

MULTICLASS SPECTRAL CLUSTERING

Karlo Grozdanić, Mislav Jelašić, Luka Karlič

16. prosinac 2021.

1 Uvod

Klasteriranje, odnosno grupiranje podataka, jedna je od najčešćih tehnika analize podataka koja svoje primjene nalazi u raznim područjima statistike, strojnog učenja, prirodnih znanosti i slično. Smisao klasteriranja je grupirati zadane točke u tzv. klastere na način da određena sličnost među točkama iste grupe bude velika, a među točkama koje pripadaju različitim grupama mala. Algoritmi spektralnog klasteriranja tipično kreću od lokalnih svojstava podataka kako bi procijenili njihovu globalnu strukturu. Tijekom godina razvijeno je više metoda za rješavanje problema, od kojih se posebno ističe kriterij spektralnog normaliziranog reza. Zadatak ovog seminara je, koristeći tehnike spektralne analize, prikazati osnovni algoritam klasteriranja točaka u grafu prema nekom unaprijed zadanom kriteriju.

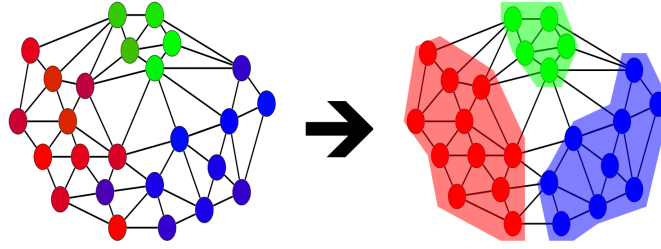


Figure 1: Ideja klasteriranja (grupiranja) podataka putem grafa

2 Spektralno K-particioniranje grafa

2.1 Težinski graf i mjere povezanosti

Problem klasteriranja podataka praktično je prikazati pomoću modela grafa. Definiramo težinski graf kao uređenu trojku $G = (\mathbb{V}, \mathbb{E}, W)$, gdje je $\mathbb{V} = \{v_1, v_2, \dots, v_n\}$ konačan skup vrhova, $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ skup bridova koji ih povezuju i $W = (w_{ij}) \in \mathbb{R}^{n \times n}$ matrica težina koja opisuje vjerojatnost da dva vrha pripadaju istoj grupi (matrica susjedstva). Pretpostavljamo da je W nenegativna i simetrična. Klasteriranje N točaka u K grupa svodi se na traženje K -particije grafa u oznaci $\Gamma_{\mathbb{V}}^K = \{\mathbb{V}_1, \dots, \mathbb{V}_K\}$, odnosno familije nepraznih skupova $\mathbb{V}_1, \mathbb{V}_2, \dots, \mathbb{V}_K$ takvih da je $\mathbb{V} = \cup_{i=1}^K \mathbb{V}_i$ i $\mathbb{V}_i \cap \mathbb{V}_j = \emptyset, \forall i \neq j$. Za skupove $\mathbb{A}, \mathbb{B} \subseteq \mathbb{V}$ definiramo njihov stupanj povezanosti kao sumu težina svih bridova čiji je jedan vrh iz skupa \mathbb{A} , a drugi iz skupa \mathbb{B} :

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

Kako je W simetrična matrica, vrijedi $links(\mathbb{A}, \mathbb{B}) = links(\mathbb{B}, \mathbb{A})$.

Stupanj skupa $\mathbb{A} \subseteq \mathbb{V}$ definiramo kao sumu težina svih bridova čiji je jedan vrh u skupu \mathbb{A} :

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

Relativna povezanost skupa \mathbb{A} sa skupom \mathbb{B} definira se kako omjer stupnja povezanosti skupa \mathbb{A} sa skupom \mathbb{B} i stupnja od \mathbb{A} :

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

Primijetimo da $linkratio(\mathbb{A}, \mathbb{A})$ mjeri udio težina bridova koji ostaju u \mathbb{A} , dok njegov komplement $linkratio(\mathbb{A}, \mathbb{V} \setminus \mathbb{A})$ mjeri udio težina bridova koji pobjegnu iz \mathbb{A} . Kako bismo preciznije opisali povezanosti između elemenata particije uvodimo redom pojmove normalizirane asocijacije i normaliziranog reza particije $\Gamma_{\mathbb{V}}^K$:

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

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

Cilj je odrediti particiju grafa koja istovremeno ima snažnu povezanost unutar pojedinog elementa particije te slabu povezanost između različitih elemenata particije. To se ostvaruje odabirom particije koja maksimizira normaliziranu asocijaciju i minimizira normalizirani rez. Uočimo da su vrijednosti $knassoc(\Gamma_{\mathbb{V}}^K)$ i $kncuts(\Gamma_{\mathbb{V}}^K)$ realni brojevi između 0 i 1, čiji zbroj iznosi točno 1 zbog jednakosti $degree(\mathbb{V}_l) = links(\mathbb{A}, \mathbb{V}) + links(\mathbb{A}, \mathbb{V} \setminus \mathbb{V}_l)$. To znači da je optimalnost u smislu normalizirane asocijacije ekvivalentna optimalnosti u smislu normaliziranog reza

$$\epsilon(\Gamma_{\mathbb{V}}^K) = knassoc(\Gamma_{\mathbb{V}}^K) \longrightarrow \max \iff \varphi(\Gamma_{\mathbb{V}}^K) = kncuts(\Gamma_{\mathbb{V}}^K) \longrightarrow \min$$

po svim K -particijama grafa $\Gamma_{\mathbb{V}}^K$ budući da se one postižu istovremeno. U nastavku ćemo zato promatrati isključivo kriterij maksimiziranja normalizirane asocijacije. Također, pokazat ćemo da se povećanjem broja klastera vrijednost ϵ jednoliko smanjenje.

