da je potrebno izvršiti spektralnu dekompoziciju Gramove matrice dimenzije $p \times p$, i ako je p veliki broj, čak je i aproksimirani algoritam spor.

U ovoj tezi ćemo primeniti aproksimativni SVD metod umesto spektralne dekompozicije da bi još više smanjili vremensku složenost. Ulazni parametri aproksimativnog SVD algoritma su $p \times p$ Gramova matrica i broj $m < p$. U ovom slučaju vremenska složenost je $O(nm^2 + p^2m + m^3)$, na koje se dodaje složenost linearne regresije na m -dimenzionalnim slučajnim vektorima karakteristika koja se primenjuje na kraju algoritma. Dokazaćemo da je greška prediktora naučena ovom metodom skoro ista u očekivanju kao i greška kernel prediktora. Pored toga, empirijski ćemo pokazati da je ovaj prediktor bolji od onog koji koristi samo Nistromovu metodu.

4.1 Uvod

Kernel metodi [Williams and Rasmussen, 2006, Schölkopf et al., 2002b, Shawe-Taylor et al., 2004] su klasa algoritama mašinskog učenja koji uče obrasce u prostorima karakteristika velike (moguće beskonačne) dimenzije dobijene obično nelinearnim preslikava-

njem ulaznog prostora: $\phi : R^D \rightarrow F$. Ovo preslikavanje je često nepoznato i računski nerešivo zbog beskonačne dimenzionalnosti. Iz kernel metoda se razvijaju modeli mašinskog učenja definisani kao

$$f(x, \lambda, b) = \sum_i \lambda k(x_i, x) + b,$$

koji ne zavise eksplicitno od nepoznatog preslikavanja, ali su linearna kombinacija kernel funkcija definisanih na ulaznom prostoru. Ta kernel funkcija zapravo predstavlja i skalarni proizvod u prostoru karakteristika:

$$k(x, y) = \phi(x)^T \phi(y).$$

Na ovaj način, računski nerešivo računanje skalarnog proizvoda u prostoru sa velikim ili beskonačnim dimenzijama zamenjuje se računanjem funkcije kernela u ulaznom prostoru. Sastavni deo svake kernel metode je kernel matrica, čiji su elementi vrednosti kernel funkcije, pa je ona pozitivno semidefinitna i dimenzije $n \times n$, kada je n broj primera. Još jedna pozitivna strana kernel metoda je razdvajanje samog algoritma učenja i definisanje prostora u kojem se učenje izvodi jer je izbor kernel izbor vektorskog prostora kome pripadaju vektori karakteristika. To znači da možemo dizajnirati kernel funkciju od ulaznih primera nezavisno od dobrog algoritma učenja. Pored toga, ulazni primeri ne moraju biti vektori, kernel možemo definisati na grafovima, slikama, stringovima itd [Schölkopf et al., 2004]. Međutim, sa negativne strane kernel metode obično imaju vremensku složenost koja je kubna po broju podataka što je neizvodivo za velike skupove podataka. Na primer, vremenska složenost koja je potrebna samo za izračunavanje kernel matrice je kvadratna po broju primera. Takođe memorijska složenost je $O(n^2)$ što je nerešivo za brojne primene. Rešenja ovog problem uključuju aproksimaciju kernel matrice kao u [Bach and Jordan, 2005, Mahoney and Drineas, 2009].

Metode aproksimacije koje ćemo analizirati u ovom poglavlju aproksimiraju kernel matricu koristeći slučajni podskup ulaznog skupa i predstavljaju popularno rešenje problema vremenske i memorijske složenosti kernel metoda. Od ovih metoda Nistromov metod postiže dobre rezultate i u praksi i teorijski. U Nistromovoj metodi [Williams and Seeger, 2001, Gittens, 2011, Kumar et al., 2012] nasumično selektujemo m (gde je $m < n$) kolona kernel matrice i aproksimiramo celu kernel matricu na osnovu njih. Glavna prednost ovog algoritma je njegova vremenska složenost koja se smanjuje na $O(nm^2 + m^3)$. Memorijska složenost je takođe smanjena na $O(nm)$ jer ne zahteva izračunavanje celokupne matrice.

Autori u [Rahimi and Recht, 2008] su predložili brzu metodu izračunavanja slučajnih karakteristika što zauzvrat dovodi do brzog algoritama učenja. Konkretno, ulazni prostor se preslikava pomoću neke slučajne funkcije i njegovi elementi se nazivaju slučajnim karakteristikama. Ovaj pristup se pojavljuje sa različitim formulacijama u [Rahimi and Recht, 2009] [Yang et al., 2012] [Le et al., 2013]. Na primer, posledica primene Nistromove metode na kernel matricu je računanje m -dimenzionalnih slučajnih vektora karakteristika izračunatih za svaki ulazni vektor. Ti slučajni vektori nazivaju se slučajnim Nistromo-

vim karakteristikama [Yang et al., 2012]. Proizvoljan broj $m < n$ predstavlja i veličinu slučajnog podskupa ulaznog skupa i dimenziju vektora slučajnih karakteristika.

Glavna ideja se ogleda u tome da se koristi p (gde $p > m$) nasumično izabranih kolona kernel matrice za konstrukciju m -dimenzionalnih slučajnih vektora karakteristika uz zadržavanje vremenske složenosti linearne po n i da će tako kreirani vektori proizvesti bolje rezultate nego algoritmi koji upotrebljavaju samo m kolona. Ovo se razlikuje od Nistromove metode koja koristi m izabranih kolona za izvođenje m -dimenzionalnih vektora karakteristika. Autori u [Li et al., 2015b] su Nistromovu metodu promenili primenom metode aproksimacije SVD-a [Halko et al., 2011] da bi je ubrzali. Korišćenje standardne Nistromove metode zahtevaće izvođenje SVD-a na $p \times p$ simetričnoj podmatrici kernel matrice i ona će proizvesti p -dimenzionalne slučajne vektore karakteristika. Međutim, ako se primeni aproksimativni SVD kao u [Li et al., 2015b] algoritam će proizvesti m -dimenzionalne slučajne vektore karakteristika. Ovo takođe omogućava da algoritam zadrži vremensku složenost linearnu po n .

U ovoj glavi ćemo izvršiti teorijsku analizu ove metode aproksimacije primenjene na problem regresije najmanjih kvadrata. Pored toga, pokazaćemo teoretski da ovaj algoritam sa subkvadratnom složnošću pokazuje iste prediktivne performanse kao i kernel regresija. Pokazaćemo da možemo odabrati p tako da očekivana greška aproksimirane kernel regresije bude približno ista kao greška same kernel regresije. Dalje, pokazujemo na skupovima podataka iz stvarnog sveta da m -dimenzionalni slučajni vektori karakteristika izvedeni iz p slučajno izabranih ulaznih vektora daju bolje rezultate od slučajnih Nistromovih karakteristika.

4.2 Brza kernel regresija

U ovom odeljku ćemo opisati algoritam za brzu aproksimativnu kernel regresiju. Pretpostavimo da je skup ulaznih podataka sledeći skup:

$$T = \{(x_i \in \mathbf{R}^d, y_i \in \mathbf{R})\}_{i=\overline{1,n}} \quad (4.1)$$

Algoritam se sastoji iz dva glavna koraka. U prvom koraku svaki vektor karakteristika se preslikava u m -dimenzionalni slučajni vektor karakteristika, koji se naziva slučajni prikaz. U drugom koraku se primenjuje linearna regresija. Sada ćemo ukratko objasniti Nistromovu metodu, koja se koristi za konstrukciju m -dimenzionalnih slučajnih vektora karakteristika i njeno proširenje koje se dobija kombinacijom sa slučajnim SVD-om.

4.3 Nistromov vektor karakteristika

Slučajne karakteristike su vektori malih dimenzija izvedeni iz ulaznog skupa pomoću nekog slučajnog preslikavanja. Svaki ulazni vektor se preslikava u odgovarajući vektor slučajnih karakteristika. U literaturi je prisutno nekoliko vrsta slučajnih karakteristika kao što su Slučajne Furijeove karakteristike [Rahimi and Recht, 2008] ili Slučajne Nistromove karakteristike [Yang et al., 2012]. Njihova glavna prednost je upotreba kernela sa vremenskom složenošću linearnom po broju instanci.

Pretpostavimo da imamo skup podataka $\{(x_i \in \mathbf{R}^d)_{i=1}^n\}$ i kernel (pozitivna semidefinitna funkcija)

$$k : \mathbf{R}^d \times \mathbf{R}^d \rightarrow \mathbf{R}.$$

Gramova matrica

$$K_{ij} = k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$$

je $n \times n$ pozitivna semidefinitna matrica. Funkcija $\Phi(x)$ preslikava podatke iz \mathbf{R}^d u prostor velike dimenzije (moguće beskonačne). U metodi slučajnih karakteristika, svaki ulazni vektor x_i je preslikan u m -dimenzionalni slučajni vektor karakteristika r_i tako da $r_i^T r_j$ aproksimira $k(x_i, x_j)$:

$$k(x_i, x_j) \approx r_i^T r_j.$$

Nistromova metoda je metoda aproksimacije matrice i kada se primeni na Gramovu matricu K , njene posledice su slučajne karakteristike. Konkretno za dato $m \ll n$ Nistromov algoritam uzorkuje m vektora podataka $\{\hat{x}_i\}_{i=\overline{1,m}}$ iz ulaznog skupa. Matrica aproksimacije \tilde{K} izračunava se na sledeći način:

$$K \approx \tilde{K} := k(x_{1:n}, \hat{x}_{1:m})k(\hat{x}_{1:m}, \hat{x}_{1:m})^+k(x_{1:n}, \hat{x}_{1:m})^T \quad (4.2)$$

Algoritam 9 Kernel regresija: NF algoritam.

[Halko et al., 2011, McWilliams et al., 2013]

procedure NF(T, k, m)

Input: skup podataka $T = \{(x_i \in \mathbf{R}^d, y_i \in \mathbf{R})\}_{i=\overline{1,n}}$;

Input: kernel k ;

Input: broj uzorkovanih vektora karakteristika za Nistromovu metodu m ;

Kreirati slučajnu permutaciju $perm$ iz $\overline{1, n}$;

for $i = \overline{1, m}$ **do**

$\hat{x}_i = x_{perm_i}$; (* Uzorkovati m vektora $\{\hat{x}_i\}_{i=\overline{1,m}}$; *)

end for

$(\hat{V}, \hat{D}) = eig(k(\hat{x}_{1:m}, \hat{x}_{1:m}))$; (* Odraditi spektralnu dekompoziciju $k(\hat{x}_{1:m}, \hat{x}_{1:m}) = \hat{V}\hat{D}\hat{V}^T$ *)

for $i = \overline{1, n}$ **do**

$r_i = \hat{D}^{-1/2}\hat{V}^T k(x_{1:n}, \hat{x}_{1:m})^T$; (* Nistromove karakteristike *)

end for

(* Linije 9–12 će biti izmenjeni u našem algoritmu 10 *)

Uraditi linearnu regresiju na (r_i, y_i) $i = \overline{1, n}$;

Output: Linearni regresor.

end procedure

gde je

$$k(\hat{x}_{1:m}, \hat{x}_{1:m})^+ = \hat{V}\hat{D}^{-1}\hat{V}^T$$

pseudo inverz matrice $k(\hat{x}_{1:m}, \hat{x}_{1:m})$. Kolone matrice \hat{V} su sopstveni vektori, a dijagonalni elementi matrice \hat{D} su sopstvene vrednosti matrice $k(\hat{x}_{1:m}, \hat{x}_{1:m})$. Slučajni vektor karakteristika r_i definišemo na sledeći način

$$r_i = \hat{D}^{-1/2}\hat{V}^T k(x_{1:n}, \hat{x}_{1:m})^T. \quad (4.3)$$

Prema izrazu(4.2) $\tilde{K}_{ij} = r_i^T r_j$. Stoga slučajni vektor r_i aproksimira $\Phi(x_i)$. Dalje, autori u [McWilliams et al., 2013] zovu slučajni vektor r_i vektor Nistromovih karakteristika ili slučajni Nistromov prikaz . Preslikavanje $z(x_i) = r_i$ naziva se slučajnim Nistromovim preslikavanjem.

