VIŠEKLASNO SPEKTRALNO GRUPIRANJE

Babić Marija
Runac Borna
Stanišić Matea
Vajdić Ivan

Prosinac 2019.

Sadržaj

1 Uvod

Metoda spektralnog particioniranja grafa primjenjuje se na nacrt strujnog kruga, balansiranje opterećenja i segmentaciju slika. Ne pretpostavljamo ništa o globalnoj strukturi podataka, već prikupljamo lokalni dokaz o tome koliko dva slična podatka pripadaju istoj klasi te donosimo globalnu odluku da se svi podaci podijele u disjunktne skupove po nekom kriteriju.

Ono što čini spektralnu metodu privlačnom je to što je globalni optimum u kontinuiranoj domeni dobiven preko svojstvene dekompozicije. Međutim, da bi dobili diskretno rješenje od svojstvenih vektora, često trebamo riješiti i drugi problem klasteriranja, ali u prostoru manje dimenzije. Odnosno, svojstvene vektore tretiramo kao koordinate skupa točaka. Razne heuristike klastera kao što su k-means, dinamičko programiranje, „greedy prunig" ili iscrpna pretraga naknadno se koriste na novoj točki kako bi se odredile particije.

Pokazat ćemo da postoji iscrpan način za otkrivanje diskretnog optimuma. To je bazirano na činjenici da se kontinuirani optimum sastoji ne samo od svojstvenih vrijednosti, nego od cijele familije razapete svojstvenim vektorima kroz ortonormirane transformacije. Cilj je pronaći pravu orotonormiranu transformaciju koja vodi do diskretizacije.

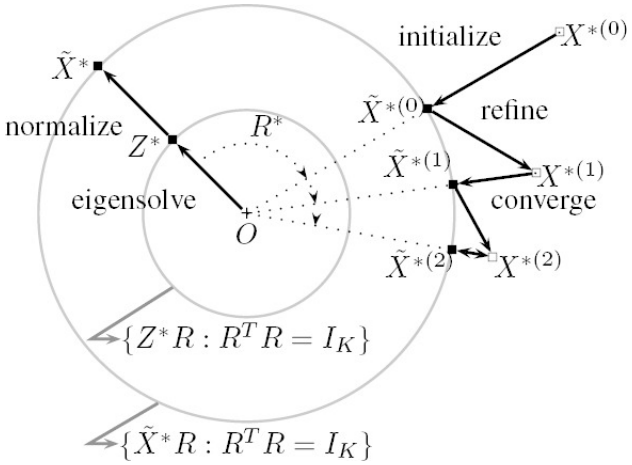
Metoda kojom rješavamo problem, sastoji se od dva koraka.

U prvom koraku riješimo pojednostavljeni neprekidni optimizacijski problem. Globalni optimumi su dani od strane svojstvenih vektora koji podliježu proizvoljnim ortonormiranim transformacijama.

U drugom koraku iterativno dobijemo diskretno rješenje koje je najbliže neprekidnim optimumima. Pri tom koristimo alternirajći postupak optimizacije: neprekidni optimum najbliži diskretnom rješenju, pronađe se traženjem najbolje ortonormirane transformacije, a diskretno rješenje najbliže neprekidnom se pronađe ne-maksimalnom supresijom.

Ovakvim se iteracijama smanjuje udaljenost izmedu diskretnog i neprekidnog optimuma. Nakon konvergencije, dobijemo gotovo globalno optimalnu particiju.

Kako bi još bolje shvatili naš algoritam, pojasnit ćemo ga na njegovom shematskom dijagramu:



U prvom koraku dobijemo svojstvene vektore Z^*. Prikazano na slici kao unutarnji krug, Z^* generira cijelu familiju globalnih optimuma kroz ortonormiranu transformaciju R. Poslije normiranja duljine, svaki optimum odgovara podjeljenom rješenju u kontinuiranoj domeni koju predstavlja vanjski krug.

Nakon toga slijedi drugi korak, u kojem iteracijama dobijemo diskretno rješenje nabliže neprekidnom optimumu. Počinjemo s diskretnim rješenjem $X^{*(0)}$, te nađemo $\widetilde{X}^{*(0)}$ računajući R^* koji najviše približava $\widetilde{X}^{*(0)}$ do $X^{*(0)}$. Preko neprekidnog optimuma $\widetilde{X}^{*(0)}$ pronađemo diskretnu verziju i tako dalje. Algoritam konvergira prema paru rješenja ($X^{*(2)}$, $\widetilde{X}^{*(2)}$) koji su najbliži jedno drugome. Optimalnost $\widetilde{X}^{*(2)}$ garantira da je $X^{*(2)}$ blizu globalnog optimuma.

2 Višeklasni normalizirani rezovi

Težinski graf dan je s $\mathbb{G}$ = ($\mathbb{V}$, $\mathbb{E}$, W), gdje je $\mathbb{V}$ skup svih čvorova, $\mathbb{E}$ skup svih bridova koji povezuju dva čvora, a W je matrica afiniteta, sa težinama koje opisuju kolika je vjerojatnost da dva čvora pripadaju istoj grupi. W je po pretpostavci nenegativna i simetrična matrica.

Neka $[n]$ označava skup prirodnih brojeva od 1 do n, tj. $[n] = \{1, 2, .., n\}$. Neka $\mathbb{V} = [N]$ označava skup svih elemenata koje želimo grupirati. Za grupiranje N točaka u K grupa trebamo naći dekompoziciju skupa $\mathbb{V}$ na K disjunktnih skupova, odnosno želimo da vrijedi $\mathbb{V} = \cup_{l=1}^{K} \mathbb{V}_l$ te $\mathbb{V}_l \cap \mathbb{V}_k = \emptyset$ za $k \neq l$. Navedeni problem nazivamo *K-particioniranje*.

