SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO–MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

UVOD U SLOŽENO PRETRAŽIVANJE PODATAKA

Prof.dr.sc. Zlatko Drmač

Višeklasno spektralno klasteriranje

Mia Matijašević | Maja Piskač | Mia Tadić

U ovom seminaru proučavamo grupiranje podataka algoritmom *spektralno klasteriranje*.
Uz dobivenu **diskretnu** klastering formulaciju, najprije translatiramo taj diskretni problem na **neprekidni** problem te rješavamo neprekidni optimizacijski problem koristeći svojstvenu dekompoziciju. Razjašnjavamo ulogu svojstvenih vektora kao generatora svih optimalnih rješenja kroz ortonormirane transformacije.
Zatim rješavamo optimalni **diskretni** problem koji traži diskretno rješenje najbliže neprekidnom optimumu. Diskretizacija je efikasno izračunata iterativnom metodom koristeći SVD i ne-maksimalnu supresiju. Konačno diskretno rješenje je blizu globalno optimalnog.
Ova metoda je neovisna o inicijalizaciji i konvergira brže od ostalih klastering metoda.
Prikazat ćemo i testiranje segmentacije slike.

KLJUČNE RIJEČI
klasteriranje, svojstvena dekompozicija, diskretno, neprekidno, optimum, inicijalizacija, konvergencija

30. siječnja 2020.

Sadržaj

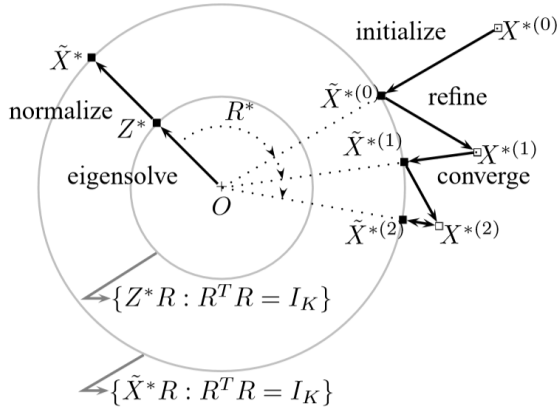
1 | UVOD

Spektralno particioniranje grafa se uspješno primjenjuje na probleme segmentacije slike, uravnoteženja opterećenja (engl. *load balancing*) te dizajniranja, konstruiranja i održavanja elektroničkog sustava (engl. *circuit layout*). Glavna značajka postupka je da se ne uključuju nikakve pretpostavke o globalnoj strukturi podataka, već se promatra **lokalno** pripadaju li dva podatka istoj klasi, a tek se na kraju donosi globalna odluka za podjelu svih podataka u disjunktne skupove po nekom određenom kriteriju. Uglavnom se pokušava koristiti kriterij koji se jednostavno skalira te očuva iste odnose i u manjim dimenzijama.

Ono što je privlačno kod spektralnog klasteriranja je što se globalni optimum u neprekidnoj domeni dobiva pomoću svojstvene dekompozicije. S druge strane, za dobivanje diskretnog rješenja pomoću svojstvenih vektora često treba rješavati drugi klastering problem, i to u manjedimenzionalnom prostoru. To jest, svojstveni vektori se tretiraju kao geometrijske koordinate skupa točaka. Brojne klastering heuristike, poput K-sredina algoritma (engl. *K-means*), se u nekom trenutku primijene i ovdje (na tom skupu točaka) kako bi se dobile particije.

Pokazujemo da postoji način da se povrati diskretni optimum. Temelji se na činjenici da se neprekidni optimum ne sastoji samo od svojstvenih vektora, već i od cijele familije vektora dobivenih od svojstvenih vektora prolazeći kroz ortonormirane transformacije. Cilj je naći onu pravu ortonormiranu transformaciju koja vodi diskretizaciji.

Na sljedećoj slici predstavljamo shemu algoritma, a ispod slike je i shematski opis algoritma.



SLIKA 1 Shematski dijagram algoritma.

Okvirni koraci algoritma:

1. Najprije izračunamo svojstvene vektore Z^*. U unutarnjem krugu je prikazano kako Z^* generira cijelu familiju globalnih optimuma kroz ortonormiranu transformaciju R. Nakon normiranja duljine vektora, svaki optimum odgovara jednom rješenju $\tilde{X}^{*(i)}$ na neprekidnoj domeni (vanjski krug).
2. Zatim iterativnom metodom pronalazimo diskretno rješenje najbliže neprekidnom optimumu. Počevši od diskretnog rješenja $X^{*(0)}$, nađemo $\tilde{X}^{*(0)}$ tako da izračunamo R^* koja daje najbliži $\tilde{X}^{*(0)}$ vektoru $X^{*(0)}$. Dobivši neprekidni optimum $\tilde{X}^{*(0)}$, izračunamo novo, njemu najbliže diskretno rješenje, i tako dalje ponavljamo postupak dok ne dođemo do konvergentnog para rješenja. Algoritam konvergira paru rješenja ($X^{*(2)}$, $\tilde{X}^{*(2)}$), koje je najbliže jedno drugome. Optimalnost od $\tilde{X}^{*(2)}$ osigurava da je $X^{*(2)}$ blizu globalnom optimalnom rješenju.

2 | VIŠEKLASNI NORMALIZIRANI REZOVI

Težinski graf reprezentiramo uređenom trojkom $\mathbb{G} = (\mathbb{V}, \mathbb{E}, W)$, gdje je $\mathbb{V}$ skup svih čvorova, $\mathbb{E}$ skup svih bridova koji povezuju čvorove, a W je matrica sličnosti s težinama koje karakteriziraju mogućnost da dva čvora pripadaju istoj grupi. W je po pretpostavci nenegativna i simetrična.