Obuka linearne regresije na ovom skupu podataka je ista kao i obuka na modelu kernel regresije, gde se umesto kernel matrice koristi njena Nistromova aproksimacija.

4.4 Aproksimacija spektralne dekompozicije

Sada ćemo ukratko objasniti aproksimaciju dekompozicije koja se koristi u ekstenziji Nistromove metode. Pretpostavimo da imamo realnu simetričnu matricu

$$W \in \mathbf{R}^{p \times p}.$$

U ovom poglavlju ćemo primeniti ovaj algoritam na matrici kreiranoj tokom Nistromove metode iz uzorkovanih redova i kolona ($W = k(\hat{x}_{1:p}, \hat{x}_{1:p})$ gde su $\hat{x}_{1:p}$ uzorkovane instance). Mi želimo da izvršimo spektralnu dekompoziciju matrice W . Vremenska složenost spektralne dekompozicije je $O(p^3)$. Korišćenjem algoritma iz [Halko et al., 2011] aproksimirana dekompozicija se može izračunati sa vremenskom složenošću od $O(pm^2 + m^3)$ uz upotrebu slučajne matrice dimenzija $p \times m$ gde je m ulazni parametar. Više reči o tome je bilo u poglavlju 1.1. Ova metoda se sastoji od nekoliko koraka:

- Generisati Gausovu slučajnu matricu

$$\Omega \in \mathbf{R}^{p \times (m+l)},$$

gde je l mali parametar prekomernog uzorkovanja (obično 5).

- Konstruisati matricu

$$Y = W\Omega \in \mathbf{R}^{p \times (m+l)}.$$

Parametar prekomernog uzorkovanja l je izabran jer povećava šanse da matrica Y zahvati m -dimenzionalni potprostor prostora kolona matrice W .

- Izvršiti QR dekompoziciju matrice Y . Matrica Q nam daje sledeću aproksimaciju $W \approx QQ^T W$ odakle sledi

$$W \approx QQ^T W QQ^T.$$

- Generisati matricu

$$B = Q^T W Q$$

i izvršiti spektralnu dekompoziciju matrice

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

- Kolone matrice QU su približni sopstveni vektori matrice W i dijagonalni elementi matrice Λ su približne sopstvene vrednosti matrice W .
- Matrica V je izvedena iz približnih sopstvenih vektora matrice W kojima odgovaraju m najvećih približnih sopstvenih vrednosti. Stoga je

$$W = V \Lambda_m V^T$$

gde matrica Λ_m sadrži najveće približne sopstvene vrednosti.

U sledećem odeljku ćemo primeniti ove algoritme na Nistromovu metodu. Umesto da uzorkujemo m vektore iz ulaznog skupa podataka, mi uzorkujemo $p > m$ vektora i

izvršavamo metodu koja predstavlja kombinaciju Nistromove metode i aproksimirane SVD dekompozicije da bismo kreirali m -dimenzionalne vektore slučajnih karakteristika.

4.5 Vektori slučajnih karakteristika kreirani ekstenzijom Nistromove metode

Pretpostavimo da imamo skup podataka $\{(x_i \in \mathbf{R}^d)_{i=1}^n\}$ i kernel (pozitivna semidefinitna funkcija)

$$k : \mathbf{R}^d \times \mathbf{R}^d \rightarrow \mathbf{R}.$$

Za dato $p \ll n$ Nistromov algoritam uzorkuje p vektora karakteristika $\{\hat{x}_i\}_{i=1}^p$ iz ulaznog skupa. Podsetimo se da je iz Nistromove metode sledeća matrica aproksimacija Gramove matrice ($K_{ij} = k(x_i, x_j)$):

$$\tilde{K} := k(x_{1:n}, \hat{x}_{1:p})k(\hat{x}_{1:p}, \hat{x}_{1:p})^+k(x_{1:n}, \hat{x}_{1:p})^T.$$

Iz nje izvodimo p -dimenzionalne Nistromove karakteristike

$$r_i = \hat{D}^{-1/2}\hat{V}^T k(x_{1:n}, \hat{x}_{1:p})^T.$$

Međutim, naš cilj je preslikavanje ulaznih vektora u m -dimenzionalni prostor korišćenjem p uzorkovanih primera, gde je $p > m$. Iako mi dodajemo još jednu aproksimaciju, želimo da zadržimo glavno svojstvo slučajnih Nistromovih karakteristika, a to je da za svaku grešku ϵ postoji dovoljno veliko p tako da greška postignuta m -dimenzionalnim vektorima karakteristika (izvedeno iz p uzorkovanih primera) je manja od ϵ .

Da bismo proizveli m -dimenzionalne slučajne vektore, predlažemo da koristimo aproksimativnu dekompoziciju sopstvenih vrednosti za izračunavanje pseudo inverza i stoga za izračunavanje aproksimacije kernel matrice kao u [Li et al., 2015b]:

$$K \approx L := k(x_{1:n}, \hat{x}_{1:p})k(\hat{x}_{1:p}, \hat{x}_{1:p})^*k(x_{1:n}, \hat{x}_{1:p})^T$$

gde je

$$k(\hat{x}_{1:p}, \hat{x}_{1:p})^* = \hat{V}\hat{D}^{-1}\hat{V}^T$$

aproksimacija pseudo inverza matrice $k(\hat{x}_{1:p}, \hat{x}_{1:p})$. Kolone matrice $\hat{V} \in \mathbb{R}^{p \times m}$ su približni sopstveni vektori, a dijagonalni elementi matrice $\hat{D} \in \mathbb{R}^{m \times m}$ su približne sopstvene vrednosti matrice $k(\hat{x}_{1:p}, \hat{x}_{1:p})$. Približne sopstvene vrednosti i približni sopstveni vektori izračunavaju se pomoću aproksimiranog (nasumičnog) SVD-a (4.4). Ovu matricu ćemo

zvati RNiström kernel matrica, a ovu metodu RNiström metoda ili tačnije RNiström (p , m) metoda. Stoga se slučajne karakteristike koje ćemo nazvati RNiström karakteristike izračunavaju na sledeći način:

$$r_i = \hat{D}^{-1/2} \hat{V}^T k(x_{1:n}, \hat{x}_{1:p})^T$$

Na kraju primenjujemo algoritam linearne regresije. Sastavljajući sve zajedno dobijamo algoritam (10) koji ćemo nazvati RNiström kernel regresija.

Algoritam 10 Kernel regresija: RNF algorithm.

- 1: **procedure** RNF(T, k, p, m, l)
 - 2: **Input:** skup podataka $T = \{(x_i \in \mathbf{R}^d, y_i \in \mathbf{R})\}_{i=\overline{1,n}}$;
 - 3: **Input:** kernel k ;
 - 4: **Input:** broj uzorkovanih vektora karakteristika p , dimenzija slučajnih vektora karakteristika m ;
 - 5: Kreirati slučajnu permutaciju $perm$ iz $\overline{1, n}$;
 - 6: **for** $i = \overline{1, p}$ **do**
 - 7: $\hat{x}_i = x_{perm_i}$; (* Uzorkovati m vektora $\{\hat{x}_i\}_{i=\overline{1,p}}$; *)
 - 8: **end for**
 - 9: Izračunati $\Omega \in \mathbb{R}^{p \times (m+l)}$ slučajnu Gausovu matricu;
 - 10: $W = k(\hat{x}_{1:p}, \hat{x}_{1:p})$;
 - 11: $Y = W\Omega$;
 - 12: $(Q, R) = qr(Y)$; (* Izračunati QR dekompoziciju Y *)
 - 13: $B = Q^T W Q$;
 - 14: $(U, \Lambda) = eig(B)$; (* Izračunati spektralnu dekompoziciju $B = U\Lambda U^T$ *)
 - 15: $\hat{V} = QU$;
 - 16: $\hat{D} = \Lambda$;
 - 17: **for** $i = \overline{1, n}$ **do**
 - 18: $r_i = \hat{D}^{-1/2} \hat{V}^T k(x_{1:n}, \hat{x}_{1:p})^T$; (* RNiström karakteristike *)
 - 19: **end for**
 - 20: (* Linije 9–19 su izmenjeni linije 9–12 iz algoritma 9 *)
 - 21: Uraditi linearnu regresiju na $(r_i, y_i) \quad i = \overline{1, n}$;
 - 22: **Output:** Linearni regresor.
 - 23: **end procedure**
-

U sledećem odeljku, teorema (4.6.4), pokazujemo da je približni prediktor (prediktor naučen na slučajnim karakteristikama) dovoljno blizu originalnom prediktoru (najbolji sveukupni prediktor).

4.6 Analiza približne greške

U slučaju regresije sa RNiström karakteristikama, originalna kernel matrica se zamenjuje sa RNiström kernel matricom L . Ne upoređujemo direktno aproksimaciju kernel matrice, već analiziramo kvalitet regresije (tj. grešku predikcije) koja koristi ovu aproksimaciju. Dokazaćemo da je očekivana greška RNiström regresije, očekivanje od

$$\|(L + n\lambda I)^{-1}Lz - z\|^2,$$

s obzirom na raspodelu aproksimacije, skoro ista kao greška kernel regresije

$$\|(K + n\lambda I)^{-1}Kz - z\|^2.$$

Podsetimo se da je matrica L RNiström aproksimacija matrice K .

Prvo nabrajamo dve teoreme koje ćemo koristiti u dokazu leme o granici približnog matičnog proizvoda.