2.2 Reprezentacija normaliziranih rezova

Neka je $\mathbb{G} = (\mathbb{V}, \mathbb{E}, W)$ težinski graf i $\mathbb{V} = \{v_1, v_2, \dots, v_n\}$. Označimo sa $X \in \mathbb{R}^{N \times K}$ binarnu matricu particije pridruženu particiji $\Gamma_{\mathbb{V}}^K$. Preciznije, $X = [X_1, \dots, X_K]$, gdje je X_l binarni indikator od V_l :

$$X(i, l) = \langle i \in \mathbb{V}_l \rangle = \begin{cases} 1, & i \in \mathbb{V}_l \\ 0, & inače \end{cases}, \quad i \in \mathbb{V}, l \in [K].$$

Ovakvo definirana matrica particije sadrži točno jednu jedinicu u svakom retku pa vrijedi jednakost $X1_K = 1_N$, gdje je $1_M \in \mathbb{R}^M$ vektor stupac koji se sastoji od M uzastopnih jedinica. Za simetričnu težinsku matricu W definiramo dijagonalnu matricu čiji su dijagonalni elementi stupnjevi vrhova:

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

Stupnjeve povezanosti i stupnjeve skupova možemo zapisati u matričnom obliku:

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

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

Kompletna optimizacijska zadaća po varijabli X , nazvana PNCX program, sada glasi:

$$(PNCX) \begin{cases} \text{maksimiziraj } \epsilon(X) = \frac{1}{K} \sum_{l=1}^K \frac{X_l^T W X_l}{X_l^T D X_l} \\ \text{uz } X \in \{0, 1\}^{N \times K}, X1_K = 1_N \end{cases}$$

Ovakav problem minimiziranja reza je NP-težak (čak i za $K = 2$), odnosno nije egzaktno rješiv u polinomijalnom vremenu. Naš zadatak je razviti brzi algoritam koji nalazi približno rješenje.

2.3 Modeliranje problema

Kako bismo iskoristili metode spektralne analize za nalaženje aproksimativnog diskretnog rješenja, potrebno je program PNCX zapisati u obliku svojstvene zadaće. Koristeći jednostavne tvrdnje poput $X^T D X = \text{diag}(X_l^T D X_l)_{l=1}^K$ i $\sum_{l=1}^K X_l^T W X_l = \text{tr}(X^T W X)$ dolazimo do sljedećeg zapisa:

$$\sum_{l=1}^K \frac{X_l^T W X_l}{X_l^T D X_l} = \text{tr}((X^T W X)(X^T D X)^{-1}) = \text{tr}((X^T D X)^{-\frac{1}{2}}(X^T W X)(X^T D X)^{-\frac{1}{2}}).$$

Relaksiranjem nepoznanice na realnu domenu htjeli bismo trenutačni diskretni problem prevesti na odgovarajući neprekidni koji posjeduje globalno optimalno rješenje. Definirajmo normalizacijsku funkciju f koja preslikava matricu X u Z formulom:

$$Z = f(X) = X(X^T DX)^{-\frac{1}{2}}.$$

Konkretno, matrica Z sadrži iste stupce kao početna particijska matrica X , ali podijeljene korijenom stupnjeva particija. Samim time je i Z diskretna matrica koja u svakom retku ima točno jedan element različit od nule. Uočimo da vrijedi i sljedeće:

$$Z^T DZ = (X^T DX)^{-\frac{1}{2}} (X^T DX) (X^T DX)^{-\frac{1}{2}} = I_K,$$

gdje je $I_K \in \mathbb{R}^{K \times K}$ jedinična matrica. Gornjim postupkom dolazimo do nove optimizacijske zadaće, u varijabli Z , koju nazivamo PNCZ program:

$$(PNCZ) \begin{cases} \text{maksimiziraj } \epsilon(Z) = \frac{1}{K} \text{tr}(Z^T W Z) \\ \text{uz } Z^T DZ = I_K \end{cases}$$

Lako se pokaže da su problemi PNCX i PNCZ ekvivalentni, odnosno $X = (X_1, \dots, X_K)$ je rješenje od PNCX ako i samo ako je $Z = (Z_1, \dots, Z_K)$ rješenje od PNCZ. Maksimizaciju funkcije $\epsilon(Z)$, bez uvjeta diskretnosti, realiziramo pomoću poopćenog problema svojstvenih vrijednosti oblika $Wx = \lambda Dx$, odnosno $(D^{-1}W)x = \lambda x$. Međutim, zbog jednostavnosti, rješavat ćemo ekvivalentni simetrični svojstveni problem: $(D^{-\frac{1}{2}}WD^{-\frac{1}{2}})(D^{\frac{1}{2}}x) = \lambda(D^{\frac{1}{2}}x)$. Promotrimo sljedeće rezultate:

Propozicija 1 (*Ortonormalna invarijantnost*) Neka je $R \in \mathbb{R}^{K \times K}$ ortogonalna matrica. Ako je Z rješenje PNCZ programa, tada je rješenje i cijeli skup $\{ZR : R^T R = I_K\}$. Dodatno, $\epsilon(ZR) = \epsilon(Z)$.

Drugim riječima, proizvoljne rotacije i refleksije čuvaju rješenje PNCZ problema (u smislu točnosti). Označimo s $P = D^{-1}W$ normaliziranu matricu težina. Kako je P (retčano) stohastička, lako se provjeri da 1_N njen trivijalni svojstveni vektor koji odgovara najvećoj svojstvenoj vrijednosti 1.

Propozicija 2 (*Optimalno rješenje*) Neka su $V = [V_1, \dots, V_N]$ i $S = \text{Diag}(s_1, \dots, s_N)$, $s_1 \geq \dots \geq s_N$, matrice svojstvenih vektora i svojstvenih vrijednosti od P , odnosno rješenja svojstvene zadaće $PV = PS$. Par (V, S) može se dobiti preko ortonormalne matrice $\bar{V} = [\bar{V}_1, \dots, \bar{V}_N]$ koja rješava simetrični problem svojstvenih vrijednosti matrice $D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$:

$$\begin{aligned} V &= D^{-\frac{1}{2}}\bar{V} \\ D^{-\frac{1}{2}}WD^{-\frac{1}{2}}\bar{V} &= \bar{V}S, \bar{V}^T \bar{V} = I_N \end{aligned}$$

