SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO - MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

Uvod u složeno pretraživanje podataka

Simultano klasteriranje dokumenata i riječi

Petra Sočo, Jelena Zaninović
Zagreb, 20.12.2020.

Sadržaj

1 Uvod

Klasteriranje kolekcije podataka je sasvim općenito grupiranje elemenata na temelju sličnosti. Skup podataka koji imamo na raspolaganju mogu biti primjerice riječi ili dokumenti. Većina postojećih algoritama klasterira riječi i dokumente odvojeno. U pozadini klasteriranja dokumenata je distribucija riječi koje se pojavljuju: dva dokumenta su slična ako se u njima pojavljuju iste riječi. Grupiranje riječi ide po principu zajedničkog pojavljivanja: dvije riječi su slične ako se zajedno pojavljuju u nekom dokumentu. Iz prethodnog se može primijetiti dualnost između ta dva procesa i ima ih smisla pokušati simultano provesti. Jedan od pristupa može biti da promatramo riječi i dokumenate kao bipartitni graf pa se tada problem simultanog klasteriranja svodi na problem particioniranja bipartitnog grafa. Algoritam će se sastojati od dva dijela: dekompozicije singularnih vrijednosti pripadajuće matrice i provođenja k-means algoritma na jednodimenzionalnom skupu podataka. U danjem tekstu opravdavamo taj postupak prateći teoriju iz [1] i provodimo eksperimente na različitim skupovima podataka.

2 Model bipartitnog grafa

Težinski, neusmjereni graf je uređeni par $G = (\mathcal{V}, E)$ zadan skupom vrhova $\mathcal{V} = \{1, 2, \ldots, |\mathcal{V}|\}$ i bridova $\{i, j\}$, gdje svaki brid ima težinu E_{ij}. Za takav graf možemo definirati matricu susjedstva $\mathbf{M}$ na sljedeći način:

$$M_{ij} := \begin{cases} E_{ij} & \exists\, brid\ \{i,j\}, \\ 0 & inače. \end{cases}$$

k-particija skupa S je familija $\mathcal{P} = \{U_1, \ldots, U_k\}$, $U_i \subseteq S$ takvih da: $\emptyset \notin \mathcal{P}$; $\bigcup_{i=1}^{k} U_i = S$ i $U_i \cap U_j = \emptyset$ za $i \neq j$. Neka je $\mathcal{V}_1$ i $\mathcal{V}_2$ proizvoljna 2-particija skupa $\mathcal{V}$. Njihova razlika može se mjeriti kao ukupna težina bridova koji ih povezuju i tu mjeru povezanosti nazivamo rez particije, tj. definiramo:

$$cut(\mathcal{V}_1, \mathcal{V}_2) := \sum_{i \in \mathcal{V}_1, j \in \mathcal{V}_2} M_{ij}.$$

Prethodna definicija može se lako poopćiti na k-particiju:

$$cut(\mathcal{V}_1, \ldots, \mathcal{V}_k) := \sum_{i<j} M_{ij}.$$

Uvedimo sada pojam bipartitnog grafa. Neusmjereni, bipartitni graf je uređena trojka $G = (\mathcal{D}, \mathcal{W}, E)$ gdje su $\mathcal{D} = \{d_1, \ldots, d_n\}$ i $\mathcal{W} = \{w_1, \ldots, w_m\}$ dva skupa vrhova, a $E = \{\{d_i, w_j\} : d_i \in \mathcal{D}, w_j \in \mathcal{W}\}$ je skup bridova. U našem slučaju $\mathcal{D}$ je skup dokumenata, a $\mathcal{W}$ skup riječi koje su sadržane u njima. Brid $\{d_i, w_j\}$ postoji ako se riječ w_j pojavljuje u dokumentu d_i, a

jačina te veze izražena je pozitivnom težinom E_{ij} koju postavljamo na brid koji ih povezuje. Matrica susjedsva bipartitnog grafa $\mathbf{M}$ ima oblik

$$\mathbf{M} = \begin{bmatrix} 0 & \mathbf{A} \\ \mathbf{A}^T & 0 \end{bmatrix},$$

gdje je $\mathbf{A}$ dimenzije $m \times n$ i vrijedi $A_{ij} = E_{ij}$. Dakle, prvih m redaka i stupaca odgovaraju riječima, a zadnjih n odgovaraju dokumentima.