Objasnimo kako povezujemo težinski graf s našim algoritmom. Skup $\mathbb{V} = \{v_1, \ldots, v_N\}$ čine podaci koje želimo podijeliti u klase na temelju određenog kriterija sličnosti $W = (w)_{i,j=1,\ldots,N}$. Dva podatka v_i, v_j smatramo sličnima ako je $w_{ij} > 0$, a nisu slični ako je $w_{ij} = 0$. Iz toga slijedi nenegativnost matrice W. Skup $\mathbb{E}$ sastavljen je od bridova, a brid između vrhova v_i, v_j postoji ako je $w_{ij} > 0$. Napomenimo da je graf $\mathbb{G}$ neusmjeren — ako postoji brid između v_i i v_j, onda postoji i brid između v_j i v_i, i naravno s istom težinom, odnosno sličnošću. Iz toga slijedi simetričnost matrice W. Bridovi predstavljaju povezanost među našim podacima. Preformulacija našeg algoritma u kontekstu težinskog grafa je da želimo pronaći najbolju particiju grafa $\mathbb{G}$ u K klasa tako da su vrhovi u istoj klasi što povezaniji, tj. što sličniji, te da su vrhovi iz različitih klasa što manje slični. To jest, klasteriranje N točaka u K klasa je dekomponiranje skupa $\mathbb{V}$ u K disjunktnih skupova: $\mathbb{V} = \bigcup_{l=1}^{K} \mathbb{V}_l$, gdje je $\mathbb{V}_k \cap \mathbb{V}_l = \varnothing$, $\forall k \neq l$. Ovo K-particioniranje označavamo sa $\Gamma_{\mathbb{V}}^{K} = \{\mathbb{V}_1, \ldots, \mathbb{V}_K\}$.

2.1 | Kriteriji višeklasnog particioniranja

Neka su $\mathbb{A}, \mathbb{B} \subset \mathbb{V}$.

Definiramo *links*$(\mathbb{A}, \mathbb{B})$ kao sumu težina veza koje spajaju vrhove iz $\mathbb{A}$ s vrhovima u $\mathbb{B}$:

$$links(\mathbb{A}, \mathbb{B}) = \sum_{i \in \mathbb{A}, j \in \mathbb{B}} W(i,j). \tag{1}$$

Definiramo i funkciju *degree* koja za zadani skup zbraja sve težine veza koje izlaze iz vrhova tog skupa:

$$degree(\mathbb{A}) = links(\mathbb{A}, \mathbb{V}). \tag{2}$$

Sada definiramo sljedeći omjer koji predstavlja udio veza skupa $\mathbb{A}$ kojeg on ima sa skupom $\mathbb{B}$:

$$linkratio(\mathbb{A}, \mathbb{B}) = \frac{links(\mathbb{A}, \mathbb{B})}{degree(\mathbb{A})}. \tag{3}$$

Posebno su zanimljiva dva specijalna slučaja funkcije *linkratio*: $linkratio(\mathbb{A}, \mathbb{A})$ te $linkratio(\mathbb{A}, \mathbb{V} \setminus \mathbb{A})$. Prvi slučaj mjeri koliko veza iz $\mathbb{A}$ ostaju u $\mathbb{A}$, a drugi slučaj koliko veza iz $\mathbb{A}$ *pobjegne* iz $\mathbb{A}$. Ova dva slučaja su zanimljiva iz razloga što dobro grupiranje zahtijeva jaku povezanost unutar klasa i slabu povezanost između klasa. Ta dva kriterija matematiziramo formulacijama *K-klasne normalizirane asocijacije* i *K-klasni normalizirani rezovi*:

$$knassoc\left(\Gamma_{\mathbb{V}}^{K}\right) = \frac{1}{K} \sum_{l=1}^{K} linkratio(\mathbb{V}_l, \mathbb{V}_l). \tag{4}$$

$$kncuts\left(\Gamma_{\mathbb{V}}^{K}\right) = \frac{1}{K} \sum_{l=1}^{K} linkratio(\mathbb{V}_l, \mathbb{V} \setminus \mathbb{V}_l). \tag{5}$$

Kako vrijedi $knassoc\left(\Gamma_{\mathbb{V}}^{K}\right) + kncuts\left(\Gamma_{\mathbb{V}}^{K}\right) = 1$, maksimiziranje asocijacija i minimiziranje rezova se odvija simultano. Između mnogih drugih sličnih kriterija s rezovima, samo minimalni rezovi i normalizirani rezovi imaju ovo dualno svojstvo. No loša strana minimalnih rezova je da su osjetljivi na šumove (engl. *noise-sensitive*), to jest nekoliko izoliranih čvorova lako može odvući rezove od globalno optimalne particije, dok su normalizirani rezovi neovisni o težinskoj smetnji. Budući da su *knassoc* i *kncuts* ekvivalentni kriteriji, odsad ne radimo razliku između njih te kao **kriterij** za **K-klasne *normalizirane* rezove** promatramo

$$\epsilon\left(\Gamma_{\mathbb{V}}^{K}\right) = knassoc\left(\Gamma_{\mathbb{V}}^{K}\right), \tag{6}$$

gdje je ϵ vrijednost između 0 i 1, ovisno o K, i bez mjerne jedinice je.

Za svaki kriterij K-klasnog particioniranja, treba ispitati njegovu ovisnost o broju K. Formule (4) i (5) ne daju jasan odgovor na pitanje ovisnosti, ali pokazat ćemo da se gornja granica od ϵ smanjuje kako se povećava K.

2.2 | Reprezentacija particija

Koristimo *NxK matricu particije X* za reprezentaciju $\Gamma_{\mathbb{V}}^{K}$.

Neka je $X = [X_1, \ldots, X_K]$, gdje je X_l binarni indikator za $\mathbb{V}_l$:

$$X(i,l) = \langle i \in \mathbb{V}_l \rangle, \quad i \in \mathbb{V}, l \in [K], \tag{7}$$

gdje je $\langle\cdot\rangle$ jednako 1 ako je ono unutar šiljastih zagrada istinito, odnosno 0 ako je neistinito. Kako je svaki čvor raspoređen u samo jednu klasu, postoji *isključujući* uvjet između stupaca matrice X: $X1_K = 1_N$, gdje je 1_d vektor jedinica dimenzije $dx1$.