2.1 Definiranje problema

Za dva podskupa skupa svih čvorova, $\mathbb{A}, \mathbb{B} \subset \mathbb{V}$, definirat ćemo povezanost (engl. *links*) među njima kao

$$\text{links}(\mathbb{A}, \mathbb{B}) = \sum_{i \in A, j \in B} W(i, j).$$

Stupanj (engl. *degree*) skupa $\mathbb{A}$ definirat ćemo kao povazanost skupa sa svim čvorovima

$$\text{degree}\left(\mathbb{A}\right) = \text{links}\left(\mathbb{A}, \mathbb{V}\right).$$

Stupanj skupa ćemo koristiti kao pojam veličine (*norme*) skupa, pa prema tome definiramo

$$\text{linkratio}\left(\mathbb{A}, \mathbb{B}\right) = \frac{\text{links}\left(\mathbb{A}, \mathbb{B}\right)}{\text{degree}\left(\mathbb{A}\right)}.$$

Linkratio nam zapravo označava udio veza skupa $\mathbb{A}$ sa $\mathbb{B}$ među svim vezama koje skup $\mathbb{A}$ ima. Omjer $\text{linkratio}\left(\mathbb{A}, \mathbb{A}\right)$ nam govori koliko veza ostaje unutar samog skupa $\mathbb{A}$, dok omjer $\text{linkratio}\left(\mathbb{A}, \mathbb{V} \setminus \mathbb{A}\right)$ nam govori koliko veza pobjegne iz skupa $\mathbb{A}$. Da bi dobro grupirali naše podatke, povezanost unutar jedne particije mora bit što bolja, a povezanost između dvije particije što slabija. Ova dva cilja možemo prikazati pomoću sljedeća dva kriterija:

$$\text{knassoc}\left(\Gamma_{\mathbb{V}}^{K}\right) = \frac{1}{k} \sum_{l=1}^{K} \text{linkratio}\left(\mathbb{V}_l, \mathbb{V}_l\right)$$

$$\text{kncuts}\left(\Gamma_{\mathbb{V}}^{K}\right) = \frac{1}{k} \sum_{l=1}^{K} \text{linkratio}\left(\mathbb{V}_l, \mathbb{V} \setminus \mathbb{V}_l\right)$$

gdje je $\Gamma_{\mathbb{V}}^{K} = \{\mathbb{V}_l : l = 1, \ldots, K\}$ jedna particija skupa $\mathbb{V}$ na K skupova. Kriterij *knassoc* nazivamo *K-normalizirane asocijacije*, dok *kncuts* nazivamo *K-normalizirani rezovi*. Direktno iz ove dvije definicije vidimo da vrijedi

$$\text{knassoc}\left(\Gamma_{\mathbb{V}}^{K}\right) + \text{kncuts}(\Gamma_{\mathbb{V}}^{K}) = 1,$$

što znači da su maksimizacija asocijacija (knassoc) i minimizacija rezova (kncuts) ekvivalentni. Kod minimizacije rezova nekoliko izoliranih čvorova može jako pokvariti particiju, dok je maksimizacija asocijacija robusnija u odnosu na perturbaciju težina matrice W. Definiramo ciljnu funkciju našeg problema K-particioniranja kao

$$\varepsilon\left(\Gamma_{\mathbb{V}}^{K}\right) = \text{knassoc}\left(\Gamma_{\mathbb{V}}^{K}\right).$$

Vrijedi da je $\varepsilon \in [0, 1]$ bez obzira na K.

Za bilo koji kriterij K-particioniranja moramo provjeriti njegovo ponašanje u odnosu na K. To znači da želimo provjeriti kako se ponaša kad mijenjamo K i povećava li se točnost kako se K povećava. Pokazat ćemo da se gornja međa od ε monotono smanjuje kako se K povećava.

2.2 Reprezentacija particije

Da bismo particiju $\Gamma_{\mathbb{V}}^{K}$ prikazali u memoriji računala, koristit ćemo matricu particije reda $N \times K$. Neka je $X = [X_1, ..., X_K]$, gdje X_l označava binarni indikator za $\mathbb{V}_l$, odnosno matematički

$$X(i, l) = \langle i \in \mathbb{V}_l \rangle, \qquad i \in \mathbb{V}, l \in [K],$$

gdje je $\langle \cdot \rangle = 1$ ako je argument unutar zagrada istinit, a 0 inače. Kako jedan čvor može bit dodijeljen jednoj i samo jednoj particiji, stupci matrice X su međusobno isključivi, tj, vrijedi $X1_K = 1_N$, gdje je 1_d vektor-stupac dimenzije $d \times 1$ čije su sve komponente jednake 1. Definiramo matricu stupnjeva za simetričnu matricu težina W kao:

$$D = \text{Diag}(W1_N)$$

gdje Diag$(\cdot)$ označava dijagonalnu matricu generiranu svojim vektorskim argumentom. Ovo nam omogućava da funkcije *links* i *degree* zapišemo na drugačiji način

$$\begin{aligned} \text{links}(\mathbb{V}_l, \mathbb{V}_l) &= X_l^T W X_l \\ \text{degree}(\mathbb{V}_l) &= X_l^T D X_l. \end{aligned}$$

Problem maksimiziranja K-particioniranih normaliziranih rezova izražavamo kao optimizacijski problem s varijablom X, kojeg zovemo PNCX:

$$\text{maksimiziraj} \qquad \varepsilon(X) = \frac{1}{K} \sum_{l=1}^{K} \frac{X_l^T W X_l}{X_l^T D X_l} \tag{PNCX}$$

uz zahtjeve $X \in \{0, 1\}^{N \times K}$, te $X1_K = 1_N$.

3 Rješavanje problema K-normaliziranih rezova

Kao što je već spomenuto u Uvodu (1), problem PNCX se rješava u dva koraka.

1. korak: Aproksimacija diskretnog rješenja kontinuiranim rješenjem koristeći svojstvenu dekompoziciju.
2. korak: Pronalazak diskretnog optimalnog rješenja tako da izaberemo diskretnu matricu particije koje je najbliža kontinuiranom optimumu.