Štoviše, matrice V i S su realne te svakih K različitih svojstvenih vektora čine kandidata za lokalno optimalno rješenje s funkcijom cilja

$$\epsilon([V_{\pi_1}, \dots, V_{\pi_K}]) = \frac{1}{K} \sum_{l=1}^K s_{\pi_l},$$

gdje je π vektor indeksa od K različitih elemenata skupa prirodnih brojeva $\{1, \dots, N\}$. Globalni optimum se stoga postiže za $\pi = [1, \dots, K]$ što daje rješenje problema:

$$\begin{aligned} Z^* &= [V_1, \dots, V_K] \\ \Lambda^* &= \text{Diag}([s_1, \dots, s_K]) \\ \epsilon(Z^*) &= \frac{1}{K} \text{tr}(\Lambda^*) = \max_{Z^T DZ = I_K} \epsilon(Z) \end{aligned}$$

Globalno optimalno rješenje programa PNCZ nije jedinstveno. Prema Ky-Fanovom teoremu ono čini potprostor razepet s prvih K najvećih svojstvenih vektora od P preko ortonormiranih matrica:

$$\{Z^*R : R^T R = I_K, PZ^* = Z^*\Lambda\}$$

Još je potrebno rješenje prikazati u obliku particijske matrice X primjenom inverza normalizirajuće funkcije f^{-1} koji matricu Z vraća natrag u X tako da joj normira retke:

$$X = f^{-1}(Z) = \text{Diag}(\text{diag}^{-\frac{1}{2}}(ZZ^T))Z.$$

Na taj način rješenje Z^*R vraćamo u prostor jedinične sfere. Budući da je R ortogonalna, vrijedi $f^{-1}(Z^*R) = f^{-1}(Z^*)R$ pa čitav skup rješenja polaznog problema možemo zapisati kao:

$$\{\tilde{X}^*R : \tilde{X}^* = f^{-1}(Z^*), R^T R = I_K\}$$

Propozicija 3 (*Ograničenost odozgo*) Za bilo koji prirodni broj K vrijedi:

$$\max \epsilon(\Gamma_{\mathbb{V}}^K) \leq \max_{Z^T D Z = I_K} \epsilon(Z) = \frac{1}{K} \sum_{l=1}^K s_l$$

$$\max_{Z^T D Z = I_{K+1}} \epsilon(Z) \leq \max_{Z^T D Z = I_K} \epsilon(Z)$$

Iz svega navedenoga zaključujemo da je naš optimizacijski problem dobro postavljen. Ipak, jasno se vidi da povećanjem broja klastera dobivamo manje precizna rješenja.

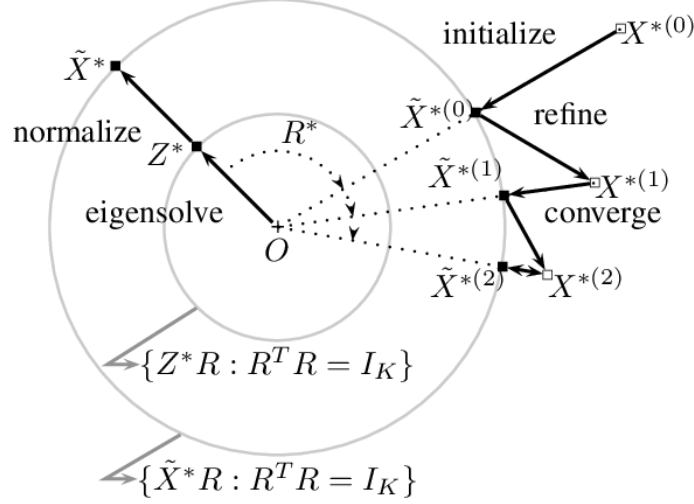


Figure 2: Shematski prikaz algoritma

2.4 Nalaženje optimalnog rješenja

Uz pomoć rješenja Z^* relaksiranog PNCZ problema želimo pronaći diskretno rješenje X^* za PNCX problem. Iako ono nije nužno maksimizator PNCX problema, zbog neprekidnosti je skoro globalno optimalno. Kako sva ekvivalentna rješenja $Z^* R$ maksimiziraju ciljnu funkciju, ima smisla tražiti diskretno rješenje X^* najbliže nekom $Z^* R$. Želimo pronaći particijsku matricu X^* iz pripadnog skupa koja odgovara nekom rješenju $\tilde{X}^* R = f^{-1}(Z^*) R$. Imamo sljedeću varijantu Prokrastovog problema u dvije varijable:

$$(POD) \begin{cases} \text{minimiziraj } \phi(X, R) = \|X - \tilde{X}^* R\|_F^2 \\ \text{uz } X \in \{0, 1\}^{N \times K}, X 1_K = 1_N, R^T R = I_K, \end{cases}$$

gdje $\|M\|_F := \sqrt{\text{tr}(MM^T)}$ označava Frobeniusovu normu matrice M . Kako znamo da vrijedi

$$\|X - \tilde{X}^* R\|_F^2 = \|X\|_F^2 + \|\tilde{X}^*\|_F^2 - 2\text{tr}(X R^T \tilde{X}^{*T}),$$

potrebno je minimizirati samo zadnji član, odnosno maksimizirati vrijednost $\text{tr}(X R^T \tilde{X}^{*T})$. To se uobičajeno postiže SVD dekompozicijom. Gornji nelinearni optimizacijski problem svodimo na dva jednostavnija potproblema u pogledu alternirajućih iteracija najmanjih kvadrata (ALS):

$$(PODX) \begin{cases} \text{minimiziraj } \phi(X) = \|X - \tilde{X}^* R^*\|_F^2 \\ \text{uz } X \in \{0, 1\}^{N \times K}, X 1_K = 1_N, \end{cases}$$

$$(PODR) \begin{cases} \text{minimiziraj } \phi(R) = \|X^* - \tilde{X}^* R\|_F^2 \\ \text{uz } R^T R = I_K, \end{cases}$$

gdje je u PODX programu fiksiran $R = \tilde{R}$, a u PODR programu $X = X^*$. Dakle, prvo nađemo neprekidno rješenje $\tilde{X}^* R$ najbliže danom X^* , a zatim odredimo njemu najbliže diskretno rješenje. Rješenja zapisanih problema daju nam sljedeći teoremi iz spektralne analize:

Teorem 4 Neka je $\tilde{X} = \tilde{X}^* R^*$. Optimalno rješenje programa PODX je dano s:

$$X(i, l) = \langle l = \arg \max_{k \in [K]} \tilde{X}(i, k) \rangle, i \in \mathbb{V}$$

Teorem 5 Rješenje programa PODR dano je SVD dekompozicijom matrice $X^* \tilde{X}^*$:

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

$$X^* \tilde{X}^* = U \Omega \tilde{U}^T, \Omega = \text{Diag}(\omega_1, \dots, \omega_K),$$

gdje je $U^T U = I_K$, $\tilde{U}^T \tilde{U} = I_K$ i $\omega_1 \geq \dots \geq \omega_K$.