Teorema 4.6.1. [Gross and Nesme, 2010] Neka je \mathcal{C} konačan skup pozitivnih semi-definitnih matrica dimenzija n . Uzorkujemo (X_1, \dots, X_p) iz \mathcal{C} uniformno bez zamene i uzorkujemo (Y_1, \dots, Y_p) iz \mathcal{C} uniformno sa zamenom. Neka je f konveksna funkcija na skupu \mathcal{C} . Tada važi:

$$E_X[f(\sum_i X_i)] \leq E_Y[f(\sum_i Y_i)]$$

Teorema 4.6.2. (Matrična Bernstein-ova teorema) [Mackey et al., 2014] Neka je $(Y_k)_{k \geq 1}$ nezavistan niz simetričnih i pozitivno semidefinitnih slučajnih matrica koje zadovoljavaju

$$EY_k = 0 \text{ i } \|Y_k\| \leq R \text{ za svaki indeks } k$$

Neka je $\sigma^2 = \|\sum_k EY_k^2\|$. Tada je za svako $t \geq 0$,

$$P(\lambda_{\max}(\sum_k Y_k) \geq t) \leq d \exp\left(\frac{-t^2}{3\sigma^2 + 2Rt}\right)$$

$$E[\lambda_{\max}(\sum_k Y_k)] \leq \sigma\sqrt{3 \log d} + R \log d$$

Sada definišemo lemu koja analizira aproksimaciju matičnog proizvoda. Za n -dimenzionalnu matricu $\frac{1}{n}\Psi^T\Psi$ i njenu aproksimaciju $\frac{1}{p}\Psi_S^T\Psi_S^T$ gde Ψ_S sadrži samo p izabranih kolona matrice Ψ , sledeća lema daje gornju granicu očekivanja spektralne norme njihove razlike. Ovaj rezultat ćemo primeniti u dokazu teoreme (4.6.4), gde ćemo ga koristiti za dokazivanje monotonosti regularizovanih RNyström aproksimacija kernel matrica.

Postavka leme je vrlo slična lemi u [Bach, 2013], sa razlikom da je dokaz malo drugačiji i, po našem mišljenju, jednostavniji. Koristili smo noviju verziju Bernsteinove teoreme iz [?], dok se u [Bach, 2013] koristi teorema iz [Tropp, 2011, 2012]. Takođe, umesto rezultata iz [Tropp, 2011, 2012] koristili smo teoremu (4.6.1).

Lema 4.6.1. *Neka je matrica $\Psi \in \mathbb{R}^{n \times r}$ takva da za svako i važi*

$$\|\Psi(i, :)\| \leq R.$$

Neka je $S \subset \{1, 2, \dots, n\}$ slučajni podskup sa elementima odabranim uniformno bez zamene. Onda važi

$$E\left[\frac{1}{n}\Psi^T\Psi - \frac{1}{p}\Psi_S^T\Psi_S^T\right] \leq \sqrt{\frac{R^2}{p}\lambda_{\max}\left(\frac{1}{n}\Psi^T\Psi\right)3\log n} + \frac{1}{p}\lambda_{\max}\left(\frac{1}{n}\Psi^T\Psi\right)\log n$$

Dokaz. Posmatramo matricu

$$A = \frac{1}{n}\Psi^T\Psi - \frac{1}{p}\Psi_S^T\Psi_S^T = \frac{1}{n}\sum_{i=1, \overline{n}}\psi_i\psi_i^T - \frac{1}{p}\sum_{i \in S}\psi_i\psi_i^T$$

gde ψ_i predstavlja i -tu vrstu matrice Ψ . Za matricu A znamo da je $E[A] = 0$. Neka je

$$B = \frac{1}{n}\Psi^T\Psi - \frac{1}{p}\Psi_J^T\Psi_J^T$$

slučajna matrica izvedena na sličan način kao matrica A sa jednom razlikom:

$$J \subset \{1, 2, \dots, n\}$$

je slučajni podskup čiji su elementi nasumično odabrani sa zamenom. Stoga se matrica B može izraziti na sledeći način:

$$B = \frac{1}{n}\sum_{i=1, \overline{n}}\psi_i\psi_i^T - \frac{1}{p}\sum_{i \in J}\psi_i\psi_i^T = \frac{1}{n}\sum_{i=1, \overline{n}}\psi_i\psi_i^T - \frac{1}{p}\sum_{i=1, \overline{n}}\sum_{j=1, \overline{p}}z_i^j\psi_i\psi_i^T$$

gde za svako i, j važi

$$P(z_i^j) = \frac{1}{n}.$$

Iz teoreme 4.6.1 i iz činjenice da je svaka norma konveksna funkcija imamo:

$$E[||A||] \leq E[||B||].$$

Ovaj rezultat koristimo zbog njegove jednostavnosti dok su autori u [Bach, 2013] koristili rezultate iz [Tropp, 2011, 2012]. Matricu B možemo napisati u sledećem obliku:

$$\begin{aligned} B &= \frac{1}{n} \sum_{i=\overline{1,n}} \psi_i \psi_i^T - \frac{1}{p} \sum_{i=\overline{1,n}} \sum_{j=\overline{1,p}} z_i^j \psi_i \psi_i^T \\ &= \frac{1}{n} \frac{1}{p} \sum_{i=\overline{1,n}} \sum_{j=\overline{1,p}} \psi_i \psi_i^T - \frac{1}{p} \sum_{i=\overline{1,n}} \sum_{j=\overline{1,p}} z_i^j \psi_i \psi_i^T \\ &= \sum_{j=\overline{1,p}} M_j \end{aligned} \quad (4.4)$$

gde je $M_j = \frac{1}{p} \sum_{i=\overline{1,n}} z_i^j (\frac{1}{n} \Psi^T \Psi - \psi_i \psi_i^T)$. Ovde smo koristili rezultat iz [Mackey et al., 2014], dok u [Bach, 2013] se koristi rezultat iz [Tropp, 2011, 2012]. Po [Bach, 2013] važi:

$$\begin{aligned} E[M_j] &= 0 \\ \left\| \sum_{j \in \overline{1,p}} E[M_j^2] \right\| &= \frac{R^2}{p} \lambda_{\max} \left(\frac{1}{n} \Psi^T \Psi \right) \end{aligned} \quad (4.5)$$

Sada možemo da primenimo teoremu (4.6.2) da bi dobili:

$$E[||B||] \leq \sqrt{\frac{R^2}{p} \lambda_{\max} \left(\frac{1}{n} \Psi^T \Psi \right) 3 \log n} + \frac{1}{p} \lambda_{\max} \left(\frac{1}{n} \Psi^T \Psi \right) \log n$$

odakle sledi željeni rezultat.

□

Sada ćemo dokazati lemu o gornjoj granici sopstvenih vrednosti matrice koja je neophodna za dokaz teoreme koja analizira grešku regresije. Ova lema pokazuje da je matrica definisana na poseban način manja od matrice identiteta. U dokazu teoreme (4.6.4) matrica, izvedena iz kernel matrice, imaće strukturu koja zadovoljava sledeću lemu, pa ćemo koristiti rezultat leme da dokažemo monotonost regularizovane RNiströmove aproksimacije.

Lema 4.6.2. *Neka je $\Phi \in \mathbb{R}^{n \times n}$ i $\gamma > 0$. Tada za matricu $\Psi = \Phi(\frac{1}{n}\Phi^T\Phi + \gamma I)^{-\frac{1}{2}}$ važi:*

$$\frac{1}{n}\Psi^T\Psi \preceq I$$

Dokaz. Iz $\gamma > 0$ imamo

$$\frac{1}{n}\Phi^T\Phi \preceq \frac{1}{n}\Phi^T\Phi + \gamma I.$$

Iz činjenice da za bilo koje tri matrice X, Y, Z važi da iz $X \preceq Y$ sledi $Z^T X Z \preceq Z^T Y Z$ dobijamo:

$$\left(\frac{1}{n}\Phi^T\Phi + \gamma I\right)^{-\frac{1}{2}} \frac{1}{n}\Phi^T\Phi \left(\frac{1}{n}\Phi^T\Phi + \gamma I\right)^{-\frac{1}{2}T} \preceq \left(\frac{1}{n}\Phi^T\Phi + \gamma I\right)^{-\frac{1}{2}} \left(\frac{1}{n}\Phi^T\Phi + \gamma I\right) \left(\frac{1}{n}\Phi^T\Phi + \gamma I\right)^{-\frac{1}{2}T}$$

$$\frac{1}{n} \left(\Phi \left(\frac{1}{n}\Phi^T\Phi + \gamma I\right)^{-\frac{1}{2}}\right)^T \Phi \left(\frac{1}{n}\Phi^T\Phi + \gamma I\right)^{-\frac{1}{2}} \preceq I$$

$$\frac{1}{n}\Psi^T\Psi \preceq I$$

□

Posledica 4.6.1. *Neka je $\Phi \in \mathbb{R}^{n \times n}$ i $\gamma > 0$. Tada za matricu*

$$\Psi = \Phi \left(\frac{1}{n}\Phi^T\Phi + \gamma I\right)^{-\frac{1}{2}}$$

i za svaki slučajni podskup $S \subset \{1, 2, \dots, n\}$ sledi da svaka sopstvena vrednost matrice

$$\frac{1}{n}\Psi^T\Psi - \frac{1}{p}\Psi_S^T\Psi_S \preceq I$$

pripada segmentu $[0, 1]$.

Kako RNiström aproksimacija matrice proširuje Nistromov metod uz primenu aproksimiranog SVD-a ovde ćemo navesti teoremu o granici očekivane greške aproksimiranog SVD-a:

Teorema 4.6.3. [Li et al., 2015b] Kod aproksimiranog SVD-a (4.4) matrice $W \in \mathbb{R}^{p \times p}$, gde računamo k približnih sopstvenih vrednosti i gde je l parametar prekomernog uzorkovanja (obično mali proizvoljan broj), očekivana greška

$$\|W - QQ^T W\|$$

(gde je uzeta u obzir slučajnost u Gausovoj slučajnoj matrici) je ograničena odozgo sa

$$\left(\sqrt{\frac{k}{l-1}} + \frac{e\sqrt{k+l}}{l\sqrt{p-k}} \right) \|WW^T\|$$

Sada smo nabrojali sve leme i teoreme koje su neophodne za dokaz našeg glavnog rezultata u ovom poglavlju. Proširićemo teoremu iz [Bach, 2013] jer proširujemo Nistromovu aproksimaciju aproksimacijom SVD-a. Obe aproksimacije, i Nistromova aproksimacija i RNiström aproksimacija, koriste p slučajno uzorkovanih kolona za aproksimaciju matrice.

Greška RNistrom kernel regresije može biti proizvoljno blizu greške originalne kernel regresije, odabirom odgovarajuće donje granice broja uzorkovanih kolona. To znači da ako zamenimo kernel matricu u kernel regresiji približnom, dobijamo približno isti prediktor. Na primer, ograničenje broja izabranih kolona u skupu podataka od 1000 elemenata (podskup skupa podataka calHousing, pogledajte tabelu (3.1)) sa greškom $\delta = 0.1$ je oko 100.