3.1 Pronalazak aproksimiranog kontinuiranog rješenja

Ako sa Z označimo matricu čiji su retci umnožak redaka od X i inverznog korijena dijagonalne matrice X^TDX, tj.

$$Z = X(X^TDX)^{-\frac{1}{2}} \tag{1}$$

i nazovemo Z *skalarnom partitivnom matricom*, tada se funkcija cilja ε može zapisati kao

$$\varepsilon(X) = \frac{1}{K}\mathrm{tr}(Z^TWZ). \tag{2}$$

Tada vrijedi

$$Z^TDZ = (X^TDX)^{-\frac{1}{2}}X^TDX(X^TDX)^{-\frac{1}{2}} = I_K,$$

gdje je I_K jedinična $K \times K$ matrica. Problem PNCX se tada može zapisati na problem

$$\text{maksimizirati} \quad \varepsilon(Z) = \frac{1}{K}tr(Z^TWZ) \tag{PNCZ}$$

$$\text{gdje je} \quad Z^TDZ = I_K \tag{PNCZ}$$

kojeg ćemo nazvati PNCZ. Promotrimo nekoliko sljedećih tvrdnji za tako definiranu skalarnu matricu Z i problem PNCZ.

Propozicija 1. (Ortogonalna nepromjenjivost) *Neka je $R \in \mathbb{R}^{K\times K}$ matrica. Ako je Z rješenje problema PNCZ, tada je i $\{ZR : R^TR = I_K\}$ rješenje. Također, rijedi $\varepsilon(ZR) = \varepsilon(Z)$*

Dokaz. Tvrdnja se dobiva korištenjem svojstva matrica $\mathrm{tr}(AB) = \mathrm{tr}(BA)$. □

Propozicija 2. (Ortogonalna svojstvena vrijednost) *Označimo sa (V, S) uređeni par svojstvene matrice V (stupci su joj svojstveni vektori) i matrice svojstvenih vrijednosti matrice s normaliziranim težinama $P = D^{-1}W$ tako da vrijedi $PV = VS$, $V = [V_1, ..., V_N]$, $S = Diag(s)$ gdje su svojstvene vrijednosti sortirane silazno $s_1 \geqslant \cdots \geqslant s_N$.*

(V, S) se dobiva iz ortonormiranog rješenja svojstvene zadaće $(\overline{V}, S)$ simetrične matrice $D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ tako da je

$$V = D^{-\frac{1}{2}} \overline{V} \tag{3}$$

$$D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \overline{V} = \overline{V} S \tag{4}$$

$$\overline{V}^T \overline{V} = I_N. \tag{5}$$

Sve komponente matrica V i S su realne. Bilo kojih K različitih svojstvenih vektora čine jedno moguće rješenje problema PNCZ te vrijedi

$$\varepsilon([V_{\pi_1}, \ldots, V_{\pi_K}]) = \frac{1}{K} \sum_{l=1}^{K} s_{\pi_l}, \tag{6}$$

gdje π_i predstavlja uređenu K-torku različitih indeksa iz $[N]$. Globalni optimum se dostiže postavljanjem $\pi = [1, \ldots, K]$, odnosno:

$$Z^* = [V_1, ...V_K], \tag{7}$$

$$\Lambda^* = Diag([s_1, \ldots, s_K]), \tag{8}$$

$$\varepsilon(Z^*) = \frac{1}{K} tr(\Lambda^*) = \max_{Z^T D Z = I_K} \varepsilon(Z). \tag{9}$$

Dokaz. Tvrdnja propozicije direktno slijedi iz *Rayleigh-Ritz teorema* i njegovih korolara i činjenice da je P stohastička matrica pa je 1_N njen trivijalni svojstveni vektor te je upravo 1 najveća svojstvena vrijednost te matrice. □

Koristeći ovu propoziciju zaključujemo da globalni optimum problema PNCZ nije jedinstven, već je skup rješenja potprostor

$$\{Z^* R : R^T R = I_K, P Z^* = Z^* \Lambda^*\} \tag{10}$$

ortogonalnih matrica koji sadrži prvih K najboljih svojstvenih vektora od P. Osim ako su sve svojstvene vrijednosti jednake, tada sva rješenja imaju optimalnu vrijednost.

Korolar 3. (Monotonost gornje međe funkcije cilja) *Za svaki K vrijedi*

$$\max \varepsilon(\Gamma_{\mathbb{V}}^K) \leqslant \max_{Z^T D Z = I_K} \varepsilon(Z) = \frac{1}{K} \sum_{l=1}^{K} s_l \tag{11}$$

$$\max_{Z^T D Z = I_{K+1}} \varepsilon(Z) \leqslant \max_{Z^T D Z = I_K} \varepsilon(Z) \tag{12}$$

■

Sljedeći korak nam je vratiti Z u diskretni prostor matrica particija. Ako je preslikavanje f preslikavalo X u Z, tada postoji f^{-1} koje će vratiti Z u X:

$$Z = f(X) = X(X^T D X)^{-\frac{1}{2}} \tag{13}$$

$$X = f^{-1}(Z) = \mathrm{Diag}(\mathrm{diag}^{-\frac{1}{2}}(Z Z^T)) Z, \tag{14}$$

gdje smo sa *diag* označili operator koji vraća dijagonalu matrice u obliku vektora retka.

Preslikavanje f^{-1} zapravo normalizira duljinu K-dimenzionalnih točaka (kordinate su predstavljene sa retcima matrice Z), tako da sve te točke leže na jediničnoj hipersferi sa središtem u središtu koordinatnog sustava. Sada jednakost

$$f^{-1}(Z^*R) = f^{-1}(Z^*)R \text{ jer je } R^TR = I_K \tag{15}$$

omogućava da zapišemo prostor optimalnih rješenja 10 u diskretnom X-prostoru:

$$\{\widetilde{X}^*R : R^TR = I_K, \widetilde{X}^* = f^{-1}(Z^*)R\}. \tag{16}$$

Kako baza ovog prostora mora biti ortogonalna, jasno je da trebamo samo K svojstvenih vektora (ne 2^K) da bi došli do K particija. Također, uočimo da iako se prvi svojstveni vektor čini trivijalan ($Z_1^* = (1_N^T D 1_N)^{-\frac{1}{2}} 1_N$ trivijalni umnožak 1_N, no $\widetilde{X}_1^*$ nije za $K > 1$) , on je bitan kao i svaki drugi jer im omogućava bazu za generiranje cijelog skupa rješenja.