Definiramo *matricu stupnjeva* za simetričnu matricu težina W:

$$D = Diag(W1_N), \tag{8}$$

gdje $Diag(\vec{x})$ označava dijagonalnu matricu s vektorom $\vec{x}$ na dijagonali.

Sada možemo zapisati već poznate funkcije (1) i (2) na sljedeći način:

$$links(\mathbb{V}_l, \mathbb{V}_l) = X_l^T W X_l \tag{9}$$

$$degree(\mathbb{V}_l) = X_l^T D X_l. \tag{10}$$

Kriterij (6) K-klasnih normaliziranih rezova je izražen kao optimizacijski problem varijable X kojeg nazivamo *PNCX*:

$$\text{maksimiziraj} \quad \epsilon(X) = \frac{1}{K}\sum_{l=1}^{K}\frac{X_l^T W X_l}{X_l^T D X_l} \tag{11}$$

$$\text{za} \quad X \in \{0,1\}^{NxK} \tag{12}$$

$$X1_K = 1_N. \tag{13}$$

Zanimljivo je da je ovo NP-problem čak i za $K = 2$ te čak i kad je graf planaran. Prezentirat ćemo brz algoritam koji pronalazi rješenje blisko globalno optimalnom rješenju.

3 | RJEŠAVANJE K-KLASNIH NORMALIZIRANIH REZOVA

Rješavamo problem *PNCX* u dva koraka. Prvo (potpoglavlje 3.1 Traženje optimalnih neprekidnih rješenja) pretvorimo tranformiranu formulaciju u problem svojstvenih vrijednosti. Pokažemo da njegov globalni optimum nije jedinstven te konačno rješenje čine generalizirani svojstveni vektori para matrica (W, D). Transformiranjem svojstvenih vektora u prostor matrica particija, dobivamo skup neprekidnih globalnim optimuma. Zatim u drugom koraku (potpoglavlje 3.2 Traženje optimalnih diskretnih rješenja) riješimo diskretni problem, gdje je diskretna matrica particija uzeta tako da je najbliža neprekidnom optimumu. Takvo diskretno rješenje je blizu globalno optimalnom.

3.1 | Traženje optimalnih neprekidnih rješenja

Pojednostavljujemo jednadžbu (11):

$$\epsilon(X) = \frac{1}{K} tr\left(Z^T W Z\right), \tag{14}$$

gdje *tr* označava trag matrice, a Z je *skalirana matrica particije*:

$$Z = X\left(X^T D X\right)^{-\frac{1}{2}}. \tag{15}$$

Kako je $X^T D X$ dijagonalna, stupci od Z su stupci od X skalirani inverznim korijenom stupnjeva particija. Prirodni uvjet na Z je

$$Z^T D Z = \left(X^T D X\right)^{-\frac{1}{2}} X^T D X \left(X^T D X\right)^{-\frac{1}{2}} = I_K, \tag{16}$$

gdje I_K označava $K x K$ matricu identiteta.

Ignorirajući ograničenja u *PNCX*, izvodimo novi problem varijable Z i nazivamo ga *PNCZ*:

$$\text{maksimiziraj} \quad \epsilon(Z) = \frac{1}{K} tr\left(Z^T W Z\right) \tag{17}$$

$$\text{za} \quad Z^T D Z = I_K. \tag{18}$$

Prenošenjem Z na neprekidnu domenu pretvaramo diskretni problem u neprekidni optimizacijski problem. Problem ima posebno svojstvo koje iskazujemo sljedećom propozicijom, a koje se može dokazati trivijalno koristeći činjenicu $tr(AB) = tr(BA)$.

Propozicija 1 (Ortonormirana invarijantnost) *Neka je R matrica dimenzija $K x K$. Ako je Z rješenje problema* PNCZ, *onda je rješenje i skup $\{ZR : R^T R = I_K\}$ te vrijedi: $\epsilon(ZR) = \epsilon(Z)$.*

Dakle, proizvoljna rotacija čuva dobro rješenje poblema *PNCZ*.

Problem *PNCZ* se spominje u Rayleigh-Ritzovom teoremu. *Propozicija 2* koja slijedi, preoblikuje Rayleigh-Ritzov teorem prema našem problemu. Teorem se može dokazati direktno, koristeći Lagrangeovu relaksaciju. Propozicija

pokazuje da su među svim optimumima upravo svojstveni vektori od (W, D), odnosno oni *normalne matrice težina* P:

$$P = D^{-1}W. \tag{19}$$

Kako je P stohastička matrica, lako je provjeriti da je 1_N trivijalan svojstveni vektor matrice P i on odgovara najvećoj svojstvenoj vrijednosti 1.

Propozicija 2 (Optimalno svojstveno rješenje) *Neka je (V, S) svojstvena dekompozicija matrice P: $PV = VS$, gdje je $V = [V_1, \ldots, V_N]$, a $S = Diag(s)$ je dijagonalna matrica sa svojstvenim vrijednostima poredanima tako da vrijedi $s_1 \geq \ldots \geq s_N$. (V, S) je dobivena iz ortonormirane svojstvene dekompozicije $(\tilde{V}, S)$ simetrične matrice $D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$, gdje je*

$$V = D^{-\frac{1}{2}}\tilde{V}, \tag{20}$$

$$D^{-\frac{1}{2}}WD^{-\frac{1}{2}}\tilde{V} = \tilde{V}S, \quad \tilde{V}^T\tilde{V} = I_N. \tag{21}$$

Dakle, V i S su realni i bilokojih K različitih svojstvenih vektora čine kandidata za *PNCZ*, sa

$$\epsilon\left(\left[V_{\pi_1}, \ldots, V_{\pi_K}\right]\right) = \frac{1}{K}\sum_{l=1}^{K} s_{\pi_l}, \tag{22}$$

gdje je π *vektor indeksa* od K različitih brojeva iz skupa $[N] = \{1, \ldots, N\}$. Globalni optimum problema *PNCZ* se postiže za $\pi = [1, \ldots, K]$:

$$Z^* = [V_1, \ldots, V_K], \tag{23}$$

$$\Lambda^* = Diag\left([s_1, \ldots, s_K]\right), \tag{24}$$

$$\epsilon\left(Z^*\right) = \frac{1}{K}tr\left(\Lambda^*\right) = \max_{Z^TDZ=I_K} \epsilon\left(Z\right). \tag{25}$$

Da sumiramo, globalni optimum problema *PNCZ* nije jedinstven. To je potprostor generiran s prvih K najvećih svojstvenih vektora od P kroz ortonormirane matrice:

$$\{Z^*R : R^TR = I_K, \quad PZ^* = Z^*\Lambda^*\}. \tag{26}$$

Osim ako svojstveni vektori nisu svi isti, Z^*R nisu više svojstveni vektori od P. Sva ta rješenja imaju optimalnu vrijednost ϵ, koja pruža nepovećavajuću gornju granicu za *PNCZ*.