Cilj je simultano grupirati dokumente i riječi, a u pozadini je činjenica da klasteriranje riječi generira klastere dokumenata i obratno. Pretpostavimo da smo dokumente grupirali u k klastera $\mathcal{D}_1, \ldots, \mathcal{D}_k$. Tada riječ w_i pripada klasteru $\mathcal{W}_m$ ako je njena veza s klasterom $\mathcal{D}_m$ jača od veza s ostalim klasterima. Jačinu veze smo mjerili težinom na bridovima pa tražene klastere riječi $\mathcal{W}_m$ možemo karakterizirati s

$$\mathcal{W}_m = \{w_i : \sum_{j \in \mathcal{D}_m} A_{ij} \geq \sum_{j \in \mathcal{D}_l} A_{ij},\ \forall l = 1, \ldots, k\},\ m = 1, \ldots, k.$$

Slično, uz zadane kalstere riječi $\mathcal{W}_1, \ldots, \mathcal{W}_k$, možemo grupirati dokumente

$$\mathcal{D}_m = \{d_j : \sum_{i \in \mathcal{W}_m} A_{ij} \geq \sum_{i \in \mathcal{W}_l} A_{ij},\ \forall l = 1, \ldots, k\},\ m = 1, \ldots, k.$$

"Najbolje" grupiranje dokumenata i riječi predstavljeno je nekom particijom bipartitnog grafa. Dakle, tražimo particiju grafa u kojoj će težine koje povezuju različite klastere biti što manje moguće, a to postižemo kad je rez particije minimiziran, tj.

$$cut(\mathcal{W}_1 \cup \mathcal{D}_1, \ldots, \mathcal{W}_k \cup \mathcal{D}_k) = \min_{\mathcal{V}_1, \ldots, \mathcal{V}_k} cut(\mathcal{V}_1, \ldots, \mathcal{V}_k),$$

gdje je $\mathcal{V}_1, \ldots, \mathcal{V}_k$ neka k-particija grafa.

3 Uloga svojstvenog problema u particioniranju grafa

Nalaženje optimalne particije grafa svodi se na nalaženje podskupova $\mathcal{V}_1^*$ i $\mathcal{V}_2^*$ takvih da je $cut(\mathcal{V}_1, \mathcal{V}_2)$ minimiziran. Rješavanju takvog problema može se pristupiti na više načina, ali ovdje ćemo usmjeriti pažnju na spektralno particioniranje grafa koje će aproksimirati traženi minimizator.

Neka je graf $G = (\mathcal{V}, E)$ zadan s n vrhova i m bridova kojima su pridružene težine E_{ij}. Definiramo prvo $n \times m$ matricu incidencije $\mathbf{I}_G$. Stupac matrice $\mathbf{I}_G$ koji pripada bridu $\{i, j\}$ ima na i-tom i j-tom mjestu $\sqrt{E_{ij}}$, $-\sqrt{E_{ij}}$ redom, dok su na ostalim mjestima nule.

Definicija 1 *Laplaceova matrica* $\mathbf{L} = \mathbf{L}_G$ *grafa* G *je* $n \times n$ *simetrična matrica takva da*

$$L_{ij} = \begin{cases} \sum_k E_{ik} & i = j, \\ -E_{ij} & i \neq j,\ \exists\, brid\ \{i, j\}, \\ 0 & inače. \end{cases}$$

Teorem 2 *Laplaceova matrica* $\mathbf{L} = \mathbf{L}_G$ *grafa* G *ima sljedeća svojstva:*

1. $\mathbf{L} = \mathbf{D} - \mathbf{M}$*, gdje je* $\mathbf{M}$ *matrica incidencije, a* $\mathbf{D}$ *dijagonalna matrica takva da* $D_{ii} = \sum_k E_{ik}$*.*
2. $\mathbf{L} = \mathbf{I}_G {\mathbf{I}_G}^T$*.*
3. $\mathbf{L}$ *je simetrična pozitivno semi-definitna matrica.*
4. *Neka je* $\mathbf{e} = [1, \ldots, 1]^T$*. Tada je* $\mathbf{Le} = \mathbf{0}$*.*
5. *Ako graf* G *ima* c *komponenti povezanosti, tada* $\mathbf{L}$ *ima* c *svojstvenih vrijednosti jednakih* 0*.*
6. *Za proizvoljni vektor* $\mathbf{x}$*,* $\mathbf{x}^T\mathbf{L}\mathbf{x} = \sum_{\{i,j\}\in E} E_{ij}(x_i - x_j)^2$*.*
7. *Za proizvoljni vektor* $\mathbf{x}$ *i skalare* α *i* β *vrijedi*

$$(\alpha\mathbf{x} + \beta\mathbf{e})^T\mathbf{L}(\alpha\mathbf{x} + \beta\mathbf{e}) = \alpha^2\mathbf{x}^T\mathbf{L}\mathbf{x}.$$

Napomena: Iz 3. tvrdnje slijedi da su su sve svojstvene vrijednosti matrice $\mathbf{L}$ realne i ne-negativne, a pomoću 4 vidimo da je $(\mathbf{0}, \mathbf{e})$ jedan svojstveni par matrice $\mathbf{L}$.