3.2 Pronalazak diskretnog optimalnog rješenja

Optimum rješenja PNCZ uglavnom nije zadovoljavajuće rješenje problema PNCX. No, moguće je koristiti rješenja od PNCZ kako bismo pronašli približna diskretna rješenja. Diskretno rješenje koje ćemo naći neće biti globalni optimum, već približni globalni optimum. Dakle, zadatak nam je pronaći diskretno rješenje koje zadovoljava PNCX i koje je najbliže kontinuiranom optimumu 16. U tome će nam pomoći sljedeći teorem.

Teorem 4. (Optimalna diskretizacija) *Neka je $\widetilde{X}^* = f^{-1}(Z^*)$. Optimalna diskretna particija $\widetilde{X}^*$ zadovoljava sljedeći program (nazovimo ga POD)*

$$\text{minimiziraj} \qquad \phi(X, R) = ||X - \widetilde{X}^*R||^2 \tag{POD}$$

$$\text{gdje su } X \in \{0,1\}^{NxK}, X1_K = 1_N, R^TR = I_K \tag{POD}$$

gdje je $||M||$ Frobeniusova norma matrice M ($||M|| = \sqrt{tr(MM^T)}$).
Lokalni optimum $(\widetilde{X}^*, R)$ od POD može se riješiti iterativno. Sa zadanim R^*, POD, reduciramo do

$$\text{minimiziraj} \qquad \phi(X) = ||X - \widetilde{X}^*R^*||^2 \tag{17}$$

$$\text{gdje su } X \in \{0,1\}^{NxK}, X1_K = 1_N. \tag{18}$$

Postavimo $\widetilde{X} = \widetilde{X}^*R^*$. Tada je optimalno rješenje:

$$X^*(i,l) = \langle l = \arg\max_{k \in [K]} \widetilde{X}(i,k) \rangle, \quad i \in \mathbb{V}. \tag{19}$$

Sa zadanim X^*, reduciramo:

$$\text{minimiziraj} \qquad \phi(R) = ||X^* - \widetilde{X}^* R||^2 \tag{20}$$

$$\text{gdje je } R^T R = I_K. \tag{21}$$

Rješenje je dano uz pomoć singularnih vektora:

$$R^* = \widetilde{U} U^T, \tag{22}$$

$$X^{*T} \widetilde{X}^* = U \Omega \widetilde{U}^T, \Omega = Diag(\omega), \tag{23}$$

gdje je $(U, \Omega, \widetilde{U})$ *dekompozicija matrice na singularne vrijednosti od* $X^{*T}\widetilde{X}^*$*, uz* $U^T U = I_K,\ \widetilde{U}^T \widetilde{U} = I_K$ *i* $\omega_1 \geqslant \cdots \geqslant \omega_K$. ■

Dokaz. Uočimo da vrijedi:

$$\phi(X, R) = ||X||^2 + ||\widetilde{X}^*||^2 - tr(X R^T \widetilde{X}^{*T} + X^T \widetilde{X}^* R) = 2N - 2tr(X R^T \widetilde{X}^{*T}).$$

Tako da je minimizacija $\phi(X, R)$ ekvivalentna maksimiziranju $tr(X R^T \widetilde{X}^{*T})$. Kako za $R = R^*$ svaki element od $tr(X R^{*T} \widetilde{X}^{*T})$ može biti optimiziran zasebno, slijedi jednakost 19. Za zadani $X = X^*$ i problem PNCZ, konstruiramo :

$$L(R, \Lambda) = tr(X^* R^T \widetilde{X}^{*T}) - \frac{1}{2} tr(\Lambda^T (R^T R - I_K)).$$

Optimum (R^*, Λ^*) mora zadovoljavati:

$$L_R = \widetilde{X}^{*T} X^* - R\Lambda = 0 \ tj.\ \Lambda^* = R^{*T} \widetilde{X}^{*T} X^*. \tag{24}$$

Slijedi $\Lambda^{*T} \Lambda^* = U \Omega^2 U^T$. A kako je $\Lambda = \Lambda^T, \Lambda^* = U \Omega U^T$, iz 24 slijedi $R^* = \widetilde{U} U^T$ i $\phi(R^*) = 2N - 2tr(\Omega)$. A upravo veća vrijednost od $tr(\Omega)$ znači da je X^* bliži $\widetilde{X}^* R^*$. □

Zbog ortonormirane invarijantnosti prostora kontinuiranog optimuma, naša metoda je dobra za svaku početnu iteraciju od X ili R. Naravno, bolja početna iteracija može ubrzati konvergenciju ka rješenju.

Za dani X^*, rješavamo PNCZ da bismo pronalašli prostor kontinuiranih optimuma $\widetilde{X}^* R^*$. Nakon toga, riješavamo PNCX kako bismo pronašli najbliže diskretno rješenje. U svakom koraku smanjujemo ϕ. Algoritam će pronaći **lokalni** optimum koje će varirati s obzirom na početnu iteraciju. Kako su $\widetilde{X}^* R^*$ globalni optimumi bez obzira kakav je R^*, bez obzira u koji $\widetilde{X}^* R^*$ algoritam konvergira, njegovo približno diskretno rješenje X^* neće se mnogo razlikovati od globalnog optimuma.

4 Algoritam

Pretpostavljamo da su nam zadani matrica W i broj klastera K.