Korolar 1 (Monotonost gornje međe) *Za svaki K vrijedi*

$$\max \epsilon\left(\Gamma_{\mathbb{V}}^K\right) \leq \max_{Z^TDZ=I_K} \epsilon\left(Z\right) = \frac{1}{K}\sum_{l=1}^{K} s_l \tag{27}$$

$$\max_{Z^TDZ=I_{K+1}} \epsilon\left(Z\right) \leq \max_{Z^TDZ=I_K} \epsilon\left(Z\right) \tag{28}$$

Zatim transformiramo Z natrag u prostor matrica particija. Ako je f preslikavanje koje skalira X na Z, onda je f^{-1} normalizacija koja prevodi Z natrag u X:

$$Z = f(X) = X\left(X^T D X\right)^{-\frac{1}{2}} \tag{29}$$

$$X = f^{-1}(Z) = Diag\left(diag^{-\frac{1}{2}}\left(ZZ^T\right)\right)Z, \tag{30}$$

gdje funkcija *diag* vraća dijagonalu matrice u obliku vektora. Ako uzmemo redove od Z kao koordinate K-dimenzionalnih točaka, ono što f^{-1} radi je da normalizira njihove duljine tako da leže na jediničnoj hipersferi centriranoj u ishodištu. S f^{-1} transformiramo neprekidni optimum Z^*R iz Z-prostora u X-prostor:

$$\{\tilde{X}^*R : \tilde{X}^* = f^{-1}(Z^*), R^TR = I_K\}. \tag{31}$$

Sada je jasno da trebamo samo K svojstvenih vektora da bismo generirali K particija. Razlog tome je što indikatori grupe imaju ograničenje da su ortogonalni. Ne mogu biti odabrani bilokako. Također, jasniju perspektivu dobivamo iz prvog svojstvenog vektora. Iako je $Z_1^* = \left(1_N^T D 1_N\right)^{-\frac{1}{2}} \cdot 1_N$ trivijalan umnožak od 1_N, $\tilde{X}_1^*$ nije za $K > 1$. Naočigled trivijalan prvi svojstveni vektor je jednako važan kao i ostali, s obzirom da kolektivno čine bazu za generiranje cijelog skupa optimuma.

3.2 | Traženje optimalnih diskretnih rješenja

Optimumi problema *PNCZ* obično nisu pogodni za problem *PNCX*. Međutim, možemo ih iskoristiti kako bismo pronašli blisko diskretno rješenje. Ovo diskretno rješenje možda nije u potpunosti najbolje rješenje problema *PNCX*, ali je blizu globalnom optimumu. Dakle, naš cilj ovdje je naći diskretno rješenje koje zadovoljava binarni uvjet originalnog problema *PNCX*, ali da je blizu neprekidnom optimumu danom s (31).

Teorem 1 (Optimalna diskretizacija) *Neka je* $\tilde{X}^* = f^{-1}(Z^*)$. *Optimalna diskretna particija* X^* *se smatra rješenjem koje zadovoljava problem pod nazivom* POD:

$$\text{minimiziraj} \quad \phi(X, R) = \left\|X - \tilde{X}^*R^*\right\|^2 \tag{32}$$

$$\text{za} \quad X \in \{0,1\}^{NxK}, \quad X1_K = 1_N \tag{33}$$

$$R^TR = I_K, \tag{34}$$

gdje je $\|M\|$ Frobeniusova norma *matrice* M: $\|M\| = \sqrt{tr\left(MM^T\right)}$. *Lokalni optimum* (X^*, R^*) *ovog bilinearnog problema se može riješiti iterativno.*

Za danu R^*, POD *reduciramo na problem* PODX *po* X*:*

$$\text{minimiziraj} \quad \phi(X) = \left\|X - \tilde{X}^*R^*\right\|^2 \tag{35}$$

$$\text{za} \quad X \in \{0,1\}^{NxK}, \quad X1_K = 1_N. \tag{36}$$

Neka je $\tilde{X} = \tilde{X}^*R^*$. *Optimalno rješenje je dano ne-maksimalnom supresijom (ako postoji više maksimuma, samo jedan, ali*

bilokoji od njih, može biti izabran kao isključujući uvjet za matricu particije):

$$X^*(i,l) = \langle l = arg\max_{k\in[K]} \tilde{X}(i,k)\rangle, \quad i \in \mathbb{V}. \tag{37}$$

Za danu X^*, POD *je reduciran na problem* PODR *po* R*:*

$$\text{minimiziraj} \quad \phi(R) = \left\|X^* - \tilde{X}^* R^*\right\|^2 \tag{38}$$

$$\text{za} \quad R^T R = 1_K, \tag{39}$$

te je rješenje dano pomoću singularnih vektora:

$$R^* = \tilde{U}U^T, \tag{40}$$

$$X^{*T}\tilde{X}^* = U\Omega\tilde{U}^T, \quad \Omega = Diag(\omega), \tag{41}$$

gdje je $(U,\Omega,\tilde{U})$ singular value decomposition (SVD) *od* $X^{*T}\tilde{X}^*$*, sa* $U^TU = I_K$, $\tilde{U}^T\tilde{U} = I_K$ *te* $\omega_1 \geq \ldots \geq \omega_K$.