Teorema 4.6.4. Neka je dato $\lambda > 0$, vektor izlaznih opservacija $z \in \mathbb{R}^n$ i kernel matrica izvedena iz ulaznih podataka $K \in \mathbb{R}^{n \times n}$. Pretpostavimo da je $d = n \|diag(K(K + n\lambda I)^{-1})\|_\infty$ i da je $R^2 = \|K\|$. Definišimo sledeću estimaciju

$$z_K = (K + n\lambda I)^{-1} K z.$$

Pretpostavimo da je skup S slučajni podskup $p > k$ indeksa iz skupa $\{1, 2, \dots, n\}$. Neka je L aproksimacija kernel matrice bazirana na RNiström (p, k) metodi gde je l parametar prekomernog uzorkovanja. Definišimo sledeću estimaciju aproksimacije

$$z_L = (L + n\lambda I)^{-1} L z.$$

Za svako $\delta \in (0, 1)$ i

$$p \geq 2 \left(\log n \sqrt{\frac{6d \log n}{10\delta}} + \frac{tR^2\lambda\delta}{2} \right)$$

za $t = 1 + \sqrt{\frac{k}{l-1}} + \frac{e\sqrt{k+l}}{l}$ sledeći izraz je korektan

$$\frac{1}{n}E[||z_L - z||^2] \leq (1 + 6\delta)\frac{1}{n}||z - z_K||^2.$$

Dokaz. Pošto je K kernel matrica, postoji matrica $\Phi \in \mathbb{R}^{n \times n}$ takva da je $K = \Phi\Phi^T$. Približna kernel matrica L koja aproksimira kernel matricu K na osnovu kombinacije Nistromove metode i aproksimacije spektralne dekompozicije može se napisati na sledeći način

$$L = K(:, S)Q(Q^T K(S, S)Q)^{-1}Q^T K(S, :)$$

gde je matrica Q izvedena iz QR dekompozicije matrice $K(S, S)\Omega$ (Ω je Gausova slučajna matricea dimenzije $p \times (l+k)$) kao u poglavlju 4.4. Matrica $K(:, S)$ se može napisati na sledeći način

$$K(:, S) = \Phi\Phi_S^T$$

gde je $\Phi_S = \Phi(S, :)$. Neka je

$$L_\gamma = \Phi\Phi_S^T Q(Q^T \Phi_S \Phi_S^T Q + p\gamma I)^{-1} Q^T \Phi_S \Phi^T$$

regulazrizovana aproksimacija kernel matrice. Možemo da napišemo

$$L_\gamma = \Phi N_\gamma \Phi^T$$

gde je

$$N_\gamma = \Phi_S^T Q(Q^T \Phi_S \Phi_S^T Q + p\gamma I)^{-1} Q^T \Phi_S.$$

Koristeći Sherman-Morrison-Woodbury identitet dobijamo:

$$\begin{aligned} N_\gamma &= \Phi_S^T Q(Q^T \Phi_S \Phi_S^T Q + p\gamma I)^{-1} Q^T \Phi_S \\ &= \Phi_S^T Q(Q^T \Phi_S (Q^T \Phi_S)^T + p\gamma I)^{-1} Q^T \Phi_S \\ &= (Q^T \Phi_S)^T (Q^T \Phi_S) ((Q^T \Phi_S)^T (Q^T \Phi_S) + p\gamma I)^{-1} \\ &= \Phi_S^T Q Q^T \Phi_S (\Phi_S^T Q Q^T \Phi_S + p\gamma I)^{-1} \\ &= I - \gamma \left(\frac{1}{p} \Phi_S^T Q Q^T \Phi_S + \gamma I \right)^{-1} \end{aligned} \tag{4.6}$$

Pomoću Sherman-Morrison–Woodbury identiteta, približna greška se može izračunati na sledeći način:

$$\begin{aligned}
\frac{1}{n} \|z - z_{L_\gamma}\| &= \frac{1}{n} \|z - (L_\gamma + n\lambda I)^{-1} L_\gamma z\| \\
&= n\lambda^2 \|(L_\gamma + n\lambda I)^{-1} z\| \\
&= n\lambda^2 z^T (\Phi N_\gamma \Phi^T + n\lambda I)^{-2} z \quad (4.7)
\end{aligned}$$

Obe funkcije $\gamma \rightarrow N_\gamma$ i greška predikcije su matricno opadajuće. Stoga, kako bi se pronašla gornja granica za grešku $\|z - z_L\|^2$ dovoljno je pronaći gornju granicu za $\|z - z_{L_\gamma}\|$ za bilo koje $\gamma > 0$ zato što je $L = L_0$. Dalje, kako bi se pronašla gornja granica za $\|z - z_{L_\gamma}\|$ za bilo koje $\gamma > 0$ dovoljno je pronaći matricu koja je donja granica za N_γ .

Neka je $\Psi = \Phi(\frac{1}{n}\Phi^T\Phi + \gamma I)^{-\frac{1}{2}}$ i matrica izvedena uzorkovanjem redova iz Ψ je

$$\Psi_S = \Phi_S(\frac{1}{n}\Phi^T\Phi + \gamma I)^{-\frac{1}{2}}.$$

Odavde sledi:

$$\begin{aligned}
\Psi^T \Psi &= (\frac{1}{n}\Phi^T\Phi + \gamma I)^{-\frac{1}{2}} \Phi^T \Phi (\frac{1}{n}\Phi^T\Phi + \gamma I)^{-\frac{1}{2}} \\
\Psi_S^T \Psi_S &= (\frac{1}{n}\Phi^T\Phi + \gamma I)^{-\frac{1}{2}} \Phi_S^T \Phi_S (\frac{1}{n}\Phi^T\Phi + \gamma I)^{-\frac{1}{2}} \\
\Psi_S^T Q Q^T \Psi_S &= (\frac{1}{n}\Phi^T\Phi + \gamma I)^{-\frac{1}{2}} \Phi_S^T Q Q^T \Phi_S (\frac{1}{n}\Phi^T\Phi + \gamma I)^{-\frac{1}{2}} \quad (4.8)
\end{aligned}$$

Primenjujući ovo na izraz N_γ i ako je $X = (\frac{1}{n}\Phi^T\Phi + \gamma I)^{\frac{1}{2}}$ dobijamo

$$\begin{aligned}
N_\gamma &= I - \gamma \left(\frac{1}{p} \Phi_S^T Q Q^T \Phi_S + \gamma I \right)^{-1} \\
&= I - \gamma \left(\frac{1}{p} \Phi_S^T Q Q^T \Phi_S + \gamma I + \frac{1}{n} \Phi^T \Phi - \frac{1}{n} \Phi^T \Phi \right)^{-1} \\
&= I - \gamma \left(X X - \frac{1}{n} X X^{-1} \Phi^T \Phi X^{-1} X + \frac{1}{p} X X^{-1} \Phi_S^T Q Q^T \Phi_S X^{-1} X \right)^{-1} \\
&= I - \gamma \left(\frac{1}{n} \Phi^T \Phi + \gamma I \right)^{-\frac{1}{2}} \left(I - \frac{1}{n} \Psi^T \Psi + \frac{1}{p} \Psi_S^T Q Q^T \Psi_S \right)^{-1} \left(\frac{1}{n} \Phi^T \Phi + \gamma I \right)^{\frac{1}{2}}
\end{aligned} \tag{4.9}$$

Definišimo $t_{S,Q} = \lambda_{\max} \left(\frac{1}{n} \Psi^T \Psi - \frac{1}{p} \Psi_S^T Q Q^T \Psi_S \right)$. Po posledici (4.6.1) znamo $0 \leq t_{S,Q} \leq 1$. Odavde sledi:

$$\begin{aligned}
\frac{1}{n} \Psi^T \Psi - \frac{1}{p} \Psi_S^T Q Q^T \Psi_S &\preceq t_{S,Q} I \\
(1 - t_{S,Q}) I &\preceq I - \frac{1}{n} \Psi^T \Psi + \frac{1}{p} \Psi_S^T Q Q^T \Psi_S \\
\left(I - \frac{1}{n} \Psi^T \Psi + \frac{1}{p} \Psi_S^T Q Q^T \Psi_S \right)^{-1} &\preceq \frac{1}{1 - t_{S,Q}} I
\end{aligned} \tag{4.10}$$

Iz ovoga sledi

$$\begin{aligned}
I - N_\gamma &= \gamma \left(\frac{1}{n} \Phi^T \Phi + \gamma I \right)^{-\frac{1}{2}} \left(I - \frac{1}{n} \Psi^T \Psi + \frac{1}{p} \Psi_S^T Q Q^T \Psi_S \right)^{-1} \left(\frac{1}{n} \Phi^T \Phi + \gamma I \right)^{\frac{1}{2}} \\
&\preceq \frac{\gamma}{1 - t_{S,Q}} \left(\frac{1}{n} \Phi^T \Phi + \gamma I \right)^{-1}
\end{aligned} \tag{4.11}$$

Odavde dobijamo $K - L_\gamma = \Phi (I - N_\gamma) \Phi^T \preceq \frac{n\gamma}{1 - t_{S,Q}} I$. Pretpostavimo da je $\frac{\gamma}{\lambda(1 - t_{S,Q})} \leq 1$, odakle sledi da je

$$(L_\gamma + n\lambda I)^{-1} \preceq \left(1 - \frac{\gamma}{\lambda(1 - t_{S,Q})} \right)^{-1} (K + n\lambda I)^{-1}$$

Primenjujući ovo na približnu grešku aproksimacije dobijamo

$$\begin{aligned}
\frac{1}{n} \|z - z_{L_\gamma}\|^2 &= \lambda^2 z^T (L_\gamma + n\lambda I)^{-2} z \\
&\leq n\lambda^2 \left(1 - \frac{\gamma}{\lambda(1 - t_{S,Q})}\right)^{-2} z^T (K + n\lambda I)^{-2} z \\
&= \left(1 - \frac{\gamma}{\lambda(1 - t_{S,Q})}\right)^{-2} \frac{1}{n} \|z - z_K\|^2 \quad (4.12)
\end{aligned}$$

Koristeći sledeću činjenicu $(1 - \frac{\epsilon}{2})^{-2} \leq 1 + 3\epsilon$ gde je $\epsilon \in (0, 1)$ imamo

$$\frac{1}{n} \|z - z_{L_\gamma}\|^2 \leq \left(1 + 6\frac{\gamma}{\lambda(1 - t_{S,Q})}\right) \frac{1}{n} \|z - z_K\|^2$$

Sledi:

$$\frac{1}{n} E[\|z - z_{L_\gamma}\|^2] \leq \left(1 + 6\frac{\gamma}{\lambda(1 - E[t_{S,Q}])}\right) \frac{1}{n} \|z - z_K\|^2$$

Sada moramo pronaći gornju granicu za $E[t_{S,Q}]$.

$$\begin{aligned}
E[t_{S,Q}] &= E\left[\left\|\frac{1}{n}\Psi^T\Psi - \frac{1}{p}\Psi_S^T Q Q^T \Psi_S\right\|\right] \\
&= E\left[\left\|\frac{1}{n}\Psi^T\Psi - \frac{1}{p}\Psi_S^T \Psi_S + \frac{1}{p}\Psi_S^T \Psi_S - \frac{1}{p}\Psi_S^T Q Q^T \Psi_S\right\|\right] \\
&\leq E\left[\left\|\frac{1}{n}\Psi^T\Psi - \frac{1}{p}\Psi_S^T \Psi_S\right\|\right] + \frac{1}{p} E\left[\left\|\Psi_S^T \Psi_S - \Psi_S^T Q Q^T \Psi_S\right\|\right] \quad (4.13)
\end{aligned}$$