Dokaz: *1* i *2* slijede iz definicije matrice $\mathbf{L}$ i direktnim računom, tj. množenjem matrica. Iz tvrdnje *2* za proizvoljni vektor $\mathbf{x}$ imamo $\mathbf{x}^T\mathbf{L}\mathbf{x} = \mathbf{x}^T\mathbf{I}_G{\mathbf{I}_G}^T\mathbf{x} = \mathbf{y}^T\mathbf{y} \geq 0$ iz čega slijedi tvrdnja *3*.

Iz rastava $\mathbf{L}\mathbf{x} = \mathbf{I}_G({\mathbf{I}_G}^T\mathbf{x})$ za proizvoljni vektor $\mathbf{x}$, vidimo da vrijedi

$$({\mathbf{I}_G}^T\mathbf{x})_k = \sqrt{E_{ij}}(x_i - x_j) \tag{1}$$

pa kad je $\mathbf{x} = \mathbf{e}$, tada je $\mathbf{Le} = \mathbf{0}$. Time je dokazana *4.* tvrdnja. *6.* slijedi iz raspisa (1), a *7.* iz tvrdnje *4*.

□

Nadalje pretpostavljamo da graf G ima jednu komponentu povezanosti. Za proizvoljnu 2-particiju $\mathcal{V}_1, \mathcal{V}_2$ definiramo vektor $\mathbf{p}$ kao vektor particije na sljedeći način

$$p_i := \begin{cases} +1 & i \in \mathcal{V}_1 \\ -1 & i \in \mathcal{V}_2. \end{cases}$$

Teorem 3 *Neka je* $\mathbf{L}$ *Laplaceova matrica grafa* G *i* $\mathbf{p}$ *vektor particije. Tada za Rayleighov koeficijent vrijedi*

$$\frac{\mathbf{p}^T\mathbf{L}\mathbf{p}}{\mathbf{p}^T\mathbf{p}} = \frac{4}{n} cut(\mathcal{V}_1, \mathcal{V}_2).$$

Dokaz: Vrijedi $\mathbf{p}^T\mathbf{p} = n$. Iz *6.* tvrdnje prethodnog teorema slijedi

$$\mathbf{p}^T\mathbf{L}\mathbf{p} = \sum_{\{i,j\}\in E} E_{ij}(p_i - p_j)^2.$$

Stoga, bridovi koji se nalaze unutar neke od particija $\mathcal{V}_1, \mathcal{V}_2$ ne pridonose prethodnoj sumi budući da će zbog definicije vektora $\mathbf{p}$ biti $p_i = p_j$. Bridovi koji povezuju element iz $\mathcal{V}_1$ s elementom iz $\mathcal{V}_2$ doprinose sumi s $(1+1)^2 E_{ij} = 4E_{ij}$.

□

Prisjetimo se da je cilj s početka priče bio minimizirati $cut(\mathcal{V}_1, \mathcal{V}_2)$, a iz prethodnog teorema lako vidimo da se to postiže za vektor $\mathbf{p}$ takav da su svi p_i jednaki 1 ili -1. Sama minimizacija reza može proizvesti skupove koji su malih veličina u odnosu na ostatak particije. Stoga trebamo funkciju cilja koja će uz minimalni rez, uzeti u obzir i potrebu za "balansiranim" particijama. Pretpostavimo prvo da smo svakom vrhu pridružili neku težinu $w(i)$ i neka je matrica $\mathbf{W}$ dijagonalna $n \times n$ matrica sastavljena od tih težina. Za podskup vrhova $\mathcal{V}_l$ definiramo težinu kao $w(\mathcal{V}_l) = \sum_{i\in\mathcal{V}_l} w(i) = \sum_{i\in\mathcal{V}_l} W_{ii}$. Reći ćemo da je 2-particija "balansirana" ako su težine podskupova približno jednake pa uzimamo sljedeću funkciju kao funkciju cilja koju želimo minimizirati.

$$\mathcal{Q}(\mathcal{V}_1, \mathcal{V}_2) = \frac{cut(\mathcal{V}_1, \mathcal{V}_2)}{w(\mathcal{V}_1)} + \frac{cut(\mathcal{V}_1, \mathcal{V}_2)}{w(\mathcal{V}_2)} \tag{2}$$

Za dvije različite particije s istom vrijednosti reza, prethodna funkcija je manja za onu particiju koja je više "balansirana"[1].