1. Izračunaj $D = \text{Diag}(W1_N)$
2. Nađi Z^*

$$\begin{aligned} D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \overline{V}_{[K]} &= \overline{V}_{[K]} \text{Diag}(s_{[K]}), \\ \overline{V}_{[K]}^T \overline{V}_{[K]} &= I_K \\ Z^* &= D^{-\frac{1}{2}} \overline{V}_{[K]}. \end{aligned}$$

3. Normaliziraj Z^*

$$\widetilde{X}^* = \text{Diag}\left(\text{diag}^{-\frac{1}{2}}(Z^* Z^{*T})\right) Z^*.$$

4. Inicijaliziraj X^* računajući R^*

$$\begin{aligned} &R_1^* = [\widetilde{X}^*(i,1), \ldots, \widetilde{X}^*(i,K)]^T, \text{ za slučajno odabrani } i \in [N] \\ &c = 0_{Nx1} \\ &\textbf{Za } k = 2, \ldots, K \textbf{ ponavljaj}: \\ &\quad c = c + \text{abs}(\widetilde{X}^* R_{k-1}^*) \\ &\quad R_k^* = [\widetilde{X}^*(i,1), \ldots, \widetilde{X}^*(i,K)]^T, \quad i = \arg\min c \end{aligned}$$

5. Inicijaliziraj $\overline{\phi}^* = 0$.
6. Pronađi optimalno diskretno rješenje X^*

$$\begin{aligned} \widetilde{X} &= \widetilde{X}^* R^* \\ X^*(i,l) &= \langle l = \arg\max_{k \in [K]} \widetilde{X}(i,k) \rangle, \quad i \in \mathbb{V}, l \in [K] \end{aligned}$$

7. Pronađi optimalnu ortonormiranu matricu R^*:

$$\begin{aligned} &X^{*T} \widetilde{X}^* = U \Omega \widetilde{U}^T, \quad \Omega \text{ dijagonalna}; U, \widetilde{U} \text{ ortogonalne} \\ &\overline{\phi} = \text{tr}(\Omega) \\ &\textbf{Ako } |\overline{\phi} - \overline{\phi}^*| < \text{(proizvoljna točnost) onda } \textbf{STANI i vrati } X^* \\ &\overline{\phi}^* = \overline{\phi} \\ &R^* = \widetilde{U} U^T \end{aligned}$$

8. Idi na 6. korak.

Najviše vremena u gore navedenom algoritmu zauzima drugi korak kada računamo prvih K svojstvenih vektora $N \times N$ matrice. U četvrtome koraku imamo $NK(K-1)$ množenja prilikom računanja centroida. Šesti korak sastoji se od NK^2 množenja radi računanja $\widetilde{X}^* R^*$. U sedmome koraku je obuhvaćeno dekomponiranje $K \times K$ matrice na singularne vrijednosti te K^3 množenja radi računanja R^*. Uočimo kako je $X^*(i,j) \in \{0,1\}$ za svaki $i \in \mathbb{V}, j \in [K]$, pa se umnožak $X^{*T}\widetilde{X}^*$ može obaviti vrlo efikasno. Složenost ovog algoritma je

$$\mathrm{O}\left(\mathrm{O}_1 + NK^2\right)$$

gdje O_1 predstavlja složenost algoritma stvaranja matrice W.

Algoritam smo testirali u programskom jeziku Octave.

5 Testiranje algoritma

Prethodno opisani algoritam valja testirati na konkretnim podacima te, ukoliko je to moguće, provjeriti točnost (efikasnost) algoritma. Za provjeru ćemo koristiti četiri različita skupa podataka. U svakom od četiri skupa podataka nalaze se informacije (mjerenja) o statističkim jedinicama koje su sudjelovale u eksperimentima, tj. opservacijskim studijama. U svakom skupu podataka svaka statistička jedinica pripada jednoj i samo jednoj „grupi" (*klasteru*) te na osnovu preostalih (numeričkih) mjerenja želimo podijeliti čitav skup podataka u $K \in \mathbb{N}$ *klastera*.

Na temelju numeričkih mjerenja želimo svaku od $N \in \mathbb{N}$ statističkih jedinica alocirati nekom od K *klastera*. U početku pretpostavljamo da je broj *klastera* K zadan, a na posljednjem primjeru pokazujemo način na koji možemo odrediti „optimalni" broj *klastera* K.

Točnost algoritma računali smo na sljedeći način: Neka statistička jedinica $i \in \{1, 2, \ldots, N\}$ pripada *klasteru* $k \in \{1, 2, \ldots, K\}$ i pretpostavimo prvo da za svaku jedinicu znamo točno kojem *klasteru* pripada. Neka je algoritam odlučio da tu i-tu jedinicu treba alocirati u l-ti *klaster*, gdje je $l \in \{1, 2, \ldots, K\}$. Definiramo

$$\delta_i(k,\, l) = \begin{cases} 1, & k = l \\ 0, & \text{inače} \end{cases},$$

za $i \in \{1, 2, \ldots, N\}$. Točnost algoritma tada definiramo kao

$$ACC = \frac{1}{N} \cdot \sum_{i=1}^{N} \delta_i(k,\, l)\,.$$

Primijetimo da je ovako definirana točnost ustvari proporcija statističkih jedinica koje smo točno alocirali (to jest, stavili ih u *pravi klaster*).

5.1 Olive dataset

Podaci za ovaj primjer preuzeti su iz [2].