Sada nalazimo gornju granicu $E[\|\Psi_S^T \Psi_S - \Psi_S^T Q Q^T \Psi_S\|]$:

$$\begin{aligned}
E[\|\Psi_S^T \Psi_S - \Psi_S^T Q Q^T \Psi_S\|] &\leq E[\|(I - Q Q^T)\Psi_S \Psi_S^T\|] \\
&= E_S[E_Q[(I - Q Q^T)\Psi_S \Psi_S^T]] \\
&\leq E_S\left[\left(1 + \sqrt{\frac{k}{l-1}} + \frac{e\sqrt{k+l}}{l\sqrt{p-k}}\right) \frac{1}{\gamma} \|\Phi_S(\Phi_S^T)\| \right] \quad (4.14)
\end{aligned}$$

Poslednja nejednakost je izvedena iz tereme(4.6.3).

$$\begin{aligned}
E[|\Psi_S^T \Psi_S - \Psi_S^T Q Q^T \Psi_S|] &\leq \frac{1}{\gamma} \left(1 + \sqrt{\frac{k}{l-1}} + \frac{e\sqrt{k+l}}{l\sqrt{p-k}}\right) E_S[|\Phi_S \Phi_S^T|] \\
&\leq \frac{1}{\gamma} \left(1 + \sqrt{\frac{k}{l-1}} + \frac{e\sqrt{k+l}}{l}\right) \|\Phi \Phi^T\|
\end{aligned} \tag{4.15}$$

Primenom leme (4.6.1) i rezultata iz ([Bach, 2013]) za $\gamma \leq \lambda$ imamo:

$$\begin{aligned}
E\left[\left|\frac{1}{n} \Psi^T \Psi - \frac{1}{p} \Psi_S^T \Psi_S\right|\right] &\leq \sqrt{\frac{1}{p} \lambda \gamma^{-1} d 3 \log n} \frac{1}{p} \log n \\
E\left[\left|\frac{1}{n} \Psi^T \Psi - \frac{1}{p} \Psi_S^T \Psi_S\right|\right] &\leq \sqrt{\frac{3}{k+1} \lambda \gamma^{-1} d \log n} \frac{1}{p} \log n
\end{aligned}$$

Kombinujući poslednja dva rezultata imamo

$$E[t_{S,Q}] \leq \sqrt{\frac{3}{k+1} \lambda \gamma^{-1} d \log n} \frac{1}{p} \log n + \frac{1}{p\gamma} \left(1 + \sqrt{\frac{k}{l-1}} + \frac{e\sqrt{k+l}}{l}\right) R^2.$$

Ubacujući $t = 1 + \sqrt{\frac{k}{l-1}} + \frac{e\sqrt{k+l}}{l}$ dobijamo

$$E[t_{S,Q}] \leq \sqrt{\frac{3}{k+1} \lambda \gamma^{-1} d \log n} \frac{1}{p} \log n + \frac{tR^2}{p\gamma}.$$

Za $p \geq 2 \left(\log n \sqrt{\frac{6d \log n}{10\delta}} + \frac{tR^2 \lambda \delta}{2} \right)$ i za $\gamma = \frac{\lambda \delta}{2}$ (gde je $\gamma \leq \lambda$) imamo

$$E[t_{S,Q}] \leq \frac{1}{2}$$

Iz $\frac{\gamma}{\lambda(1-E[t_{S,Q}])} \leq \delta \leq 1$ sledi

$$\frac{1}{n} E[||z - z_{L_\gamma}||^2] \leq \left(1 + 6 \frac{\gamma}{\lambda(1-E[t_{S,Q}])}\right) \frac{1}{n} ||z - z_K||^2$$

odakle se izvodi željeni rezultat $\frac{1}{n} E[||z - z_{L_\gamma}||^2] \leq (1 + 6\delta) \frac{1}{n} ||z - z_K||^2$.

□

4.7 Eksperimentalni rezultati

Tabela 4.1: Skupovi podataka na kojima ćemo evaluirati algoritme kernel regresije sa slučajnim karakteristikama. Prikazujemo naziv skupa, broj primera u skupu i koliko atributa ima svaki skup.

Skup podataka	Instance	Atributi
cal housing ¹⁾	7154	40
abalone ²⁾	2089	6
bank8 ¹⁾	4096	8
bank32 ¹⁾	4096	32
sarcos2 ³⁾	44484	21
sarcos3 ³⁾	44484	21
sarcos4 ³⁾	44484	21
sarcos5 ³⁾	44484	21
sarcos6 ³⁾	44484	21
sarcos7 ³⁾	44484	21
red wine quality ²⁾	4898	12
white wine quality ²⁾	4898	12
Combined Cycle Power Plant Data Set ²⁾	9568	4
Facebook Comment Volume Dataset ²⁾	40949	54
Gas Turbine CO and NOx Emission ²⁾	36733	11
SGEMM GPU kernel performance ²⁾	241600	18
Superconductivity Data ²⁾	21263	81
census8H ¹⁾	22784	8
census16H ¹⁾	22784	16
kin ¹⁾	8192	33
pumadynH ¹⁾	8192	33
pumadynM ¹⁾	8192	33

U ovom odeljku procenjujemo učinak našeg algoritma **RNF** (izvedenog iz RNiström metoda, algoritam 10) upoređujući ga sa algoritmom **NF** (izvedenim iz Nistromovog metoda, algoritam 9).

Upoređujemo algoritme na skupovima podataka iz stvarnog sveta

1. <http://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html>
2. <https://archive.ics.uci.edu/ml/datasets.html>
3. <http://www.gaussianprocess.org/gpml/data/> .

Kompletan opis skupova podataka se nalazi u tabeli (4.1):

- Skup podataka Abalone ¹⁾ se može koristiti za obuku modela koji se koristi za predviđanje starosti školjke iz njenih merenja. Broj prstenova reza ljuske koji se vide kroz mikroskop određuje starost školjke. Međutim, s obzirom da je ovo dugotrajan proces, ovaj skup podataka sadrži i druga merenja do kojih je lakše doći, a koja se zatim koriste za predviđanje starosti.

- CalHousing ¹⁾ skup podataka se koristi za obuku modela koji se koristi za predviđanje cene kuća u Kaliforniji. Sadrži sledeće attribute: srednja vrednost kuće, srednji prihod građana, srednja starost stana, ukupan broj soba, ukupan broj spavaćih soba, stanovništvo, domaćinstvo, geografska širina i dužina.
- Bank8 (8 realnih atributa) i bank32 (32 realna atributa) ¹⁾ je familija skupova podataka sintetički generisanih iz simulacije u kojoj klijenti banaka biraju svoje banke. Model je obučen za predviđanje vrednosti odbijanja, što je udeo ljudi koji napuštaju banku, jer svi otvoreni šalteri imaju pune redove.
- Podaci u Sarcos skupu podataka ³⁾ se odnose na problem inverzne dinamike za kretanje antropomorfne robotske ruke SARCOS. Model je osposobljen za predviđanje obrtnih momenata analizom vektora iz 21-dimenzionalnog ulaznog prostora (7 položaja zglobova, 7 brzina zglobova, 7 ubrzanja zglobova). *Sarcos-i* se odnosi na predviđanje *i*-tog obrtnog momenta.
- Wine quality odnosi se na dva skupa podataka koji se odnose na crvenu i belu varijantu portugalskog vina "Vinho Verde". Model je obučen da predviđa kvalitet vina samo na osnovu fizičko-hemijskih (ulaznih) podataka. Podaci o vrstama grožđa, marki vina, prodajnoj ceni vina itd. se ne nalaze u skupu podataka.
- The Combined Cycle Power Plant ²⁾ skup podataka se formira prikupljanjem podataka iz elektrane sa kombinovanim ciklusom tokom 6 godina, kada je elektrana radila sa punim opterećenjem. Atributi su vrednosti temperature (T), pritiska oko line (AP), relativne vlažnosti (RH) i izduvnog vakuuma (V) prikupljeni na sat, a vrednost koja se predviđa je neto izlaz električne energije po satu (EP) iz postrojenja.
- Skup podataka Gas Turbine CO and NOx Emission ²⁾ dolazi iz iste elektrane. Međutim, ovi podaci sadrže 11 atributa koji su zbirne mere senzora tokom jednog sata i koriste se za predviđanje emisije gasova.
- Facebook Comment Volume skup podataka ²⁾ sadrži vektore karakteristika izvučene iz facebook postova, a atribut koji se predviđa je broj komentara koje će post dobiti.
- SGEMM GPU kernel performance ²⁾ skup podataka sadrži merenje vremena izvršenja izračunavanja matričnog proizvoda $AB = C$. Parametrizovani SGEMM GPU kernel se koristi i testira na svim izvodljivim kombinacijama parametara. Sve korišćene matrice su veličine 2048 x 2048. Za svaku testiranu kombinaciju urađena su 4 izvođenja i njihovi rezultati se u milisekundama prikazuju kao poslednje 4 kolone. Kao ulaz koristimo parametre i model je obučen da predviđa prosečno vreme izvršenja.
- Superconductivty skup podataka ²⁾ sadrži 81 karakteristiku izdvojenu iz 21263 superprovodnika koje predstavljaju ulaz. U 82. koloni se čuva kritična temperatura. Model je obučen za predviđanje kritične temperature.
- Census ¹⁾ skup podataka odnosi se na dva skupa podataka census8 (8 neprekidnih atributa) i census16 (16 neprekidnih atributa) i dizajniran je za analizu podataka

Američkog Popisnog biroa. Model je obučen da predviđa srednju cenu kuće.

- Kin (Kinematics) ¹⁾ skup podataka prikuplja se simuliranjem prednje kinematike 8-člane robotske ruke. Postoje razne varijante ovog skupa podataka, a mi koristimo onu sa 32 atributa, za koju se zna da je krajnje nelinearna i sa velikim šumom.
- Pumatyn ¹⁾ skup podataka je izabran iz porodice skupova podataka sintetički generisanih iz realne simulacije dinamike robotske ruke Unimation Puma 560. Postoje razne varijante ovog skupa podataka, a mi koristimo onu sa 32 atributa, u dve varijante: jako nelinearnu i sa velikim šumom (H) ili jako nelinearnu i sa srednjim šumom (M).

Neki od skupova podataka su već podeljeni na trening skup i test skup. One koji nisu, podelili smo na sledeći način: prvih 80% podataka koristi se za obuku, a preostalih 20% za testiranje. Koristili smo Gausov kernel čiji su hiperparametri izabrani pomoću petostruke unakrsne validacije na 1000 označenih primera trening skupa. Dimenzija slučajnih karakteristika je 10. U **NF** algoritmu, 10-dimenzionalni vektor slučajnih karakteristika izveden je iz 10 slučajno izabranih primera (kolone u Gramovoj matrici). Međutim u našem algoritmu **RNF** se izvodi 10-dimenzionalni slučajni vektor karakteristika sa dodatnom aproksimacijom, a ne samo iz Nistromovog metoda, od 50 odabranih primera.