Lema 4 *Neka su* $\mathbf{L}$ *i* $\mathbf{W}$ *Laplaceova i matrica težina vrhova nekog grafa G. Tada generalizirani vektor particije* $\mathbf{q}$ *definiran s*

$$q_i := \begin{cases} +\sqrt{\dfrac{\eta_2}{\eta_1}} & i \in \mathcal{V}_1 \\ -\sqrt{\dfrac{\eta_1}{\eta_2}} & i \in \mathcal{V}_2. \end{cases}$$

zadovoljava $\mathbf{q}^T\mathbf{W}\mathbf{e} = 0$ *i* $\mathbf{q}^T\mathbf{W}\mathbf{q} = w(\mathcal{V})$. *Gdje je* $\eta_1 = w(\mathcal{V}_1)$ *i* $\eta_2 = w(\mathcal{V}_2)$.

Dokaz: Neka je $\mathbf{y} = \mathbf{W}\mathbf{e}$. Tada je $y_i = w(i) = W_{ii}$. Slijedi

$$\mathbf{q}^T\mathbf{W}\mathbf{e} = \sqrt{\frac{\eta_2}{\eta_1}} \sum_{i\in\mathcal{V}_1} w(i) - \sqrt{\frac{\eta_1}{\eta_2}} \sum_{i\in\mathcal{V}_2} w(i) = 0.$$

Slično je i $\mathbf{q}^T\mathbf{W}\mathbf{q} = \sum_{i=1}^{n} W_{ii}q_i^2 = \eta_1 + \eta_2 = w(\mathcal{V})$.

□

[1]Zanemarimo na trenutak značenje riječi "balansirana"

Teorem 5 *Koristeći notaciju prethodne leme, vrijedi*

$$\frac{\mathbf{q}^T\mathbf{L}\mathbf{q}}{\mathbf{q}^T\mathbf{W}\mathbf{q}} = \frac{cut(\mathcal{V}_1,\mathcal{V}_2)}{w(\mathcal{V}_1)} + \frac{cut(\mathcal{V}_1,\mathcal{V}_2)}{w(\mathcal{V}_2)}$$

Dokaz: Vektor $\mathbf{q}$ možemo zapisati kao

$$\mathbf{q} = \frac{\eta_1+\eta_2}{2\sqrt{\eta_1\eta_2}}\,\mathbf{p} + \frac{\eta_1-\eta_2}{2\sqrt{\eta_1\eta_2}}\,\mathbf{e}$$

Iz tvrdnje 7. teorema 2 imamo

$$\mathbf{q}^T\mathbf{L}\mathbf{q} = \frac{(\eta_1+\eta_2)^2}{4\eta_1\eta_2}\mathbf{p}^T\mathbf{L}\mathbf{p}.$$

Uvrštavanjem izraza za $\mathbf{p}^T\mathbf{L}\mathbf{p}$ i $\mathbf{q}^T\mathbf{W}\mathbf{q}$ iz teorema 3 i leme 4, laganim računom dolazimo do tvrdnje teorema.

□

Vidimo da je desna strana izraza u prethodnom teoremu upravo funkcija koju želimo minimizirati (2) pa pažnju sada možemo usmjeriti na lijevu stranu, odnosno traženje "optimalnog" generaliziranog vektora particije $\mathbf{q}$ uz uvjete dane lemom 4. Sljedeći teorem omogućuje da diskretni problem nalaženja optimalnog generaliziranog vektora $\mathbf{q}$ zamijenimo kontinuiranim, odnosno prebacivanjem u realnu domenu možemo aproksimirati traženi vektor.

Teorem 6 *Minimum problema*

$$\min_{\mathbf{q}\neq 0} \frac{\mathbf{q}^T\mathbf{L}\mathbf{q}}{\mathbf{q}^T\mathbf{W}\mathbf{q}},\ \textit{uz uvjet}\ \mathbf{q}^T\mathbf{W}\mathbf{e} = 0$$

se postiže za svojstveni vektor druge najmanje svjostvene vrijednosti generaliziranog svojstvenog problema

$$\mathbf{L}\mathbf{z} = \lambda\mathbf{W}\mathbf{z}. \tag{3}$$

Dokaz: Iz napomene nakon iskaza teorema 2 imamo da je $(0,\mathbf{e})$ je jedan svojstveni par generaliziranog svojstvenog problema uz najmanju svojstvenu vrijednost pa uz uvjet $\mathbf{q}^T\mathbf{W}\mathbf{e} = 0$ tvrdnja slijedi iz Courant Ficherovog teorema o spektru simetrične matrice (v.[3]).