Prikazani postupak pronalaženja rješenja spektralnog klasteriranja, uz pomoć matrice rotacije, zovemo spektralna rotacija.

2.5 Algoritam (NC)

Dana je matrica težina W i broj elemenata particije K :

1. Izračunaj matricu stupnjeva $D = \text{Diag}(W 1_N)$
2. Nađi optimalno rješenje Z^* svojstvenog problema:

$$D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \bar{V}_{[K]} = \bar{V}_{[K]} \text{Diag}(s_{[K]}), \bar{V}_{[K]}^T \bar{V}_{[K]} = I_{[K]}$$

$$Z^* = D^{-\frac{1}{2}} \bar{V}_{[K]} \text{ (opcionarno)}$$

3. Normaliziraj Z^* : $\tilde{X}^* = \text{Diag}(\text{diag}^{-\frac{1}{2}}(Z^* Z^{*T})) Z^*$
4. Inicijaliziraj X^* preko R^* na sljedeći način:

$$R_1^* [\tilde{X}^*(i, 1), \dots, \tilde{X}^*(i, K)]^T, i \in [N] \text{ slučajan}$$

$$c = 0_{N \times 1}$$

$$\text{Za } k = 2, \dots, K \text{ radi :}$$

$$c = c + \text{abs}(\tilde{X}^* R_{k-1}^*)$$

$$R_k^* = [\tilde{X}^*(i, 1), \dots, \tilde{X}^*(i, K)]^T, i = \arg \min c$$

5. Postavi parametar konvergencije $\bar{\phi}^* = 0$.
6. Nađi optimalno diskretno rješenje X^* :

$$\tilde{X} = \tilde{X}^* R^*$$

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

7. Nađi optimalnu ortogonalnu matricu R^* pomoću SVD dekompozicije:

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

$$\bar{\phi} = \text{tr}(\Omega)$$

$$\text{Ako je } |\bar{\phi} - \bar{\phi}^*| < \text{mašinska preciznost, stani i ispiši } X^*$$

$$\bar{\phi}^* = \bar{\phi}$$

$$R^* = \tilde{U} U^T$$

8. Idi na korak 6.

Vremenska složenost, koju izražavamo u terminima N i K , ovisi o implementaciji. Najviše vremena utroši se u 2. koraku za dobivanje matrice $V_{[K]}$ pomoću svojstvenih vektora, čija je složenost $\mathcal{O}(NK^2)$. 4. korak ima ukupno $NK(K-1)$, a 6. korak NK^2 množenja. Dekompozicija singularnih vrijednosti u 7. koraku može se postići u vremenu $\mathcal{O}(K^3)$. Kako je X^* binarna matrica, ona se efikasno dobija sa K^2N zbrajanja. Cjelokupna metoda ima vremensku složenost $\mathcal{O}(KN(K+N))$.

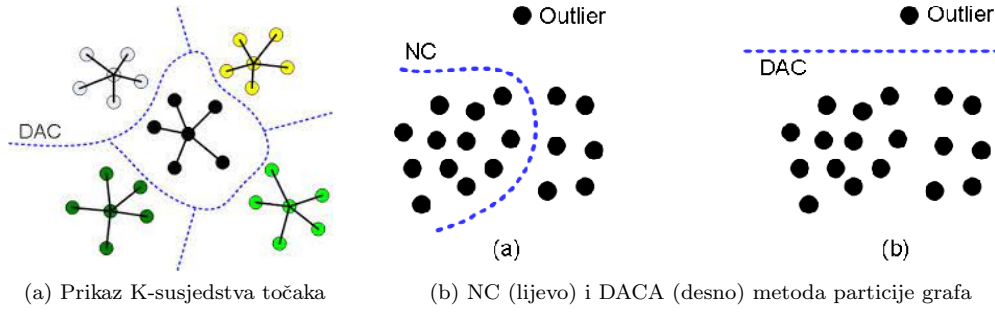


Figure 3: Ilustracija klasteriranja pomoću DAC algoritma

2.6 Metoda bazirana na diskriminantnoj analizi

Prije prezentiranja samog koncepta uvedimo dodatne oznake. Neka je $W = (w_{ij})_{N \times N}$ i dalje matrica težina, a $W^* = (w_{ij}^*)$ matrica težina k-najbližih vrhova (susjeda) dana sa:

$$w_{ij}^* = \begin{cases} 1, & \text{ako } j \in N_k(i) \text{ ili } i \in N_k(j) \\ 0, & \text{inače,} \end{cases}$$

gdje $N_k(i)$ označava skup k-najbližih susjeda vrha v_i . Neka je još $\widehat{W} = (\widehat{w}_{ij})_{N \times N} = (w_{ij} \cdot w_{ij}^*)_{N \times N}$, \widehat{D} dijagonalna matrica dana s $\widehat{d}_{ii} = \sum_j \widehat{w}_{ij}$ za $1 \leq i, j \leq N$ i $Q = \widehat{D} - \widehat{W}$. Ako je Q singularna matrica, zamjenjujemo ju s $Q + \epsilon I_N$ za neki mali $\epsilon > 0$, pri čemu je I_N identiteta. Novi DAC problem sada je dan sa:

$$(DAC) \begin{cases} \text{maksimiziraj } g(X) = \frac{1}{K} \sum_{n=1}^K \frac{X_n^T \widehat{W} X_n}{X_n^T Q X_n} = \frac{1}{K} \sum_{n=1}^K \frac{[X_n (X_n^T X_n)^{-\frac{1}{2}}]^T \widehat{W} [X_n (X_n^T X_n)^{-\frac{1}{2}}]}{[X_n (X_n^T X_n)^{-\frac{1}{2}}]^T Q [X_n (X_n^T X_n)^{-\frac{1}{2}}]} \\ \text{uz } X \in \{0, 1\}^{N \times K}, X 1_K = 1_N, \end{cases}$$

gdje X_n predstavlja n -ti stupac od X . Kompaktnost unutar klase i separabilnost između različitih klasa dostižu se preko $X_n^T \widehat{W} X_n$, odnosno $X_n^T Q X_n$, danih s:

$$X_n^T \widehat{W} X_n = \sum_{i \in \mathbb{V}_n} \sum_{j \in B_i} w_{ij}; \quad X_n^T Q X_n = \sum_{i \in \mathbb{V}_n} \sum_{j \in \bar{B}_i} w_{ij},$$

gdje je $B_i = \{u | u \in \mathbb{V}_n, i \in N_k(u) \text{ ili } u \in N_k(i)\}$, $\bar{B}_i = \{u | u \notin \mathbb{V}_n, i \in N_k(u) \text{ ili } u \in N_k(i)\}$, a \mathbb{V}_n označuje skup vrhova n -te klase. Što je vrijednost $X_n^T \widehat{W} X_n$ veća, uzorci unutar klase su kompaktniji, a što je vrijednost $X_n^T Q X_n$ manja, uzorci između klasa su separabilniji. Rezultat toga je da se optimalno rješenje particije grafa postiže maksimiziranjem funkcije $g(X)$ po X . Jasno je da je DAC sposoban očuvati topološke strukture sličnosti grafa konstruirajući podgraf k -najbližih susjeda za svaki vrh grafa, što dovodi do robusnosti particioniranja. Princip particioniranja DAC prikazan je slikom 3, gdje je za svaki vrh konstruiran podgraf 5 najbližih susjeda. Optimalna granica particioniranja, označena isprekidanom crtom, dobivena je putem DAC algoritma analiziranjem unutarklasnih rubnih svojstava grafa.

2.7 Pronalaženje optimalnog rješenja

Nakon slijeda algebarskog pojednostavljivanja kriterij maksimiziranja particije postaje: $g(X) = \frac{1}{K} \text{tr}\{(P^T Q P)^{-1} (P^T \widehat{W} P)\}$, za $P = X(X^T X)^{-\frac{1}{2}}$, uz uvjet $P^T P = [X(X^T X)^{-\frac{1}{2}}]^T [X(X^T X)^{-\frac{1}{2}}] = I_K$. Imamo sljedeću zadaću:

$$(DAC) \begin{cases} \text{maksimiziraj } h(P) = \frac{1}{K} \text{tr}\{(P^T Q P)^{-1} (P^T \widehat{W} P)\} \\ \text{uz } P^T P = I_K \end{cases}$$

Rješenje \tilde{P} gornjeg optimizacijskog problema sastoji se od K svojstvenih vektora koji pripadaju prvim K najvećim svojstvenim vrijednostima matrice $Q^{-1} \widehat{W}$. Ako je Q singularna, matricu $Q^{-1} \widehat{W}$ treba zamijeniti s $(Q + \epsilon I_N)^{-1} \widehat{W}$, gdje je $\epsilon > 0$ mali, a I_N identiteta. Kao rezultat, kandidat za rješenje particioniranja grafa \tilde{X} dano je formulom

$$\tilde{X} = \text{Diag}(\text{diag}^{-\frac{1}{2}}(\tilde{P} \tilde{P}^T)) \tilde{P}.$$

Konačno, iterativni proces poboljšavanja definiran ranije može se koristiti za pronalaženje optimalnog rješenja particije grafa.

2.8 Algoritam (DACA)

Dan je skup podataka $Z = \{z_1, z_2, \dots, z_n\}$ i broj elemenata particije K :

1. Kreiraj graf $G = (V, E, W, W^*)$, gdje je $V = \{1, \dots, N\}$ skup vrhova, $E \subseteq V \times V$ skup bridova, $W^* = (w_{ij}^*)_{N \times N}$ definirana ranije, $W = (w_{ij})_{N \times N}$, npr., $w_{ij} = \exp(-\text{dist}(z_i, z_j)/2\sigma^2)$, gdje je σ faktor skaliranja, a $\text{dist}(\cdot)$ predstavlja neku funkciju udaljenosti.
2. Izračunaj $\widehat{W} = (\widehat{w}_{ij})_{N \times N} = (w_{ij} \cdot w_{ij}^*)_{N \times N}$ i $Q = \widehat{D} - \widehat{W}$, \widehat{D} dijagonalna dana s $\widehat{d}_{ii} = \sum_j \widehat{w}_{ij}$ za $1 \leq i, j \leq N$. Ako je Q singularna matrica, zamjenjujemo ju s $Q + \epsilon I_N$.
3. Odredi \tilde{P} kao K najvećih normaliziranih svojstvenih vektora $Q^{-1}\widehat{W}$.
4. Nađi kandidata za rješenje particije grafa: $\tilde{X} = \text{Diag}(\text{diag}^{-\frac{1}{2}}(\tilde{P}\tilde{P}^T))\tilde{P}$
5. Izvedi iterativnu proceduru poboljšanja rješenja na \tilde{X} opisanu u algoritmu (NC).

3 Implementacija i testiranje

3.1 Funkcije

Opisati što smo sve radili u MATLAB-u. Napisati koje su funkcije za što, koje ulazne/izlazne podatke primaju i specifičnosti implementiranja tih funkcija.