Dokaz Primijetimo da vrijedi $\phi(X,R) = \|X\|^2 + \left\|\tilde{X}^*\right\|^2 - tr\left(XR^T\tilde{X}^{*T} + X^T\tilde{X}^*R\right) = 2N - 2tr\left(XR^T\tilde{X}^{*T}\right)$. Dakle je minimiziranje $\phi(X,R)$ ekvivalentno maksimiziranju $tr\left(XR^T\tilde{X}^{*T}\right)$. Za *PODX*, uz dano $R = R^*$, kako se svaki ulaz funkcije $diag\left(XR^{*T}\tilde{X}^{*T}\right)$ može optimizirati neovisno, slijedi jednadžba (37). Za *PODR*, uz dano $X = X^*$, konstruiramo Langrangeovu funkciju koristeći simetričnu matricu Λ:

$$L(R,\Lambda) = tr\left(X^*R^T\tilde{X}^{*T}\right) - \frac{1}{2}tr\left(\Lambda^T\left(R^TR - I_K\right)\right). \tag{42}$$

Optimum (R^*,Λ^*) mora zadovoljavati

$$L_R = \tilde{X}^{*T}X^* - R\Lambda = 0, \quad \text{odnosno} \quad \Lambda^* = R^{*T}\tilde{X}^{*T}X^*. \tag{43}$$

Dakle $\Lambda^{*T}\Lambda^* = U\Omega^2U^T$. Zbog $\Lambda = \Lambda^T, \Lambda^* = U\Omega U^T$. Iz (42) imamo: $R^* = \tilde{U}U^T$ i $\phi(R^*) = 2N - 2tr(\Omega)$. Što je $tr(\Omega)$ veći, to je X^* bliži $\tilde{X}^*R^*$. ■

Zbog invarijantnosti na ortonormiranost neprekidnog optimuma, naša metoda je neovisna o početnoj inicijalizaciji X i R. Dobra inicijalizacija može ipak ubrzati konvergenciju. Jedna dobra i brza inicijalizacija je jednostavan $K - means$ klastering s K centara koji su približno ortogonalni. Računski, to je ekvivalentno inicijalizaciji R^* biranjem K redova iz X^* koji su ortogonalni jedan drugom koliko god je moguće. Za takav neortogonalni R^*, pronalazak X^* iz (37) je baš $K - means$ klasteriranje s centrima jedinične duljine.

Uz dani X^*, rješavamo *PODR* da bismo pronašli neprekidni optimum $\tilde{X}^*R^*$ najbliži njemu. Za ovaj neprekidni optimum, nadalje rješavamo *PODX* da bismo pronašli njegovo najbliže diskretno rješenje. Svaki korak reducira istu objektivnu funkciju ϕ. Možemo jedino tvrditi da te iteracije završavaju u lokalnom optimumu, koji može varirati ovisno o inicijalizaciji. Ipak, kako su $\tilde{X}^*R^*$ sve globalni optimumi neovisno o R^*, kojem god $\tilde{X}^*R^*$ *POD* konvergira, njegov diskretni optimalni X^* neće biti daleko od optimalnog rješenja.

3.3 | Algoritam

Uz danu matricu težina W i broj klasa K:

1. Izračunaj matricu stupnjeva $D = Diag(W1_N)$.
2. Nađi optimalno svojstveno rješenje Z^*:

$$D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \tilde{V}_{[K]} = \tilde{V}_{[K]} Diag\left(s_{[K]}\right), \quad \tilde{V}_{[K]}^T \tilde{V}_{[K]} = I_K, \quad Z^* = D^{-\frac{1}{2}} \tilde{V}_{[K]}$$

3. Normiraj Z^*: $Diag\left(diag^{-\frac{1}{2}}\left(Z^* Z^{*T}\right)\right) Z^*$.
4. Inicijaliziraj X^* računajući R^*:

$$R_1^* = \left[\tilde{X}^*(i,1), \ldots, \tilde{X}^*(i,K)\right]^T, \quad \text{za slučajan } i \in [N]$$
$$c = 0_{Nx1}$$
$$\text{Za } k = 2, \ldots, K:$$
$$c = c + abs\left(\tilde{X}^* R_{k-1}^*\right)$$
$$R_k^* = \left[\tilde{X}^*(i,1), \ldots, \tilde{X}^*(i,K)\right]^T, \quad \text{za } i = argminc$$

5. Inicijaliziraj parametar za mjerenje konvergencije: $\bar{\phi}^* = 0$
6. Pronađi optimalno diskretno rješenje X^*:

$$\tilde{X} = \tilde{X}^* R^*$$
$$X^*(i,I) = \langle I = argmax_{k \in [K]} \tilde{X}(i,k) \rangle, \quad i \in \mathbb{V}, I \in [K]$$

7. Pronađi optimalnu ortonormiranu matricu R^*:

$$X^{*T} \tilde{X}^* = U \Omega \tilde{U}^T, \quad \Omega = Diag(\omega)$$
$$\bar{\phi} = tr(\Omega)$$
$$\text{Ako } |\bar{\phi} - \bar{\phi}^*| < \text{preciznosti računala, onda stani i vrati } X^*$$
$$\bar{\phi}^* = \bar{\phi}$$
$$R^* = \tilde{U} U^T$$

8. Idi na korak 6.

U koraku 2, koristimo $\tilde{V}_{[K]}$ kao pokratu za $\left[\tilde{V}_1, \ldots, \tilde{V}_K\right]$. U koraku 4, $B = abs(A)$ označava matricu s apsolutnim vrijednostima elemenata od A. U koraku 3, kako $Diag\left(diag^{-\frac{1}{2}}\left(Z^* Z^{*T}\right)\right) Z^*$ skalira duljinu svakog reda na 1, možemo preskočiti skaliranje $\tilde{V}$ da bismo dobili V, to jest $Z^* = \left[\tilde{V}_1, \ldots, \tilde{V}_K\right]$ vodi istom X^*.