□

Treba još spomenuti problematiku odabira težina vrhova koje smo koristili u prethodnim računima. Jedan izbor može biti $w(i) = 1$ za svaki vrh i. Time funkcija cilja iz (2) ima sljedeći oblik

$$\mathcal{Q}(\mathcal{V}_1,\mathcal{V}_2) = \frac{cut(\mathcal{V}_1,\mathcal{V}_2)}{|\mathcal{V}_1|} + \frac{cut(\mathcal{V}_1,\mathcal{V}_2)}{|\mathcal{V}_2|}.$$

Vidimo da ćemo takivm odabirom težina za vrhove dobiti 2-particije koje će biti "balansirane" u smislu veličina. Budući da pokušavamo grupirati dokumente i riječi po sličnosti, više smisla ima težinu vrha mjeriti pomoću težina koje "izlaze" iz njega budući da one sadrže informaciju o jačini veze dokumenta i riječi, tj. stavit ćemo $w(i) = \sum_k E_{ik}$. Takvim pristupom dolazimo do kriterija normaliziranog reza. Primijetimo da će u tom slučaju matrica $\mathbf{W}$ svojstvenog problema (3) biti jednaka matrici $\mathbf{D}$ iz teorema 2.

4 Veza s dekompozicijom singularnih vrijednosti

U prethodnom odjeljku vidjeli smo da nam drugi svojstveni vektor generaliziranog svojstvenog problema $\mathbf{Lz} = \lambda\mathbf{Dz}$ nudi relaksaciju diskretnog optimizacijskog problema traženja minimalnog normaliziranog reza. U ovom odjeljku predstavljamo algoritme za klasteriranje riječi i dokumenata koristeći model bipartitnog grafa. Račun provodimo za nalaženje 2-particije, a zatim algoritam generaliziramo za k-particiju. U slučaju bipartitnog grafa:

$$\mathbf{L} = \begin{bmatrix} \mathbf{D_1} & -\mathbf{A} \\ -\mathbf{A^T} & \mathbf{D_2} \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} \mathbf{D_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{D_2} \end{bmatrix}$$

gdje su $\mathbf{D_1}$ i $\mathbf{D_2}$ dijagonalne matrice t.d. $D_1(i,i) = \sum_j A_{i,j}$, $D_2(j,j) = \sum_i A_{i,j}$. Stoga se $\mathbf{Lz} = \lambda\mathbf{Dz}$ može zapisati kao

$$\begin{bmatrix} \mathbf{D_1} & -\mathbf{A} \\ -\mathbf{A^T} & \mathbf{D_2} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{D_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{D_2} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \tag{4}$$

Pretpostavimo li da su $\mathbf{D_1}$ i $\mathbf{D_2}$ regularne, jednadžbe možemo zapisati kao

$$\mathbf{D_1^{1/2}x} - \mathbf{D_1^{-1/2}Ay} = \lambda\mathbf{D_1^{1/2}x},$$

$$-\mathbf{D_2^{-1/2}A^Tx} + \mathbf{D_2^{1/2}y} = \lambda\mathbf{D_2^{1/2}y}.$$

Neka su $\mathbf{u} = \mathbf{D_1^{1/2}x}$ i $\mathbf{v} = \mathbf{D_2^{1/2}y}$. Uvrstimo li ih u gornje jednadžbe, dobijemo

$$\mathbf{D_1^{-1/2}AD_2^{-1/2}v} = (\mathbf{1} - \lambda)\mathbf{u},$$

$$\mathbf{D_2^{-1/2}A^TD_1^{-1/2}u} = (\mathbf{1} - \lambda)\mathbf{v},$$

Ove jednadžbe definiraju dekompoziciju singularnih vrijednosti (SVD) normalizirane matrice $\mathbf{A_n} = \mathbf{D_1^{-1/2}AD_2^{-1/2}}$. Posebno, $\mathbf{u}$ i $\mathbf{v}$ su redom lijevi i desni singularni vektori, dok je (1 - λ) pripadna singularna vrijednost. Stoga, umjesto računanja svojstvenog vektora druge (najmanje) svojstvene vrijednosti iz (4), možemo izračunati lijevi i desni singularni vektor koji pripadaju drugoj (najvećoj) singularnoj vrijednosti od $\mathbf{A_n}$,

$$\mathbf{A_n v_2} = \sigma_2\mathbf{u_2}, \quad \mathbf{A_n^T u_2} = \sigma_2\mathbf{v_2} \tag{5}$$

gdje je $\sigma_2 = 1 - \lambda_2$. Numerički je lakše raditi s $w \times d$ matricom $\mathbf{A_n}$ nego s $(w+d)\times(w+d)$ matricom $\mathbf{L}$. Desni singularni vektor $\mathbf{v_2}$ dat će biparticiju dokumenata, dok će lijevi singularni vektor $\mathbf{u_2}$ dati biparticiju riječi. Proučimo li relacije u (5), jasno je da se ovo rješenje slaže s našom intuicijom da particija dokumenata inducira particiju riječi, dok particija riječi implicira particiju dokumenata.