- Iter - implementacija koraka 6. - 8. opisanih u seminaru
- DACA - implementacija algoritma DACA opisanog u seminaru
- Dist - funkcija za računanje matrice težina W
- Draw - pomoćna funkcija za vizualizaciju rezultata clusteriranja 2d točaka
- getW_z - pomoćna funkcija za računanje matrice težinskih susjedstva W^* . Optimizirana u slučajevima kada je matrica težina W rijetko popunjena.
- Generate_X - pomoćne funkcije za generiranje podataka za clusteriranje
- NC - implementacija algoritma NC opisanog u seminaru
- ImageDist - funkcija za računanje matrica težina W za RGB sliku
- DrawImg - pomoćna funkcija za vizualizaciju rezultata clusteriranja slika

Implementirane posebnosti i moguća poboljšanja:

- Korištenje vektoriziranih operacije što dovodi do bržeg izvođenja programa.
- Dodatno koristimo informacije o udaljenosti piksela uz informacije o razlici boja.
- Umjesto RGB color space-a koji nije pogodan za euklidsku udaljenost prvo koristimo CIELAB color space, a kasnije još bolji Oklab (nastao 2020.).

3.2 Testovi

Valjanost implementiranih metoda odlucili smo testirati na stvarnim podacima, ali i na umjetno stvorenima budući da na taj način možemo kontrolirati uvjete, stoga naglasiti mane, odnosno prednosti pojedine metode.

Pomoćne funkcije za generiranje podataka

- Generate1 - generira nasumične točke na dijagonali prvog kvadranta tako da su klasteri točaka međusobno razmaknuti; kontroliramo broj generiranih točaka i klastera
- Generate2 - generira nasumične klastere točaka podjednako raspršene po prvom kvadrantu; kontroliramo broj točaka u klasteru te stupčanu, odnosno retčanu, poziciju pojedinog klastera

U prethodnim primjerima očekuje se da obje metode uspješno klasteriraju točke prema očiglednim klasterima. Međutim, zašto bismo uopće implementirali dvije različite metode ako se ponašaju jednako?

Iz tog razloga ključno je uočiti slučajeve u kojima je metoda bazirana na diskriminanti (DACA) superiorna u odnosu na metodu normaliziranih rezova (NC).

Uvedimo iduće dvije funkcije za generiranje podataka.

- Generate3 - ponaša se slično kao i Generate2, međutim ključna razlika je u tome što jednu točku izoliramo, stoga predstavlja outlier.
- Generate4 - metoda koja generira točke cirkularno na dvije kružnice različitih radijusa.