Korak 2 nalazi K vodećih svojstvenih vektora od NxN matrice. On troši najviše vremena. Korak 4 vrši $NK(K-1)$ množenja u biranju K centara. Korak 6 vrši NK^2 množenja u računanju $\tilde{X}^* R^*$. Korak 7 uključuje SVD KxK matrice i K^3 množenja za računanje R^*. Kako je X^* binarna, $X^{*T} \tilde{X}^*$ se može izvršiti efikasno sa svim zbrajanjima. Zbrajajući sve zajedno, vremenska složenost algoritma je $O\left(N^{\frac{3}{2}} K + NK^2\right)$.

4 | TESTIRANJE

Testiranje prezentiranog algoritma višeklasnog spektralnog klasteriranja (kratica MSC, iz engl. *multiclass spectral clustering*) provodimo nad nekoliko različitih primjera. Algoritam smo implementirali u *Jupyter bilježnici* u programskom jeziku *Python*.

Kao parametre algoritmu šaljemo četiri parametra: matricu težina W, broj klastera u koje želimo grupirati naše podatke k, granicu udaljenosti dviju točaka koje čine konvergencijski par tol te maksimalan broj iteracija u kojem ćemo pokušati dobiti optimalno rješenje max_it. Funkcija vraća 3 objekta: diskretno rješenje X_d, vektor koji sadrži udaljenosti elemenata para rješenja rez te vrijednost $flag$ koja je 0 ako je došlo do konvergencije, odnosno 1 ako nije. Sam algoritam implementiran je prateći navedeni pseudokod u potpoglavlju 3.3 Algoritam.

Primjeri nad kojima smo testirali implementirani algoritam su:

1. skup točaka podijeljen u dvije koncentrične kružnice
2. skup točaka podijeljen u dva polumjeseca
3. segmentacija slike
4. težinski graf

Navedene primjere prezentiramo u sljedeća četiri potpoglavlja.

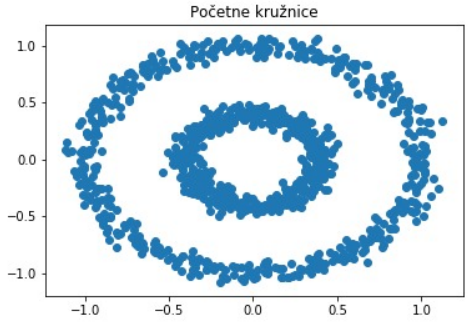
4.1 | Koncentrične kružnice

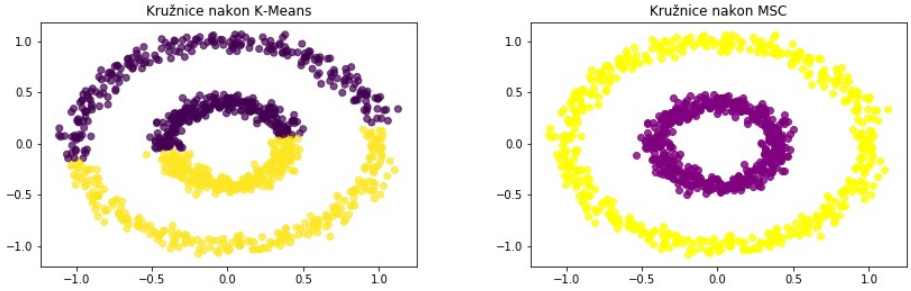
Ovo je poznati primjer korišten u klasteriranju podataka iz razloga što ga *K-Means* algoritam ne klasterira optimalno. Optimalan rezultat bi bio klasteriranje podataka u dvije kružnice, no *K-Means* podijeli podijeli skup tako da svaki klaster sadrži dvije gornje, odnosno donje, polukružnice.

Loš rezultat *K-Means*-a je naša motivacija za testiranje *MSC*-a. Test je proveden nad 1000 podataka. Podaci su generirani funkcijom *make_circles* iz *Python*-ovog paketa *sklearn.datasets*. Funkcija ih je automatski generirala tako da ih je rasporedila u dvije koncentrične funkcije. Kružnice su udaljene 0.4 mjerne jedinice.

Matrica težina *W* određena je pomoću funkcije *pairwise_distances* iz paketa *sklearn.metrics* koja vraća $n x n$ matricu M koja na mjestu (i, j) sadrži euklidsku udaljenost podatka i te podatka j. Zatim smo matricu M transformirali u matricu W koja sadrži samo nule i jedinice i to tako da na mjestu (i, j) stoji 1 ako je $M(i, j) < 0.3$, a 0 inače.

Test je proveden u 10 iteracija i s tolerancijom udaljenosti točaka kovergencijskog para 0.001. Naš *MSC* je dao dobre rezultate, dok *K-Means* nije, i to pokazujemo sljedećim slikama.

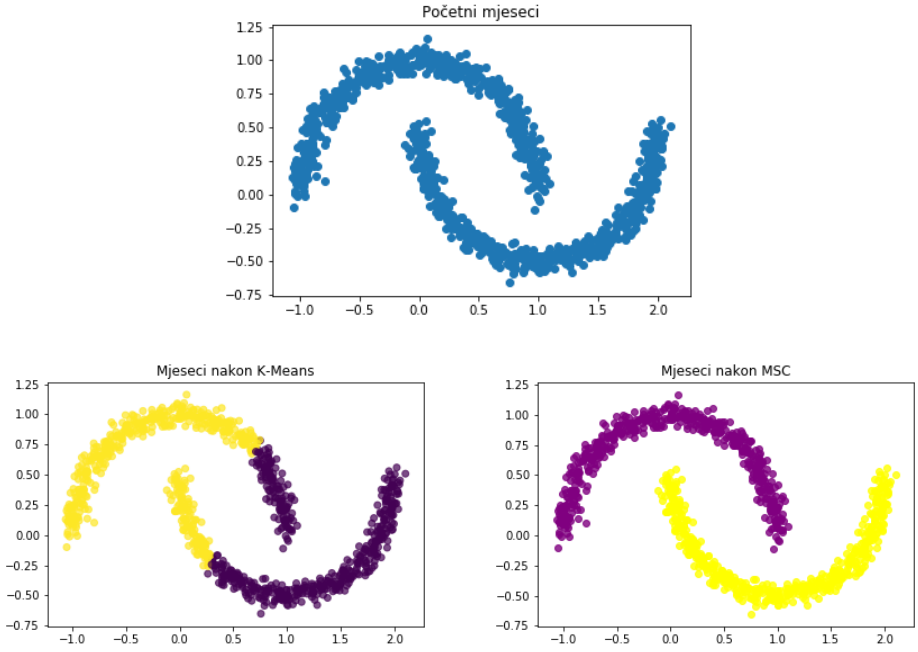




SLIKA 2 Koncentrične kružnice.