Prikazujemo srednju grešku predikcije i njenu standardnu devijaciju izračunatu na test skupovima i izveštavamo rezultate u tabelama (4.3 i 4.2). U svim skupovima podataka **RNF** nadmašuje **NF**. Stoga primećujemo da se bolji rezultati postižu kada se koristi kombinacija Nistromove metode sa aproksimacijom SVD-a u poređenju samo sa Nistromovom metodom, dok vremenska složenost ostaje linearna po veličini skupa podataka.

Tabela 4.2: Poređenje performansi algoritama **RNF** i **NF** na skupovima podataka iz stvarnog sveta. Pokazujemo srednje vreme izvršenja u sekundama. Broj slučajnih karakteristika je 10, a broj izabranih kolona je 50.

Skup podataka	NF	RNF
calHousing	0.048	0.048
abalone	0.006	0.007
bank32	0.014	0.014
bank8	0.009	0.009
sarcos2	0.047	0.047
sarcos3	0.047	0.047
sarcos4	0.047	0.047
sarcos5	0.047	0.047
sarcos6	0.047	0.047
sarcos7	0.047	0.047
red wine quality	0.005	0.004
white wine quality	0.007	0.007
Combined Cycle Power Plant Data Set	0.008	0.008
Facebook Comment Volume Dataset	0.082	0.081
Gas Turbine CO and NOx Emission	0.008	0.008
SGEMM GPU kernel performance	0.233	0.235
Superconductivity Data	0.059	0.060
census8H	0.025	0.024
census16H	0.031	0.030
kin	0.015	0.015
pumadynH	0.017	0.016
pumadynM	0.015	0.015

Tabela 4.3: Poređenje performansi algoritama **RNF** i **NF** na skupovima podataka iz stvarnog sveta. Pokazujemo srednju grešku predikcije i njenu standardnu devijaciju. Broj slučajnih karakteristika je 10, a broj izabranih kolona je 50.

Skup podataka	NF	RNF
calHousing	0.672 ± 0.052	0.639 ± 0.032
abalone	0.640 ± 0.001	0.639 ± 0.000
bank32	0.510 ± 0.027	0.492 ± 0.005
bank8	0.059 ± 0.003	0.060 ± 0.003
sarcos2	0.307 ± 0.029	0.301 ± 0.020
sarcos3	0.248 ± 0.024	0.241 ± 0.025
sarcos4	0.083 ± 0.018	0.075 ± 0.008
sarcos5	0.498 ± 0.066	0.454 ± 0.037
sarcos6	0.694 ± 0.072	0.643 ± 0.038
sarcos7	0.101 ± 0.019	0.085 ± 0.007
red wine quality	0.913 ± 0.048	0.936 ± 0.057
white wine quality	0.931 ± 0.032	0.929 ± 0.014
Combined Cycle Power Plant Data Set	0.111 ± 0.042	0.086 ± 0.016
Facebook Comment Volume Dataset	1.131 ± 0.001	1.131 ± 0.001
Gas Turbine CO and NOx Emission	0.550 ± 0.178	0.393 ± 0.066
SGEMM GPU kernel performance	0.982 ± 0.091	0.946 ± 0.042
Superconductivity Data	1.630 ± 0.018	1.602 ± 0.018
census8H	1.596 ± 0.047	1.495 ± 0.031
census16H	1.728 ± 0.032	1.664 ± 0.025
kin	0.996 ± 0.003	0.997 ± 0.002
pumadynH	0.864 ± 0.017	0.841 ± 0.004
pumadynM	0.832 ± 0.016	0.807 ± 0.005

Glava 5

Zaključak

U glavi 2 smo rešavali problem spektralnog klasterovanja sa ograničenjima. Predložili smo i implementirali novi algoritam zasnovan na algoritmu spektralnog klasterovanja bez ograničenja na višeslojnom grafu. Algoritam koristi grafovski prikaz podataka i ograničenja različitim slojevima višeslojnog grafa, na koji se primenjuje algoritam spektralnog klasterovanja bez ograničenja na osnovu [Dong et al., 2014]. Proverili smo naš algoritam na skupovima podataka iz stvarnog sveta, sa različitim vrstama ograničenja (tvrđim, mekim, nedoslednim i sa šumom) i pokazali smo da postiže superiorne ili slične rezultate sa savremenim najboljim algoritmom [Wang et al., 2014a].

Dalje, razmotrili smo problem vremenske složenosti našeg algoritma, koja je kvadratna po broju instanci. Kao rešenje ovog problema predložili smo upotrebu metode Nistroma kojom se postiže linearno vreme po broju predmeta. Koliko nam je poznato ovo je jedini algoritam spektralnog klasterovanja sa ograničenjima takve složenosti koji se može primeniti i na tvrda i na meka ograničenja, bez obzira na veličinu samog skupa ograničenja.

U glavi 3 smo rešavali probleme koji nastaju kod spektralnog klasterovanja kada nemamo pristup dobrom skupu ograničenja i rešili smo ga implementacijom novog algoritma za spektralno klasterovanje koji se samostalno uči. Naš algoritam se može koristiti i za spektralno klasterovanje sa ograničenjima i bez ograničenja. Samoučenje se primenjuje kao samoučenje novih ograničenja, bez obzira na vrstu ulaza. Stoga, čak i slučaju bez ograničenja, veštački ih dodajemo kroz proces samoučenja. Algoritam je fleksibilan, u smislu da ograničenja mogu biti i meka i tvrda.

Algoritam postiže vremensku složenost linearnu po broju uzoraka i postiže memorijsku složenost koja je takođe linearna po broju uzoraka podataka. Ovo se postiže metodom Nistroma. Prema našim saznanjima, ovo je jedini algoritam spektralnog klasterovanja koji se samostalno uči sa ovom vremenskom i memorijskom složenošću. Uporedili smo ga sa drugim algoritmima (spektralnog klasterovanja sa ograničenjima) sa ovom složenošću i postigli smo bolje performanse.

U glavi 4 smo razmatrali problem ubrzavanja kernel regresije pomoću slučajnih Nistromovih karakteristika. Metode kernela generalno imaju kubnu vremensku složenost,

ali kada se koriste u kombinaciji sa slučajnim karakteristikama, ta složenost je linearna. U radu [Bach, 2013] je dokazano da je greška regresije koja koristi Nistromove vektore karakteristika proizvoljno blizu greške kernel regresije za dovoljno veliki broj uzorkovanih kolona. Ovde smo analizirali grešku regresije koja koristi RNistrom karakteristike (kombinacija Nistromove metode i aproksimacije svd-a). Ulazni skup

$$\{(x_i \in \mathbf{R}^d)_{i=1}^n\}$$

preslikava se u skup m -dimenzionalnih vektora slučajnih karakteristika

$$\{(r_i \in \mathbf{R}^m)_{i=1}^n\}$$

izveden korišćenjem skupa od p ulaznih vektora $\{\hat{x}_i\}_{i=\overline{1,p}}$ nasumično uzorkovanih iz ulaznog skupa. RNistrom karakteristike računaju se kao

$$r_i = \hat{D}^{-1/2} \hat{V}^T k(x_{1:n}, \hat{x}_{1:p})^T,$$

gde su kolone matrice $\hat{V} \in \mathbb{R}^{p \times m}$ približni sopstveni vektori i dijagonalni elementi matrice $\hat{D} \in \mathbb{R}^{m \times m}$ su približne sopstvene vrednosti matrice $k(\hat{x}_{1:p}, \hat{x}_{1:p})$, izračunati iz aproksimativnog spektralnog razlaganja. Prema tome, mi smo dodali još jednu aproksimaciju Nistromovoj metodi i pokazali da se glavno svojstvo Nistromovih karakteristika zadržava, tj. da za svaku grešku ϵ postoji dovoljno veliko p tako da je greška koju pravi regresor naučen na ulaznim vektorima r_i manja od ϵ . Pored toga, empirijski smo pokazali da je upotreba RNistromove metode bolja za računanje m -dimenzionalnih vektora karakteristika od Nistromove metode, upoređujući njihove performanse na stvarnim skupovima podataka.

Rezultati predstavljeni u glavama (2 i 4) su objavljeni u radovima:

- A. Trokicić and B. Todorović. On expected error of randomized nystrom kernel regression. *Filomat*, 34(11):3871–3884, 2020
- A. Trokicić and B. Todorović. Randomized nystrom features for fast regression: An error analysis. In *International Conference on Algebraic Informatics*, pages 249–257. Springer, 2019b
- A. Trokicić and B. Todorović. Constrained spectral clustering via multilayer graph embeddings on a grassmann manifold. *International Journal of Applied Mathematics and Computer Science*, 29(1):125–137, 2019a

Bibliografija

- E. Abbe. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531, 2017.
- D. Achlioptas and F. McSherry. Fast computation of low-rank matrix approximations. *Journal of the ACM (JACM)*, 54(2):9–es, 2007.
- C. C. Aggarwal and H. Wang. A survey of clustering algorithms for graph data. In *Managing and mining graph data*, pages 275–301. Springer, 2010.
- F. Bach. Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory*, pages 185–209, 2013.
- F. R. Bach and M. I. Jordan. Predictive low-rank decomposition for kernel methods. In *Proceedings of the 22nd international conference on Machine learning*, pages 33–40. ACM, 2005.
- C. T. Baker. The numerical treatment of integral equations. 1977.
- S. Basu, I. Davidson, and K. Wagstaff. *Constrained clustering: Advances in algorithms, theory, and applications*. CRC Press, 2008.
- M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.
- T. D. Bie and N. Cristianini. Fast sdp relaxations of graph cut clustering, transduction, and other combinatorial problems. *The Journal of Machine Learning Research*, 7: 1409–1436, 2006.
- Y. Chen and E. J. Candès. The projected power method: An efficient algorithm for joint alignment from pairwise differences. *Communications on Pure and Applied Mathematics*, 71(8):1648–1714, 2018.
- Y. Chen, J. Fan, C. Ma, and K. Wang. Spectral method and regularized mle are both optimal for top-k ranking. *Annals of statistics*, 47(4):2204, 2019.
- Y. Chen, Y. Chi, J. Fan, and C. Ma. Spectral methods for data science: A statistical perspective. *arXiv preprint arXiv:2012.08496*, 2020.
- A. Choromanska, T. Jebara, H. Kim, M. Mohan, and C. Monteleoni. Fast spectral clustering via the Nyström method. In *International Conference on Algorithmic Learning Theory*, pages 367–381. Springer, 2013.