5 Algoritam

Iz prethodne priče zaključujemo da singularni vektori $\mathbf{u}_2$ i $\mathbf{v}_2$ "nose" informaciju o aproksimaciji optimalne particije, ali još ostaje otvoreno pitanje kako čitati te podatke. Drugi svojstveni vektor matrice $\mathbf{L}$ je dan s

$$\mathbf{z}_2 = \begin{bmatrix} \mathbf{D_1}^{-1/2}\mathbf{u}_2 \\ \mathbf{D_2}^{-1/2}\mathbf{v}_2 \end{bmatrix}. \tag{6}$$

Budući da je cilj 2-particija, tražimo bi-modalne vrijednosti m_1 i m_2 koje ćemo pridružiti riječima i dokumentima kako bismo direktno čitali rješenje. Jedan od pristupa je tražiti m_j takve da minimiziraju funkciju

$$\sum_{j=1}^{2} \sum_{\mathbf{z}_2(i)\in m_j} (\mathbf{z}_2(i) - m_j)^2.$$

Prethodni izraz je upravo funkcija cilja koju minimizira klasični k-means algoritam pa je naš postupak sljedeći:

Algoritam biparticije

1. Za dani $\mathbf{A}$, izračunati $\mathbf{A}_n = \mathbf{D}_1^{-1/2}\mathbf{A}\mathbf{D}_2^{-1/2}$,
2. Izračunati druge singularne vektore $\mathbf{u}_2$ i $\mathbf{v}_2$ matrice $\mathbf{A}_n$ i definirati $\mathbf{z}_2$ kao u (6),
3. Provesti *k-means* algoritam na jednodimenzionalnim podacima $\mathbf{z}_2$.

Prethodni postupak možemo generalizirati na traženje k klastera riječi i dokumenata. Iskoristimo $l = \lceil \log_2 k \rceil$ singularnih vektora $\mathbf{u}_2, \ldots, \mathbf{u}_{l+1}$ i $\mathbf{v}_2, \ldots, \mathbf{v}_{l+1}$ i definiramo

$$\mathbf{Z} = \begin{bmatrix} \mathbf{D_1}^{-1/2}\mathbf{U} \\ \mathbf{D_2}^{-1/2}\mathbf{V} \end{bmatrix}. \tag{7}$$

gdje je $\mathbf{U} = [\mathbf{u}_2, \ldots, \mathbf{u}_{l+1}]$ i $\mathbf{V} = [\mathbf{v}_2, \ldots, \mathbf{v}_{l+1}]$. Tražimo l-dimenzionalne točke $m_j,\ j = 1, \ldots, k$ koje ćemo pridružiti dokumentima i riječima. Stoga, mnimiziramo funkciju

$$\sum_{j=1}^{k} \sum_{\mathbf{Z}(i)\in m_j} ||\mathbf{Z}(i) - \mathbf{m}_j||^2,$$

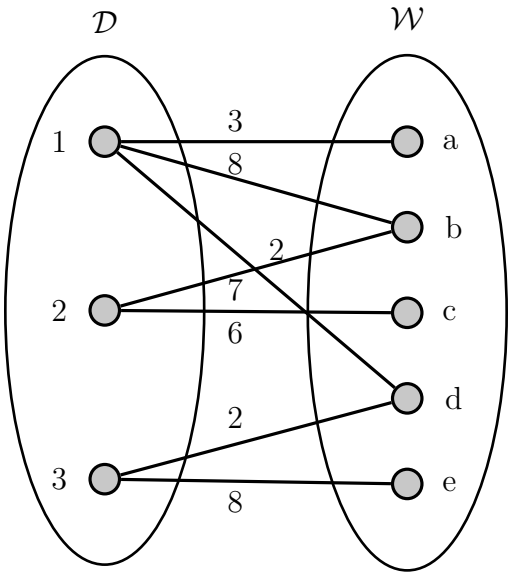
gdje je redak $\mathbf{Z}(i),\ i = 1, \ldots, d + w$ dokument ili riječ.