4.2 | Polumjeseci

Analognim postupkom kao u prethodnom potpoglavlju, klasterirali smo i takozvane polumjesece generirane pomoću funkcije *make_moons* iz istog paketa *sklearn.datasets*. Ponovno je korišteno 1000 podataka, 10 iteracija i tolerancija 0.001 te ponovno vidimo da je naš *MSC* dao dobre rezultate, dok *K-Means* nije.



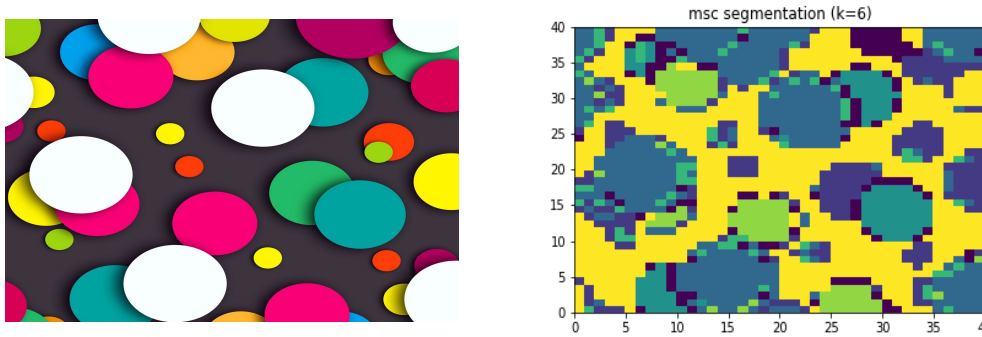
SLIKA 3 Polumjeseci.

4.3 | Segmentacija slike

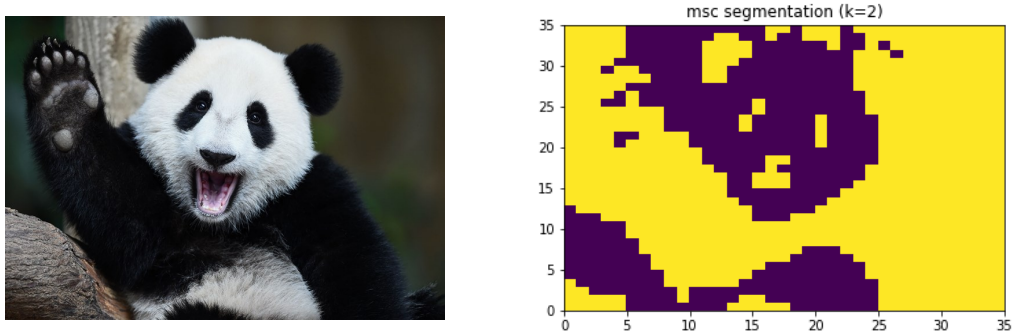
Nadalje provodimo testiranje nad dva različita primjera, prva slika sadrži tridesetak krugova u različitim bojama, a druga slika je slika pande. Slike su preuzete s interneta. Objema smo morali promijeniti veličinu, i to na 40x40 za krugove te 35x35 za pandu jer je za veće dimenzije naše računalo na kojem smo provodili testiranje imalo poteškoća s izvršavanjem koda. Time smo dobili lošije kvalitete slika koje su utjecale i na klasteriranje, no rezultati su zadovoljavajući i dovoljni za naše dokazivanje superiornosti *MSC*-a.

Ovdje se matrica težina W računa na sljedeći način. Slika se transformira u matricu pomoću funkcije *reshape* iz paketa *numpy* tako da se svakom pikselu dodijeli jedan redak matrice gdje su elementi odgovarajuće vrijednosti *rgb (red, green, blue)* boja. Zatim se ponovno primjenjuje funkcija *pairwise_distances* da bismo odredili udaljenosti između tih vrijednosti, odnosno koliko se boje razlikuju, te ponovno dobivamo već spomenutu matricu M. Sada definiramo $W[i,j] = 1$ ako je $M[i,j] < 261$ u slučaju krugova, odnosno $M[i,j] < 21$ u slučaju pande (ove vrijednosti određene su testiranjem), a $W[i,j] = 0$ inače. Sliku krugova testiramo sa šest klastera, a sliku pande s dva klastera.

Ovo su naše originalne slike te rezultat *MSC*-a:



SLIKA 4 Originalna slika krugova i segmentacija dobivena *MSC*-om na 6 klastera.

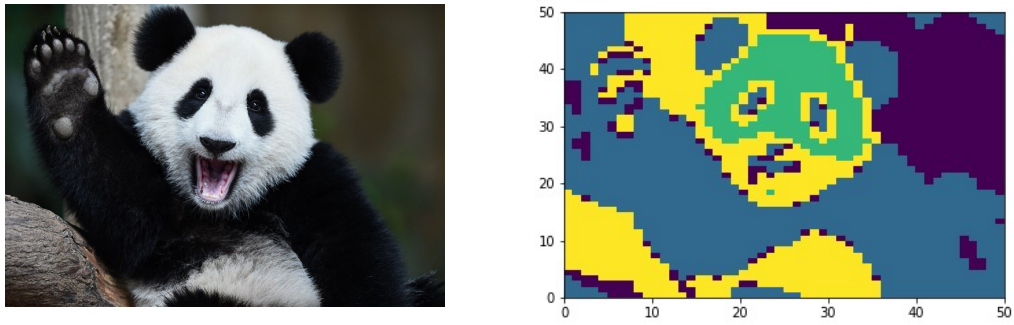


SLIKA 5 Originalna slika pande i segmentacija dobivena *MSC*-om na 2 klastera.