3.3 Rezultati

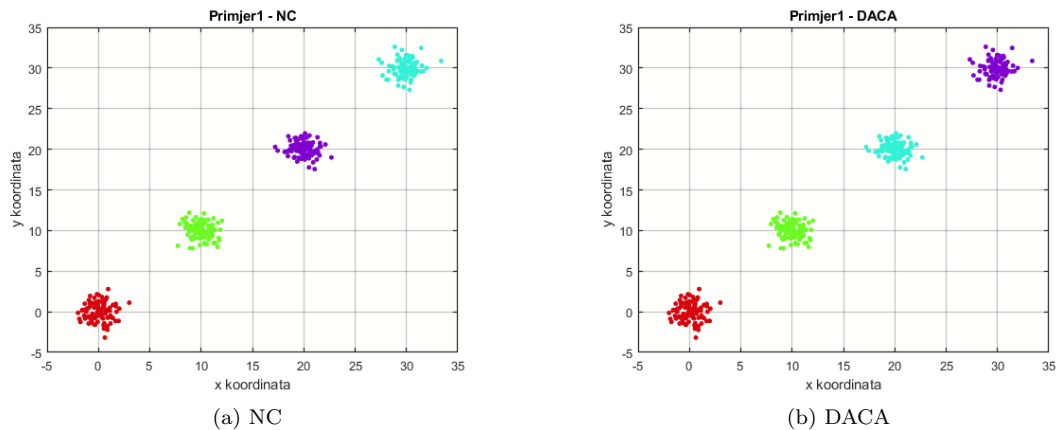


Figure 4: Rezultat klasteriranja točaka dobivenih funkcijom Generate1

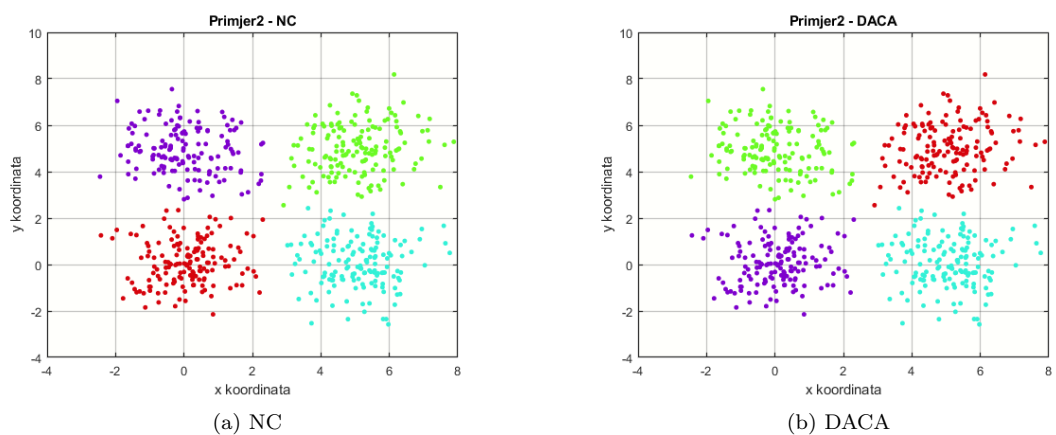


Figure 5: Rezultat klasteriranja točaka dobivenih funkcijom Generate2

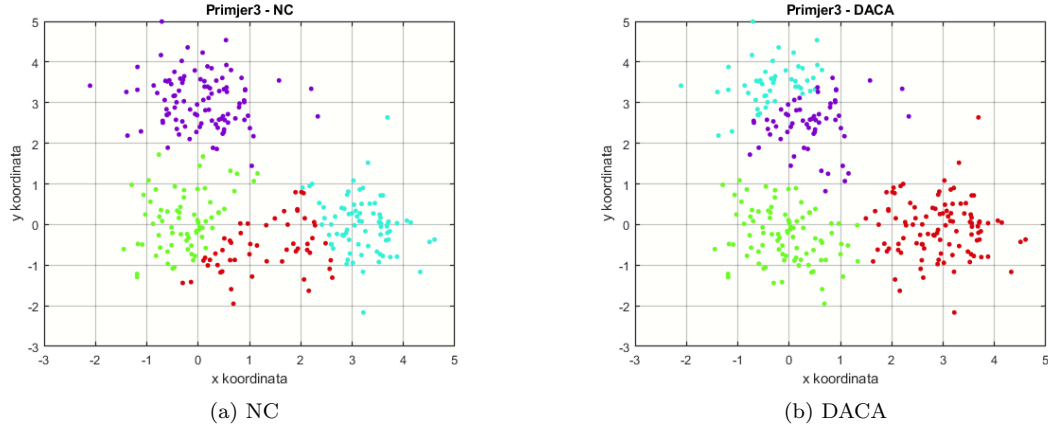


Figure 6: Rezultat klasteriranja tocaka dobivenih funkcijom Generate3

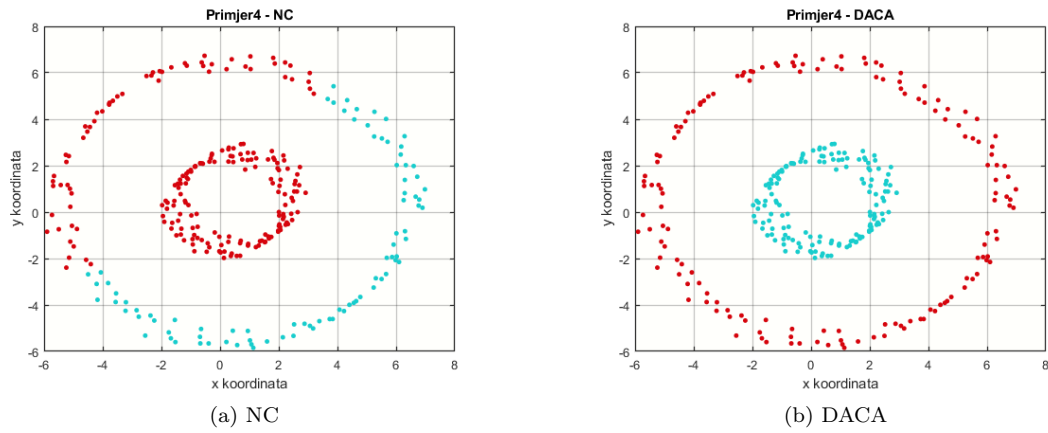


Figure 7: Rezultat klasteriranja tocaka dobivenih funkcijom Generate4

4 Primjer: segmentacija slike

U području računalnog vida od posebnog je interesa specifičan oblik klasteriranja koji nazivamo semantička segmentacija. Ideja je na danoj slici razdvojiti dijelove koji pripadaju istoj 'grupi piksela' na temelju semantičkog značenja. Iako semantička segmentacija slike nije fokus našeg rada, uočili smo da pripadne metode, NC te DACA, uspijevaju donekle uspješno rješavati zadani problem.

4.1 Algoritam

Neka je dana slika $I \in \mathbb{R}^{m \times n \times 3}$ te broj segmenata k .

1. Izračunaj matricu $W = (w_{ij}) \in \mathbb{R}^{l \times l}$ težinskih vrijednosti, pri čemu je $l = mn$. $W = (w_{ij})$ predstavlja koliko su pikseli i i j slike I međusobno povezani. Matrica W je nenegativna (po elementima) te simetrična, $W = W^T$.

U ovom koraku možemo na mnoge načine računati povezanost piksela primjenjujući različite funkcije sličnosti.

2. Na danoj matrici W provedi željeni algoritam - NC ili DACA.

4.2 Funkcije povezanosti piksela

Promatrali smo uspješnost segmentacije primjenjivanjem različitih funkcija za dobivanje matrice W težinskih koeficijenata. Udaljenosti piksela promatramo samo na pikselima čije su udaljenosti pozicija manje od *radijusa*. Ostale smatramo nesusjednim te je njihova povezanost 0.

Za funkciju povezanosti koristimo $w_{ij} = e^{-\frac{d(f_i, f_j)}{2\sigma^2}}$ gdje je d standardna metrika u \mathbb{R}^n prostoru, n ovisi o dimenziji podataka, f_i boja tog piksela, a σ skalirajući faktor.

- Grayscale: Dimenzija prostora boja piksela je R .
- RGB / Oklab: Dimenzija prostora boja piksela je R^3 .
- Dist: Na funkciju udaljenosti dodajemo uvjet udaljenosti piksela $e^{\frac{d(x_i, x_j)}{-2\sigma_x^2}}$, gdje je x_+ pozicija tog piksela na slici, a σ_x dodatni skalirajući faktor za udaljenost piksela.

5 Rezultati semantičke segmentacije

Problem semantičke segmentacije prionuli smo relativno naivno. Rezultati se mogu znatno poboljšati uvođenjem novih informacija, primjerice o tome gdje se nalaze rubovi predmeta. Jedan od načina na koji to možemo postići je konvolucijom gradijentnim filtrima.

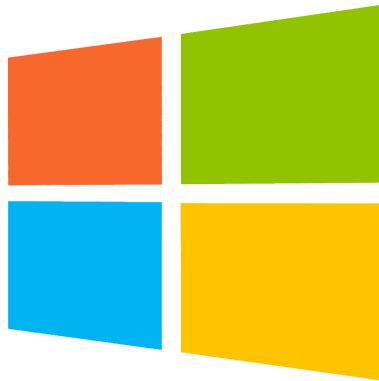
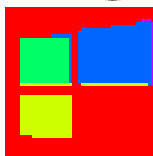


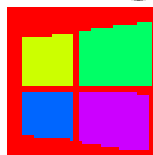
Figure 8: Originalna slika

DACA, gray



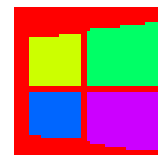
(a)

DACA, rgb



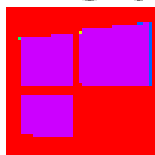
(b)

DACA, Oklab



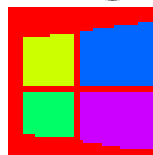
(c)

NC, gray



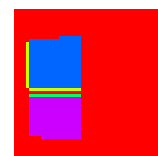
(d)