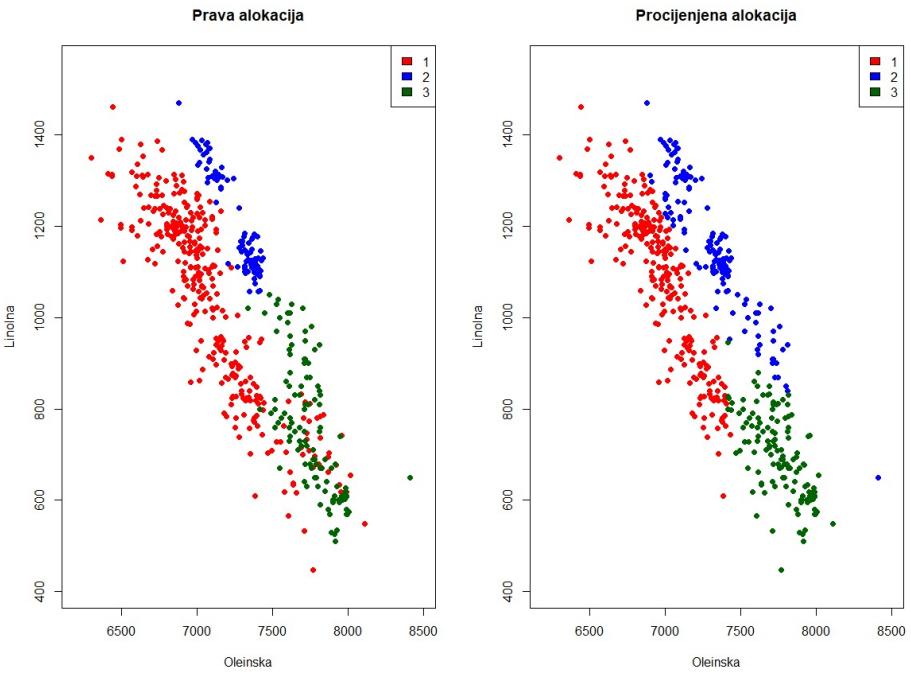
Maslinova ulja proizvedena u različitim regijama Italije imaju različit okus. Mogu li se ulja iz različitih talijanskih regija prepoznati samo po udjelima masnih kiselina? Podaci preuzeti iz navedene literature sadrže postotke 8 različitih masnih kiselina (palmitinska, palmitoleinska, stearinska, oleinska, linolna, linolenska, arahidska i eikosenska) mjerenih na uzorku 572 talijanska maslinova ulja. Za naš primjer koristimo podatke o postocima oleinske i linolne kiseline.

Znamo da ulja dolaze iz tri različite regije: južna Italija, Sardinija i sjeverna Italija, te ih želimo *klasterirati* na temelju postotaka ovih dviju uljnih kiselina. Ovdje znamo da je broj *klastera* jednak $K = 3$. ovom slučaju dobili smo da je točnost algoritma

$$ACC = 0.8182\,,$$

odnosno 81.82%, što je iznenađujuće dobar rezultat. Zaključujemo da s velikom sigurnošću možemo prepoznati iz koje talijanske regije dolazi (slučajno odabrano) maslinovo ulje tek na temelju postotka oleinske i linolne kiseline koje ono sadrži.

Grafički prikaz *klasteriranja* možemo vidjeti na Slici 1. Primjećujemo da razlika nema mnogo, a najočitije su u donjem desnom kutu slike - vidimo da su tamo podaci iz prve i treće regije na neki način „izmiješani" te algoritam nije mogao prepoznati razlike između ta dva *klastera*.



Slika 1: Grafički prikaz *klasteriranja* (Olive)

5.2 Wineratings dataset

Podaci za ovaj primjer preuzeti su iz [1].

Prikupljeno je 38 uzoraka vina sorte *Pinot Noir* napravljena u tri različite francuske pokrajine. Profesionalni kušači vina davali su ocjene vinu na temelju arome, okusa i kvalitete. Razlikuju li se vina između te tri pokrajine, to jest, možemo li samo na temelju dobivenih ocjena podijeliti dani uzorak u 3 pokrajine? U ovom slučaju je broj *klastera* ponovno jednak $K = 3$, budući da znamo da vina dolaze iz 3 regije.

Algoritam daje točnost

$$ACC = 0.4737\,,$$

odnosno 47.37%, što i nije naročito velik postotak. Nažalost, budući da su podaci višedimenzionalni (trodimenzionalni) ne možemo ih (efikasno) grafički prikazati kako bismo uočili netočno alocirane jedinice. Statističkom analizom podataka (konkretno, MANOVA-testom koji koristi Pillaijev trag kao testnu statistiku) možemo zaključiti da nema značajne razlike (p-vrijednost $= 0.947$) između ocjena vina proizvedenih u regijama 1 i 2 (pogotovo u aromi i okusu). Iako regija 3 ima značajno (praktično i statistički) veće ocjene od preostale dvije regije (p-vrijednost $= 0.0006$), algoritam nije uspješno prepoznao razlike između

te tri regije, to jest, nije s velikom točnošću podijelio dane jedinice u 3 kategorije, upravo zbog izuzetne sličnosti ocjena vina proizvedenih u regijama 1 i 2.

5.3 Student dataset

Je li prosjek ocjena studenta na neki način povezan s mjestom na kojem taj student voli sjediti u učionici? Odgovor na slično pitanje pokušala je dati profesorica Jessica Utts sa Sveučilišta u Kaliforniji te je u tu svrhu anketirala svoje studente u razdoblju od 2004. do 2005. godine. Između ostalog, anketa je ispitivala studentov prosjek ocjena (GPA - Grade Point Average), broj dana u mjesecu u kojima student izlazi na zabave, broj sati koje provede tjedno učeći te mjesto na kojem preferira sjediti u učionici (naprijed, u sredini ili odostraga).

Podaci za ovaj primjer preuzeti su iz [1].

Broj *klastera* jednak je $K = 3$ te algoritam daje točnost

$$ACC = 0.5783\,,$$

to jest 57.83%. Uzrok ovoj baš i ne previsokoj točnosti najvjerojatnije je činjenica da su broj izlazaka i broj sati učenja diskretne varijable (poprimaju vrijednosti na skupu $\{0, 1, 2, \ldots, G\}$, gdje je G neka gornja ograda, npr. 31 u slučaju prve varijable). To uzrokuje laganu *grupiranost* podataka već na temelju te dvije varijable i zbog toga nije nikako lako razlučiti između pretpostavljene tri kategorije (preferencije mjesta sjedenja).

5.4 Chromatin dataset

Podaci za ovaj primjer preuzeti su iz [2].

Ljudski kromosom ogromna je molekula, duljine čak 2 do 3 centimetra, a čine ga oko 100 milijuna baznih parova. Nalazi se u staničnoj jezgri koja je u promjeru duljine tek 0.01 milimetar. Kod diobe stanice, točnije u G1 fazi interfaze, oni nisu vidljivi jer su tada raspršeni u staničnoj jezgri. U toj podfazi dolazi do transkripcije i udvostručenja DNK molekula te se samim time i kromosom udvostručuje (do procesa iduće mitoze).