Algoritam k-particije

1. Za dani $\mathbf{A}$, izračunati $\mathbf{A}_n = \mathbf{D}_1^{-1/2}\mathbf{A}\mathbf{D}_2^{-1/2}$,
2. Izračunati $l = \lceil \log_2 k \rceil$ singularnih vektora $\mathbf{u}_2, \ldots, \mathbf{u}_{l+1}$ i $\mathbf{v}_2, \ldots, \mathbf{v}_{l+1}$ matrice $\mathbf{A}_n$ i definirati matricu $\mathbf{Z}$ kao u (7),
3. Provesti *k-means* algoritam na l-dimenzionalnim podacima $\mathbf{Z}$.

6 Testni primjer

Pretpostavimo da je zadan sljedeći težinski, bipartitni graf.

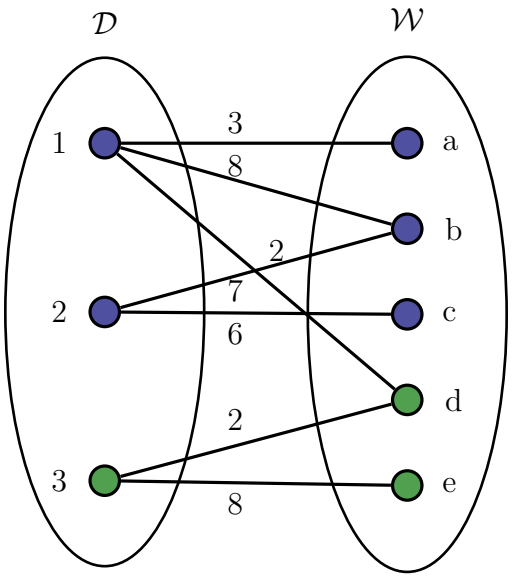


Matrice koje su uključene u prethodni račun dane su sa:

$$\mathbf{A} = \begin{bmatrix} 3 & 0 & 0 \\ 8 & 7 & 0 \\ 0 & 6 & 0 \\ 2 & 0 & 2 \\ 0 & 0 & 8 \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{bmatrix},$$

$$\mathbf{L} = \begin{bmatrix} \mathbf{3} & 0 & 0 & 0 & 0 & -3 & 0 & 0 \\ 0 & \mathbf{15} & 0 & 0 & 0 & -8 & -7 & 0 \\ 0 & 0 & \mathbf{6} & 0 & 0 & 0 & -6 & 0 \\ 0 & 0 & 0 & \mathbf{4} & 0 & -2 & 0 & -2 \\ 0 & 0 & 0 & 0 & \mathbf{8} & 0 & 0 & -8 \\ -3 & -8 & 0 & -2 & 0 & \mathbf{13} & 0 & 0 \\ 0 & -7 & -6 & 0 & 0 & 0 & \mathbf{13} & 0 \\ 0 & 0 & 0 & -2 & -8 & 0 & 0 & \mathbf{10} \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{bmatrix}.$$

Rezultat koji daje algoritam je $\mathcal{D}_0 = \{1, 2\},\ \mathcal{D}_1 = \{3\}$, a korespondirajući klasteri riječi su $\mathcal{W}_0 = \{a, b, c\},\ \mathcal{W}_1 = \{d, e\}$.



7 Eksperimentalni rezultati

Budući da ćemo pokušati testirati algoritam na velikom skupu dokumenata, treba prvo reći i nešto o dohvaćanju i obradi podataka što je teorija sama za sebe i postoji opsežna literatura i na tu temu. Ovdje samo navodimo postupak koji je korišten u ovom slučaju budući da on ovisi i o setu podataka s kojim raspolažemo.

7.1 Obrada podataka

Manji skupovi dokumenata pripremljni su manualno, a veći skupovi preuzeti su iz "TMDB 5000 Movie" (www.kaggle.com/tmdb/tmdb-movie-metadata) i "BBC News Summary" (www.kaggle.com/pariza/bbc-news-summary). Tekstovi su obrađeni koristeći MATLAB-ov Text Analytics Toolbox: tekstove smo pretvorile u niz riječi (tokenizedDocument); odstranile smo najčešće riječi (removeStopWords); izbrisale smo interpunkcijske znakove (removePunctuation); izvadile smo korijene riječi (normalizeWords).

Primjer obrađenog dokumenta:
childhood hour other seen other saw bring passion common spring same sourc taken sorrow awaken heart joi same tone lovd lovd alon childhood dawn ...