NC, rgb



(e)

NC, Oklab



(f)

Figure 9: Rezultat segmentacije: samo vrijednosti piksela



Figure 10: Originalna slika

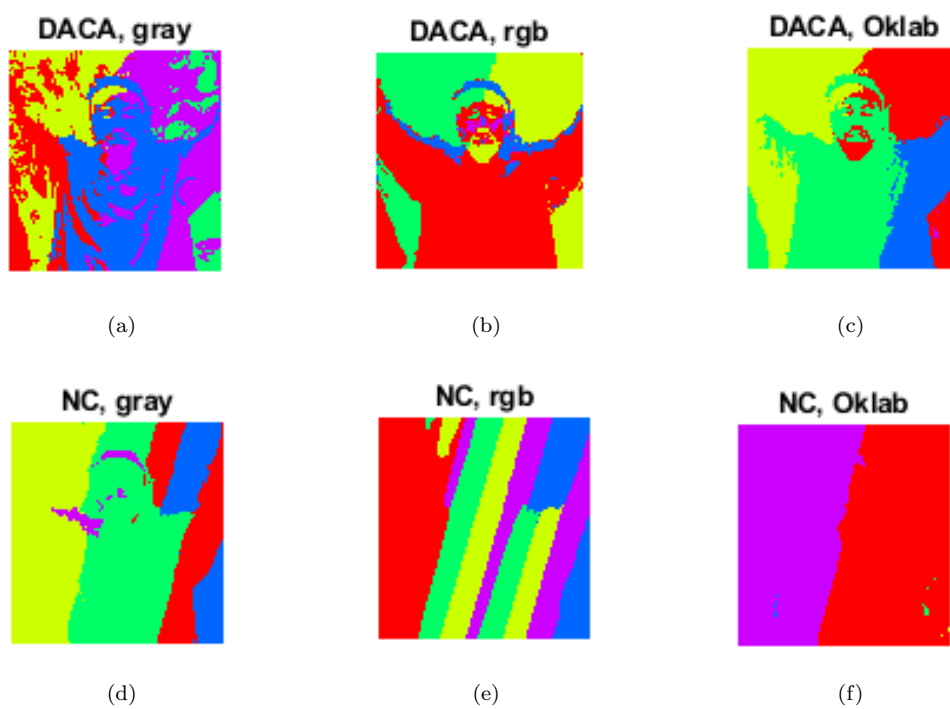


Figure 11: Rezultat segmentacije: samo vrijednosti piksela



Figure 12: Originalna slika

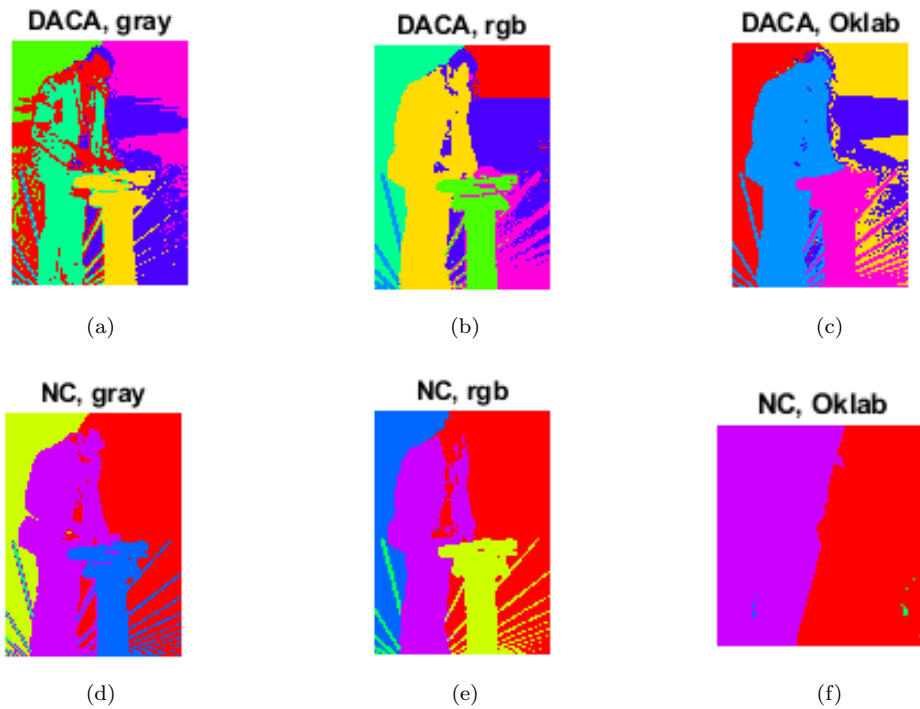


Figure 13: Rezultat segmentacije: samo vrijednosti piksela

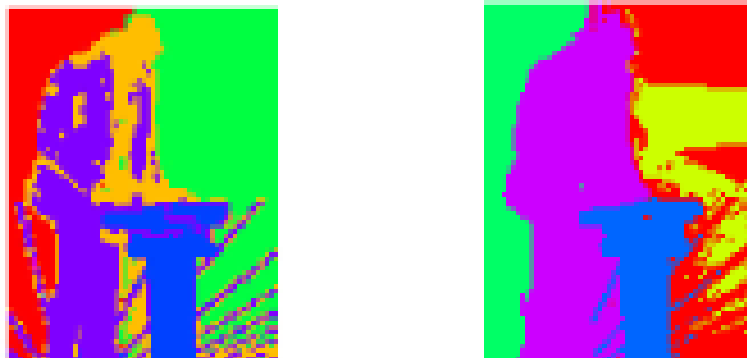


Figure 14: Segmentacija uzevši u obzir i udaljenost piksela. Grayscale (lijevo), Oklab (desno)

6 Reference

1. S. X. Yu i J. Shi, *Multiclass Spectral Clustering*
2. X. Li, Z. Zhang, Y. Wang, W. Hu, *Multiclass Spectral Clustering Based on Discriminant Analysis*
3. J. Shi, J. Malik, *Normalized Cuts and Image Segmentation*
4. B. Ottosson, <https://bottosson.github.io/posts/oklab/>