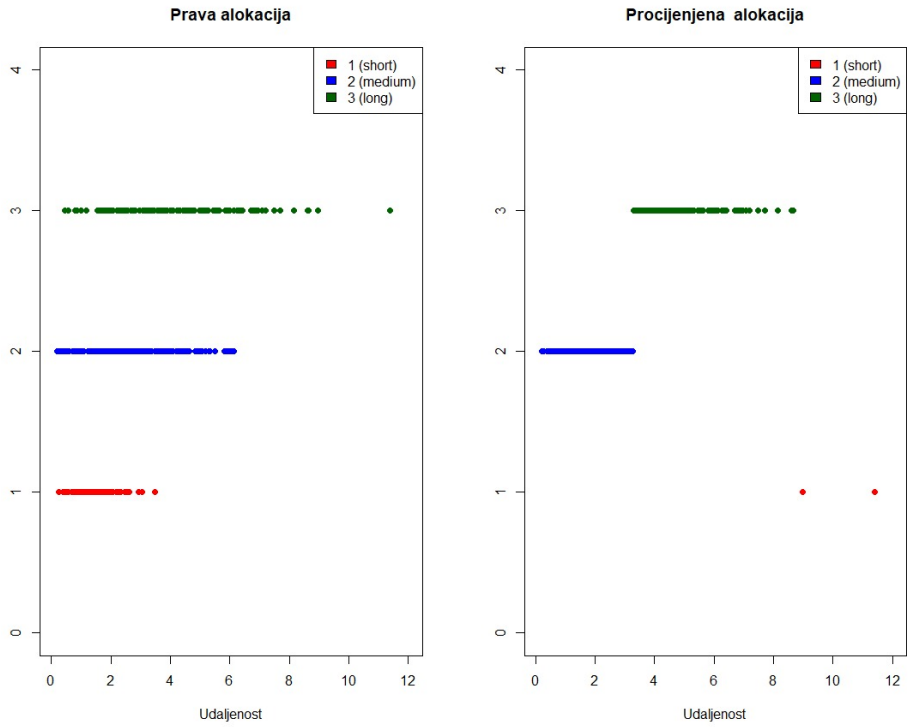
Izuzetna složenost ovog, a i ostalih procesa koji se zbivaju u stanici pokreću mnoga pitanja u vezi same prostorne organizacije kromosoma - laički rečeno, nije jasno *kako kromosom stane u jezgru*.
Provedena je nekolicina istraživanja ne bi li se saznalo više o toj organizaciji kromosoma, a predloženi su i raznovrsni modeli. Mi ćemo koristiti podatke jednog od tih istraživanja. U ovom istraživanju, parovi DNA sekvenci na specifičnim mjestima u ljudskom kromosomu označeni su fluorescentnim bojama i mjerene su udaljenosti između tih baznih parova (nukleobaza povezanih hidrogenskim vezama). Empirijska distribucija ovakvih mjerenja daje ideju o općenitoj organizaciji (za podatke ovog tipa, koji su dvodimenzionalna udaljenost, najčešće se prilagođuje Rayleigheva distribucija). Udaljenosti su kategorizirane prema duljini baznih parova (kratki, srednji i dugi), dakle, broj *klastera* je $K = 3$.

Točnost algoritma u ovome slučaju iznosi

$$ACC = 0.5660\,,$$

odnosno 56.60%.

Na slici ispod vidimo grafički prikaz *klasteriranja* - na lijevoj slici prikazana su mjerenja u sve tri (prave) grupe dok su na slici desno prikazane dobivene procjene. Ova dosta slaba točnost može se pripisati nejednakoj varijabilnosti među grupama. Neparametarski Levene-test jednakosti varijanci odbacuje hipotezu o jednakom raspršenju podataka između tri dane grupe (p-vrijednost $= 5.7 \cdot 10^{-8}$). Iako postoji značajna razlika između aritmetičkih sredina (to jest, lokacija) ove tri grupe (Kruskal-Wallis test, p-vrijednost $= 4.6 \cdot 10^{-33}$), zbog nejednakog raspršenja podataka nije moguće efikasno grupirati podatke (jednostavno rečeno, dobro su *izmiješani*).



Slika 2: Grafički prikaz *klasteriranja* (Chromatin)

5.4.1 Traženje *optimalnog* broja *klastera*

Kako smo već naveli na početku ovog odjeljka, implementirat ćemo proceduru za traženje „najboljeg" broja *klastera* K. Ideja je da pokrenemo naš algoritam za različite vrijednosti K (gdje je K element nekog *razumnog* skupa, na primjer

$\{1, 2, \ldots, 10\}$) te za svaki taj K nađemo postotak objašnjene varijabilnosti. Za „najbolji" K tada uzmemo onaj najmanji gdje promjena (porast) u postotku objašnjene varijabilnosti nije značajna.

S povećanjem broja *klastera* K raste i postotak objašnjene varijabilnosti (štoviše, taj postotak postaje 100% za $K = N$, to jest kad je svaka točka svoj *klaster*). Međutim, želimo izbjeći korištenje prevelikog broja *klastera*. Zbog toga trebamo odrediti neki *cutoff*, tj. najmanju takvu vrijednost K za koju vrijedi da uzimanjem $K + 1$ (ili nekog većeg broja) umjesto K ne postižemo značajno bolji rezultat. Ova metoda daje grafički kriterij za odabir takvog K.

Neka je K broj *klastera*, neka su n_i brojevi (duljine) podataka u i-tom *klasteru*, neka su $\overline{X}_i$ aritmetičke sredine u i-tom *klasteru* te neka su S_i^2 varijance u i-tom *klasteru*, za $i \in \{1, 2, \ldots, K\}$. Neka je još $\overline{X}$ aritmetička sredina svih (objedinjenih) podataka. Stavimo

$$SSG = \sum_{i=1}^{K} n_i(\overline{X}_i - \overline{X})^2 \quad \text{i} \quad SSE = \sum_{i=1}^{K}(n_i - 1)S_i^2 \,.$$

Primijetimo da je SSG varijabilnost koju smo uspjeli objasniti grupiranjem (SSG = sum of squares due to grouping), dok je SSE ona varijabilnost koja je ostala neobjašnjena zbog slučajne pogreške i prirodne varijabilnosti među podacima (SSE = sum of squares due to error). Ukupna varijabilnost je tada $SSG + SSE$, a proporcija objašnjene varijabilnosti je

$$POV = \frac{SSG}{SSG + SSE} \,.$$

Ova se metoda zove *metoda lakta* (*elbow-method*), a sa slike će biti jasno odakle dolazi takav naziv.