- F. R. Chung and F. C. Graham. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.
- T. Coleman, J. Saunderson, and A. Wirth. Spectral clustering with inconsistent advice. In *Proceedings of the 25th International Conference on Machine Learning*, pages 152–159. ACM, 2008.
- W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Self-taught clustering. In *Proceedings of the 25th international conference on Machine learning*, pages 200–207. ACM, 2008.
- T. De Bie, J. Suykens, and B. De Moor. Learning from general label constraints. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 671–679. Springer, 2004.
- W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. In *Selected Papers Of Alan J Hoffman: With Commentary*, pages 437–442. World Scientific, 2003.
- X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov. Clustering on multi-layer graphs via subspace analysis on Grassmann manifolds. *IEEE Transactions on Signal Processing*, 62(4):905–918, 2014.
- J. Fan, K. Wang, Y. Zhong, and Z. Zhu. Robust high dimensional factor models with applications to statistical machine learning. *arXiv preprint arXiv:1808.03889*, 2018.
- J. Fan, K. Li, and Y. Liao. Recent developments on factor models and its applications in econometric learning. *arXiv preprint arXiv:2009.10103*, 2020.
- M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.
- C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nystrom method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225, 2004.
- A. Gittens. The spectral norm error of the naive nystrom extension. *arXiv preprint arXiv:1110.5305*, 2011.
- G. H. Golub and C. F. Van Loan. Matrix computations. 1996. *Johns Hopkins University, Press, Baltimore, MD, USA*, pages 374–426, 1996.
- D. Gross and V. Nesme. Note on sampling without replacing from a finite collection of matrices. *arXiv preprint arXiv:1001.2738*, 2010.
- L. Hagen and A. B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE transactions on computer-aided design of integrated circuits and systems*, 11(9):1074–1085, 1992.
- N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

- J. Hamm and D. D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 376–383. ACM, 2008.
- J. Hamm and D. D. Lee. Extended Grassmann kernels for subspace-based learning. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 601–608. Curran Associates, Inc., 2009.
- M. T. Harandi, C. Sanderson, S. Shirazi, and B. C. Lovell. Graph embedding discriminant analysis on Grassmannian manifolds for improved image set matching. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2705–2712. IEEE, 2011.
- T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.
- F. Höppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy cluster analysis: methods for classification, data analysis and image recognition*. John Wiley & Sons, 2000.
- K. Kalecky and Y.-R. Cho. Primalign: Pagerank-inspired markovian alignment for large biological networks. *Bioinformatics*, 34(13):i537–i546, 2018.
- S. D. Kamvar, D. Klein, and C. D. Manning. Spectral learning. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 561–566, 2003.
- N. Kishore Kumar and J. Schneider. Literature survey on low rank approximation of matrices. *Linear and Multilinear Algebra*, 65(11):2212–2244, 2017.
- S. Kumar, M. Mohri, and A. Talwalkar. Sampling techniques for the Nyström method. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, pages 304–311, 2009.
- S. Kumar, M. Mohri, and A. Talwalkar. Sampling methods for the nyström method. *Journal of Machine Learning Research*, 13(Apr):981–1006, 2012.
- H. C. Law. Clustering, dimensionality reduction, and side information. Technical report, Michigan State Univ East Lansing Dept of Computer science/Engineering, 2006.
- Q. Le, T. Sarlós, and A. Smola. Fastfood-approximating kernel expansions in loglinear time. In *Proceedings of the international conference on machine learning*, volume 85, 2013.
- J. Li, Y. Xia, Z. Shan, and Y. Liu. Scalable constrained spectral clustering. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):589–593, 2015a.
- M. Li, X.-C. Lian, J. T. Kwok, and B.-L. Lu. Time and space efficient spectral clustering via column sampling. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2297–2304. IEEE, 2011.
- M. Li, W. Bi, J. T. Kwok, and B.-L. Lu. Large-scale nyström kernel matrix approximation using randomized svd. *IEEE transactions on neural networks and learning systems*, 26(1):152–164, 2015b.

- Z. Li, J. Liu, and X. Tang. Constrained clustering via spectral regularization. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 421–428. IEEE, 2009.
- M. Lichman. Uci machine learning repository, 2013. URL <http://archive.ics.uci.edu/ml>.
- Z. Lu and M. A. Carreira-Perpinan. Constrained spectral clustering through affinity propagation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.
- L. Mackey, M. I. Jordan, R. Y. Chen, B. Farrell, J. A. Tropp, et al. Matrix concentration inequalities via the method of exchangeable pairs. *The Annals of Probability*, 42(3): 906–945, 2014.
- M. W. Mahoney. Randomized algorithms for matrices and data. *arXiv preprint arXiv:1104.5557*, 2011.
- M. W. Mahoney and P. Drineas. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.
- C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- B. McWilliams, D. Balduzzi, and J. M. Buhmann. Correlated random features for fast semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 440–448, 2013.
- B. Mohar. Some applications of laplace eigenvalues of graphs. In *Graph symmetry*, pages 225–275. Springer, 1997.
- B. Mohar, Y. Alavi, G. Chartrand, and O. Oellermann. The laplacian spectrum of graphs. *Graph theory, combinatorics, and applications*, 2(871-898):12, 1991.
- M. E. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006.
- A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 2:849–856, 2002.
- S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.
- A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.
- A. Rahimi and B. Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in neural information processing systems*, pages 1313–1320, 2009.
- R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: transfer

- learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, pages 759–766. ACM, 2007.
- B. Schölkopf, A. J. Smola, F. Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002a.
- B. Schölkopf, A. J. Smola, et al. Learning with kernels: Support vector machines, regularization. *Optimization, and Beyond*. MIT press, 1(2), 2002b.
- B. Schölkopf, K. Tsuda, and J.-P. Vert. *Support vector machine applications in computational biology*. MIT press, 2004.
- J. Shawe-Taylor, N. Cristianini, et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004.
- J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.
- K. Sinha and M. Belkin. Semi-supervised learning using sparse eigenfunction bases. In *2009 AAAI Fall Symposium Series*. Citeseer, 2009.
- M. Stoer and F. Wagner. A simple min-cut algorithm. *Journal of the ACM (JACM)*, 44(4):585–591, 1997.
- A. Trokicić and B. Todorović. Constrained spectral clustering via multilayer graph embeddings on a grassmann manifold. *International Journal of Applied Mathematics and Computer Science*, 29(1):125–137, 2019a.
- A. Trokicić and B. Todorović. Randomized nyström features for fast regression: An error analysis. In *International Conference on Algebraic Informatics*, pages 249–257. Springer, 2019b.
- A. Trokicić and B. Todorović. On expected error of randomized nystrom kernel regression. *Filomat*, 34(11):3871–3884, 2020.
- J. A. Tropp. Improved analysis of the subsampled randomized hadamard transform. *Advances in Adaptive Data Analysis*, 3(01n02):115–126, 2011.
- J. A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434, 2012.
- M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
- U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.
- D. Wagner and F. Wagner. Between min cut and graph bisection. In *International Symposium on Mathematical Foundations of Computer Science*, pages 744–750. Springer, 1993.
- X. Wang, 2014. URL <https://sites.google.com/site/gnaixgnaw/home>.

- X. Wang and I. Davidson. Active spectral clustering. In *2010 IEEE International Conference on Data Mining*, pages 561–568. IEEE, 2010.
- X. Wang, B. Qian, and I. Davidson. On constrained spectral clustering and its applications. *Data Mining and Knowledge Discovery*, 28(1):1–30, 2014a.
- X. Wang, J. Wang, B. Qian, F. Wang, and I. Davidson. Self-taught spectral clustering via constraint augmentation. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 416–424. SIAM, 2014b.
- F. L. Wauthier, N. Jojic, and M. I. Jordan. Active spectral clustering via iterative uncertainty reduction. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1339–1347. ACM, 2012.
- S. White and P. Smyth. A spectral clustering approach to finding communities in graphs. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 274–285. SIAM, 2005.
- C. K. Williams and C. E. Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.
- C. K. Williams and M. Seeger. Using the nyström method to speed up kernel machines. In *Advances in neural information processing systems*, pages 682–688, 2001.
- Q. Xu, M. des Jardins, and K. Wagstaff. Constrained spectral clustering under a local proximity structure assumption. In *In Proceedings of the 18th International Conference of the Florida Artificial Intelligence Research Society (FLAIRS-05)*, pages 866–867, 2005a.
- Q. Xu, K. L. Wagstaff, et al. Active constrained clustering by examining spectral eigenvectors. In *International Conference on Discovery Science*, pages 294–307. Springer, 2005b.
- T. Yang, Y.-F. Li, M. Mahdavi, R. Jin, and Z.-H. Zhou. Nyström method vs random fourier features: A theoretical and empirical comparison. In *Advances in neural information processing systems*, pages 476–484, 2012.

Dodatak A

Biografija

Aleksandar Trokicić je rođen 6. 3. 1989. godine u Nišu gde je završio Osnovnu školu "Bubanjski heroji" i Gimnaziju "Svetozar Marković", odeljenje za talentovane matematičare. Osnovne studije na Prirodno-matematičkom fakultetu, Univerziteta u Nišu upisao je školske 2007/2008. godine, koje je završio 2010. godine sa prosečnom ocenom 10/10. Školske 2010/2011. godine upisao je master akademske studije na istom fakultetu, koje je završio 2012. godine sa prosečnom ocenom 10/10. Doktorske studije upisao je po završetku master studija. Master rad pod nazivom "Ugrađivanje Skrivenih Markovljevih modela u Hilbertov prostor" odbranio je sa najvišom ocenom.

Za vreme srednje škole učestvovao je i dobijao nagrade na domaćim (republičkim i saveznim) takmičenjima iz matematike (dve prve i dve treće nagrade), fizike (jedna druga i jedna treća nagrada) i informatike (jedna druga i dve treće nagrade). Na inostranim takmičenjima je dobio sledeće medalje: bronzanu medalju na *Internacionalnoj matematičkoj olimpijadi* u Sloveniji 2006. godine, bronzanu medalju na Balkanskoj matematičkoj olimpijadi na Kipru 2006. godine i srebrnu medalju na Juniorskoj Balkanskoj matematičkoj olimpijadi 2004. godine. U toku studija učestvovao je na sledećim internacionalnim takmičenjima iz informatike: *ACM internacionalnom studentskom timskom takmičenju iz informatike* (2010, 2011) i Bubble Cup internacionalnom studentskom timskom takmičenju iz informatike (2009, 2010, 2011).

Od 2014. angažovan je kao asistent na Prirodno-matematičkom fakultetu, Univerziteta u Nišu, gde izvodi ili je izvodio vežbe na predmetima: Računarska Inteligencija, Teorija programskih jezika, Softversko inženjerstvo, Testiranje i metrika softvera, Funkcionalno programiranje, Dizajn softvera i Računarska grafika. Kao istraživač, angažovan je od 2014. godine u okviru projekta "Razvoj metoda za izračunavanja i procesiranje informacija: teorija i primene" (br. 174013).

Držao je nastavu u drugom razredu specijalizovanog matematičkog odeljenja Gimnazije "Svetozar Marković" u Nišu iz predmeta "Računarstvo i Informatika" u toku 2015/2016 godine u trajanju od jednog polugodišta.

Učestvovao je na sledećim seminarima: *Istraživačka stanica Petnica 2004–2009*, kao učesnik seminara matematike, a posle i kao mlađi saradnik na seminaru; DAAD Seminar

- "Graph Spectra and Applications" 2012. godine.