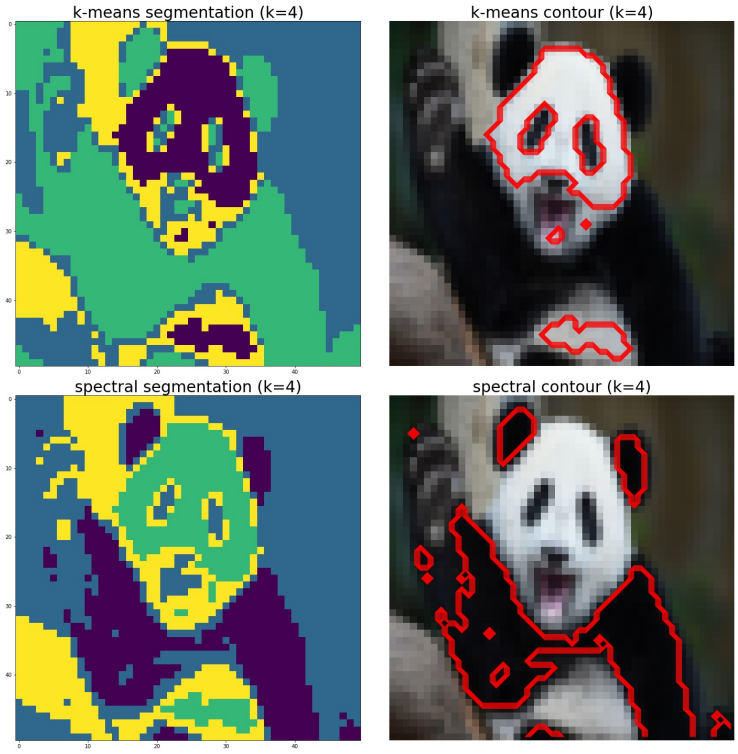
Kako bismo provjerile koliko je zapravo dobra naša implementacija algoritma, usporedili smo je i sa implementiranim *MSC*-ima u paketu *sklearn.cluster* — funkcijama *MiniBatchKMeans* i *SpectralClustering*. Usporedbu provodimo na identičnoj slici pande.

SLIKA 6 *MiniBatchKMeans* i *SpectralClustering* na 2 klastera.

Uspoređujući spektralna klasteriranja na slikama 5 i 6, vidimo da naša implementacija algoritma radi jako dobro. Pokazujemo još jedan primjer segmentacije slike pande, ali ovaj put s četiri klastera:



SLIKA 7 Originalna slika pande i segmentacija dobivena *MSC*-om na 4 klastera.

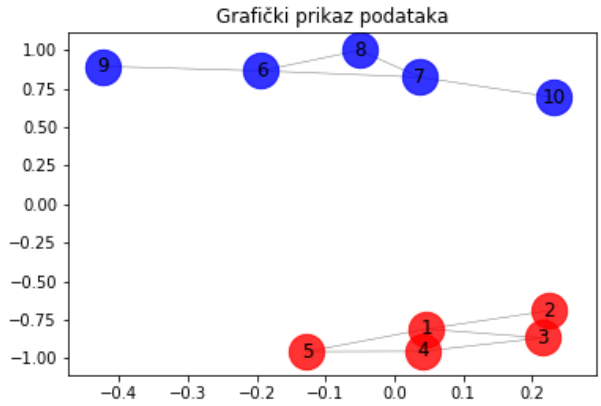


SLIKA 8 *MiniBatchKMeans* i *SpectralClustering* na 4 klastera.

4.4 | Težinski graf

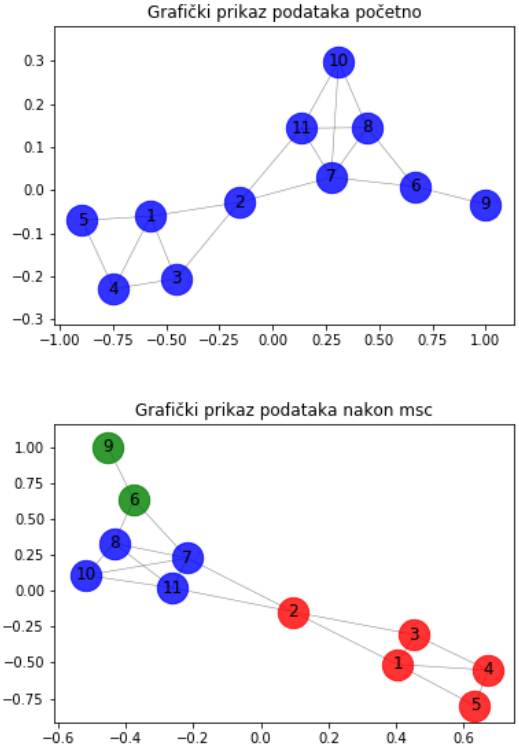
Zadnje testiranje provodimo na dva primjera težinskog grafa. Grafove generiramo pomoću paketa *networkx* tako da se zadaju željeni bridovi između određenog broja čvorova. Matrica težina generira se tako da se na mjesto (i, j) stavi 1 ako je čvor i povezan bridom s čvorom j, a 0 inače.

Prvi primjer pokazuje dva disjunktna podgrafa koje je i naš *MSC* podijelio u dva klastera na jednak način:



SLIKA 9 Tezinski graf s dva ocita klastera.

Drugi primjer prikazuje graf gdje ne uočavamo tako jasne klastere te koji klasteriramo u 3 klastera.



SLIKA 10 Tezinski graf bez jasnih klastera.

Literatura

1 Stella X. Yu, Jianbo Shi. *Multiclass Spectral Clustering*. Robotics Institute and CNBC, Carnegie Mellon University, Pittsburgh, PA 15213-3890, Dept. of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104-6389.

2 Marina Meila, Jianbo Shi. *Learning Segmentation by Random Walks*. University of Washington, Carnegie Mellon University.

3 URL: https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html#sphx-glr-auto-examples-cluster-plot-cluster-comparison-py.

4 URL: https://towardsdatascience.com/spectral-clustering-aba2640c0d5b.

Popis slika