Da bi graf bio još ilustrativniji, umjesto proporcije objašnjene varijabilnosti računali smo proporciju neobjašnjene varijabilnosti koja, jasno, iznosi

$$PNV = 1 - POV = 1 - \frac{SSG}{SSG + SSE} = \frac{SSE}{SSG + SSE} \,.$$

Gornji izraz izračunali smo za svaki $K \in \{1, 2, \ldots, 10\}$ te dobivene vrijednosti prikazali na grafu u ovisnosti o K. Graf je prikazan na slici ispod.

Gledajući Sliku 3 jasno je zašto se ova metoda zove *metoda lakta* - njen izgled podsjeća na lagano podignutu ruku, a *optimalni* broj *klastera* K nalazi se upravo u *laktu*. Drugim riječima, gledajući graf slijeva nadesno, krivulja naglo opada do traženog K, a nakon njega nema značajne promjene (to jest, značajnog pada) u vrijednosti krivulje.

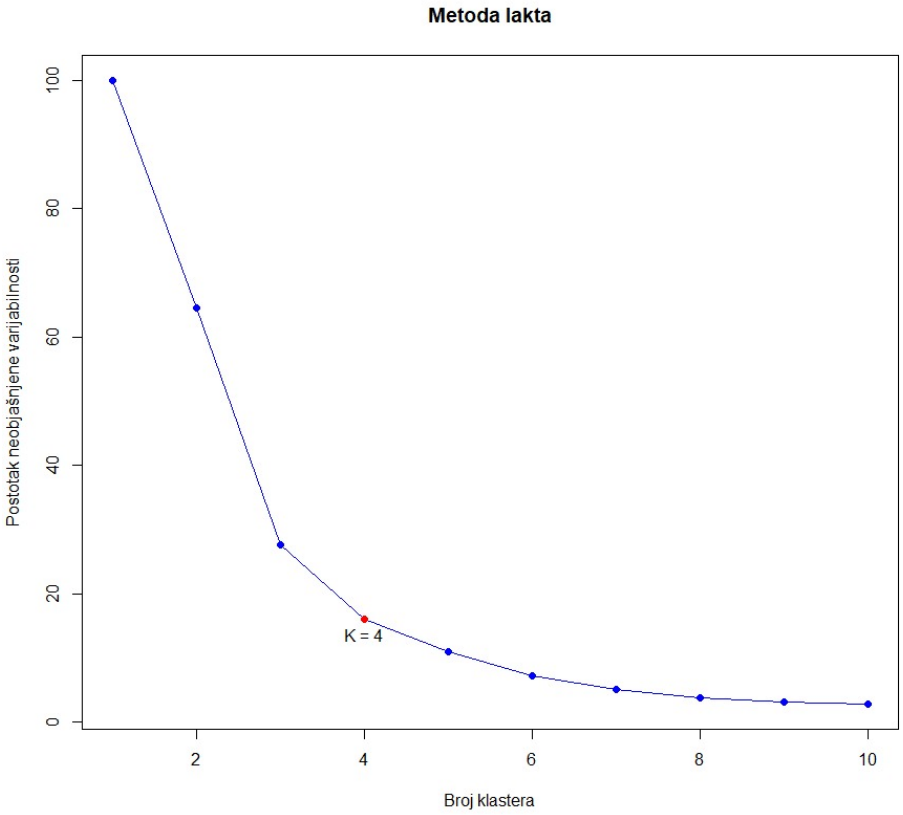
Za naš primjer dobili smo da je *optimalni* broj *klastera* $K = 4$, iako se i $K = 3$ ne čini kao loš izbor. Ovakva dvojba najčešće je rezultat korištenja ove metode - vrlo često nije očito gdje se nalazi *lakat*, tj. koji K odabrati kao *optimalni*.

Sofisticiranije metode uključivale bi statističku analizu ovog grafa. Preciznije, mogli bismo testirati sljedeće hipoteze:

$$H_0 : K \textit{ klastera} \text{ je dovoljno}$$

$$H_1 : K + 1 \textit{ klaster} \text{ je potreban}$$

U slučaju da je neka normalna distribucija razuman model za naše podatke može se izvesti egzaktna nul-distribucija ovakvog testa (Fisherova F-statistika), a u suprotnom bismo pribjegli nekoj od simulacijskih metoda, recimo, Monte Carlo metodi, za traženje aproksimativne distribucije ovakve testne statistike.



Slika 3: Optimalni K

6 Sažetak

Izložili smo iscrpan račun višeklasnog spektralnog grupiranja. S obzirom na diskretnu formulaciju grupiranja, prvo smo riješili pojednostavljeni neprekidni optimizacijski problem preko svojstvene dekompozicije. Također, pojasnili smo ulogu svojstvenih vektora kao generatora svih optimalnih rješenja kroz ortonormiranu transformaciju. Nakon toga smo krenuli na rješavanje diskretnog problema optimizacije kojim se traži diskretno rješenje najbliže neprekidnom rješenju. Iterativno, korištenjem dekompozicije singularnih vrijednosti (SVD) i ne-maksimalne supresije učinkovito postižemo diskretizaciju. Diskretna rješenja su gotovo globalno optimalna. Naša metoda je dobra za slučajne inicijalizacije i konvergira brže nego ostale metode grupiranja.

Literatura

[1] Robert F. Heckard Jessica M. Utts. *Mind on Statistics(5th ed.)* Cengage Learning, 2014.

[2] John A Rice. *Mathematical Statistics and Data Analysis(3rd ed.)* Belmont, CA : Thomson/Brooks/Cole, 2007.

[3] Jianbo Shi Stella X. Yu. „Multiclass Spectral Clustering". *Proceedings Ninth IEEE International Conference on Computer Vision* (2003).