Učestvovao je na sledećim konferencijama: "The Fourth Conference on Information Theory and Complex Systems" TINKOS 2016, "The Fifth Conference on Information Theory and Complex Systems" TINKOS 2017, "Conference on Algebraic Informatics, 8th International Conference", Niš, Serbia, June 30-July 4, 2019 CAI 2019.

Objavio je sledeće radove:

- A. Trokicić and B. Todorović. On expected error of randomized nystrom kernel regression. *Filomat*, 34(11):3871–3884, 2020
- A. Trokicić and B. Todorović. Randomized nystrom features for fast regression: An error analysis. In *International Conference on Algebraic Informatics*, pages 249–257. Springer, 2019b
- A. Trokicić and B. Todorović. Constrained spectral clustering via multilayer graph embeddings on a grassmann manifold. *International Journal of Applied Mathematics and Computer Science*, 29(1):125–137, 2019a
- A. Trokicić and B. Todorović. Error analysis of a kernel regression based on a randomized matrix approximation. *The Fifth Conference on Information Theory and Complex Systems*, TINKOS 2017, November 9-10, 2017
- A. Trokicić and B. Todorović. Nystrom views via the randomized SVD for semi supervised learning. *The Fourth Conference on Information Theory and Complex Systems*, TINKOS 2016, October 27-28, 2016
- A. Trokicić. Approximate spectral learning using Nystrom method. *Facta Universitatis, Series: Mathematics and Informatics*, Vol. 31 No 2 569–578, 2016

Подаци о докторској дисертацији


Ментор:	др Бранимир Годоровић, ванредни професор, Природно-математички факултет, Универзитет у Нишу
Наслов:	Алгоритми за брзо апроксимативно спектрално учење
Резиме:	<p>У тези се изучавају алгоритми за тренирање које користе информације које се налазе у спектру матрице (сопствени вектори и сопствене вредности) добијене из улазног скупа. Матрице које ће бити посматране су графовске матрице или кернел матрице, израчунате из улазног скупа. Како алгоритми који користе те матрице имају или квадратну или кубну временску сложеност и квадратну меморијску сложеност по броју елемената улазног скупа, у овој тези биће представљени алгоритми који апроксимирају те матрице и који имају линеарну сложеност. Алгоритми ће бити упоређени са другим алгоритмима који решавају исте проблеме и анализираће се њихове се њихове експерименталне и теоријске карактеристике</p>
Научна област:	Рачунарске науке
Научна дисциплина:	Вештачка интелигенција, машинско учење
Кључне речи:	кластероване, кернел регресија, спектралне методе, апроксимација, Нистромова метода, Лапласова матрица
УДК:	004.8(043.3)
CERIF класификација:	P 176 Вештачка Интелигенција
Тип лиценце Креативне заједнице:	CC BY-NC-ND

Data on Doctoral Dissertation

Doctoral Supervisor:	Branimir Todorović , PhD, Assistant Professor at the Faculty of Sciences and Mathematics, University of Niš
Title:	Algorithms for fast approximate spectral learning
Abstract:	<p>This thesis presents learning algorithms which use the information stored in the spectrum (eigenvalues and eigenvectors) of a matrix derived from the input set. Matrices in question are graph matrices or kernel matrices. However, the algorithms which use these matrices have either a quadratic or cubic time complexity and quadratic memory complexity. Therefore, in this thesis the algorithms will be presented that approximate those matrices and reduce the time and memory complexity to the linear one. Also, these algorithms will be compared with the other algorithms that solve this problem, and their empirical and theoretical analysis will be presented.</p>
Scientific Field:	Computer Science
Scientific Discipline:	Artificial Intelligence; Machine Learning
Key Words:	clustering, kernel regression, spectral methods, approximation, Nystrom method, Laplacian matrix
UDC:	004.8(043.3)
CERIF Classification:	P 176 Artificial Intelligence

Creative
Commons
License Type:

CC BY-NC-ND

	ПРИРОДНО - МАТЕМАТИЧКИ ФАКУЛТЕТ НИШ
	КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:	
Идентификациони број, ИБР:	
Тип документације, ТД:	монографска
Тип записа, ТЗ:	текстуални / графички
Врста рада, ВР:	докторска дисертација
Аутор, АУ:	Александар Трокицић
Ментор, МН:	Бранимир Тодоровић
Наслов рада, НР:	Алгоритми за брзо апроксимативно спектрално учење
Језик публикације, ЈП:	српски
Језик извода, ЈИ:	енглески
Земља публикавања, ЗП:	Р. Србија
Уже географско подручје, УГП:	Р. Србија
Година, ГО:	2021.
Издавач, ИЗ:	ауторски репринт
Место и адреса, МА:	Ниш, Вишеградска 33.
Физички опис рада, ФО: <small>(поглавља/страна/ цитата/табела/слика/графика/прилога)</small>	115 стр., граф. прикази
Научна област, НО:	Рачунарске науке
Научна дисциплина, НД:	Вештачка интелигенција, машинско учење
Предметна одредница/Кључне речи, ПО:	кластеровање, кернел регресија, спектралне методе, апроксимација, Нистромова метода, Лапласова матрица
УДК	004.8(043.3)
Чува се, ЧУ:	библиотека
Важна напомена, ВН:	
Извод, ИЗ:	<p>У тези се изучавају алгоритми за тренирање које користе информације које се налазе у спектру матрице (сопствени вектори и сопствене вредности) добијене из улазног скупа. Матрице које ће бити посматране су графовске матрице или кернел матрице, израчунате из улазног скупа. Како алгоритми који користе те матрице имају или квадратну или кубну временску сложеност и квадратну меморијску сложеност по броју елемената улазног скупа, у овој тези биће представљени алгоритми који апроксимирају те матрице и који имају линеарну сложеност. Алгоритми ће бити упоређени са другим алгоритмима који решавају исте проблеме и анализираће се њихове се њихове експерименталне и теоријске карактеристике.</p>

Датум прихватања теме, **ДП**:

Датум одбране, **ДО**:

Чланови комисије, **КО**: Председник:


Члан:

Члан

Члан:

Члан, менто

Образац Q4.09.13 - Издање 1

	ПРИРОДНО - МАТЕМАТИЧКИ ФАКУЛТЕТ НИШ
	KEY WORDS DOCUMENTATION

Accession number, ANO :	
Identification number, INO :	
Document type, DT :	monograph
Type of record, TR :	textual / graphic
Contents code, CC :	doctoral dissertation
Author, AU :	Aleksandar (B) Trokicić
Mentor, MN :	Branimir Todorović
Title, TI :	Algorithms for fast approximate spectral learning
Language of text, LT :	Serbian
Language of abstract, LA :	English
Country of publication, CP :	Republic of Serbia
Locality of publication, LP :	Serbia
Publication year, PY :	2021
Publisher, PB :	author's reprint
Publication place, PP :	Niš, Višegradska 33.
Physical description, PD : <small>(chapters/pages/ref./tables/pictures/graphs/appendixes)</small>	115 p. ; graphic representations
Scientific field, SF :	Computer Science
Scientific discipline, SD :	Artificial Intelligence; Machine Learning
Subject/Key words, S/KW :	clustering, kernel regression, spectral methods, approximation, Nystrom method, Laplacian matrix
UC	004.8(043.3)
Holding data, HD :	library
Note, N :	
Abstract, AB :	<p>This thesis presents learning algorithms which use the information stored in the spectrum (eigenvalues and eigenvectors) of a matrix derived from the input set. Matrices in question are graph matrices or kernel matrices. However, the algorithms which use these matrices have either a quadratic or cubic time complexity and quadratic memory complexity. Therefore, in this thesis the algorithms will be presented that approximate those matrices and reduce the time and memory complexity to the linear one. Also, these algorithms will be compared with the other algorithms that solve this problem, and their empirical and theoretical analysis will be presented.</p>
Accepted by the Scientific Board on, ASB :	
Defended on, DE :	

Defended Board, **DB**:

President:

Member:

Member:

Member:

Member, Mentor:

Образац Q4.09.13 - Издање 1

ИЗЈАВА О АУТОРСТВУ

Изјављујем да је докторска дисертација, под насловом

АЛГОРИТМИ ЗА БРЗО АПРОКСИМАТИВНО СПЕКТРАЛНО УЧЕЊЕ

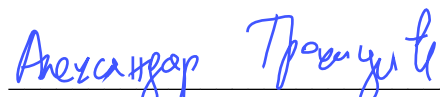
која је одбрањена на Природно-математичком факултету Универзитета у Нишу:

- резултат сопственог истраживачког рада;
- да ову дисертацију, ни у целини, нити у деловима, нисам пријављивао/ла на другим факултетима, нити универзитетима;
- да нисам повредио/ла ауторска права, нити злоупотребио/ла интелектуалну својину других лица.

Дозвољавам да се објаве моји лични подаци, који су у вези са ауторством и добијањем академског звања доктора наука, као што су име и презиме, година и место рођења и датум одбране рада, и то у каталогу Библиотеке, Дигиталном репозиторијуму Универзитета у Нишу, као и у публикацијама Универзитета у Нишу.

У Нишу, 16 03. 2021.

Потпис аутора дисертације:


Александар Б. Трокић

**ИЗЈАВА О ИСТОВЕТНОСТИ ШТАМПАНОГ И ЕЛЕКТРОНСКОГ ОБЛИКА
ДОКТОРСКЕ ДИСЕРТАЦИЈЕ**

Наслов дисертације:

АЛГОРИТМИ ЗА БРЗО АПРОКСИМАТИВНО СПЕКТРАЛНО УЧЕЊЕ

Изјављујем да је електронски облик моје докторске дисертације, коју сам предао/ла за уношење у **Дигитални репозиторијум Универзитета у Нишу**, истоветан штампаном облику.

У Нишу, 16 03. 2021.

Потпис аутора дисертације:



Александар Б. Трокић

ИЗЈАВА О КОРИШЋЕЊУ

Овлашћујем Универзитетску библиотеку „Никола Тесла“ да у Дигитални репозиторијум Универзитета у Нишу унесе моју докторску дисертацију, под насловом:

НАСЛОВ ДОКТОРСКЕ ДИСЕРТАЦИЈЕ

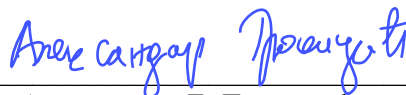
Дисертацију са свим прилозима предао/ла сам у електронском облику, погодном за трајно архивирање.

Моју докторску дисертацију, унету у Дигитални репозиторијум Универзитета у Нишу, могу користити сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons), за коју сам се одлучио/ла.

1. Ауторство (CC BY)
2. Ауторство – некомерцијално (CC BY-NC)
3. Ауторство – некомерцијално – без прераде (CC BY-NC-ND)
4. Ауторство – некомерцијално – делили под истим условима (CC BY-NC-SA)
5. Ауторство – без прераде (CC BY-ND)
6. Ауторство – делили под истим условима (CC BY-SA)

У Нишу, 16. 03. 2021 .

Потпис аутора дисертације:



Александар Б. Трокић