Dokumenti poput potonjeg su reprezentirani *bag-of-words* modelom, a pritom je zanemaren poredak riječi i sačuvan podatak o broju pojavljivanja svake riječi. Primjerice, iz modela ćemo znati da se u gornjem tekstu "childhood"

javlja 2 puta, ali ne znamo da se "hour" javlja prije "spring". Dobivene podatke ćemo analizirati mjerom *term frequency-inverse document frequency.*

Term frequency-inverse document frequency je mjera koja određuje težinu riječi u skupu dokumenata. Jedna varijanta formule za riječ r, dokument d i skup dokumenata D je dana sljedećim formulama:

$$tf(r,d) = f_{r,d}\ , \quad idf(r,D) = \ln \frac{N}{n_r}$$

$$tfidf(r,d,D) = tf(r,d) \times idf(r,D)$$

gdje je $f_{r,d}$ = "broj ponavljanja r u d", $N = card(D)$ i $n_r = \{d \in D : r \in d\}$. Naime, ovom formulom "težina" riječi u dokumentu raste s brojem pojavljivanja u tom dokumentu, a opada s brojem dokumenata u kojima se pojavljuje. Ideja je da "kaznimo" riječ ako se ponaša kao "stop word" i prioritiziramo ju ako se ponaša kao ključna riječ.

Slijedi par pokusa na različitim skupovima podataka, od lakših reprezentativnijh pema težima kojima ćemo "izazvati" algoritam.

7.2 Pokusi

Primjer 1

Koristile smo baze dokumenata s receptima, opisima lijekova i političkim vijestima. Budući da je svaki skup dokumenata bogat karakterističnim ključnim riječima, dobile smo potpuno preciznu particiju (Tablica 1). Ovim primjerom se prvenstveno provjerila "točnost" obrade podataka.

	recepti	*lijekovi*	*politika*
$\mathcal{D}_0$:	50	0	0
$\mathcal{D}_1$:	0	50	0
$\mathcal{D}_2$:	0	0	50
$\mathcal{W}_0$: minut, add, cook, heat, salt			
$\mathcal{W}_1$: medicin, take, doctor, effect, side			
$\mathcal{W}_2$: mr, govern, blair, labour, ministr			

Tablica 1: Primjer idealne particije

Primjer 2

Skup podataka "BBC News" dijeli sažetke vijesti u 5 kategorija: *sport, entertainment, politics, business, tech.* U ovom primjeru smo koristile skupove s najmanje preklapanja u pojmovima. Konačni rezultat nije savršen, ali odudaranje od idealne particije je jako maleno te se "teme" klastera dokumenata lako odrede preko pripadnih klastera riječi.

	sport	*business*	*tech*
$\mathcal{D}_0$:	1	48	1
$\mathcal{D}_1$:	49	0	1
$\mathcal{D}_2$:	0	2	48
$\mathcal{W}_0$: year, growth, new, compani			
$\mathcal{W}_1$: olymp, world, athlet, athen			
$\mathcal{W}_2$: mobil, peopl, gadget, user			

Tablica 2: Klasterima lako odredimo temu, što je indikator dobre particije

Primjer 3, 4 i 5

Broj dokumentata algoritmu obično nije predstavljao problem. Kad smo konstruirale skupine pjesama s jasnom temom (božićne, mjuzikl *Mačke*), particija bi bila precizna bez obzira na broj pjesama. Kasnije smo pridodale skup pjesama E.A.Poe-a, koje dijele ključne riječi međusobno, ali i s ostalim pjesmama. Algoritam je stoga dokumente tog skupa nasumično grupirao ili zajedno, ili skupa s ostalim pjesmama. Možemo zaključiti da u slučajevima poput toga program nema dovoljno informacija da uspješno grupira dokumente i riječi. Kako je i vidljivo u razlici tablica 4 i 5, dokumenti koji su jednako povezani s nekim klasterom znaju "šetati" između klastera. Razlike u tablicama su dobivene samo ponovnim pokretanjem programa. Unatoč tome, ovaj problem nam i ne smeta previše budući da se očekuje da "dobar algoritam" grupiranja dobro radi na velikom skupu podataka pa eventualne prednosti i zaostatke radije tražimo u sljedećim primjerima.

	Christmas	*E.A.Poe*	*Cats*
$\mathcal{D}_0$:	3	0	0
$\mathcal{D}_1$:	0	3	0
$\mathcal{D}_2$:	0	0	2
$\mathcal{W}_0$: snow, let, christma, merri, dai			
$\mathcal{W}_1$: sea, love, annabel, lee, kingdom			
$\mathcal{W}_2$: macav, cat, rum, tum, curious			

Tablica 3: Particija pjesama (točna)

	Christmas	*E.A.Poe*	*Cats*
$\mathcal{D}_0$:	3	1	0
$\mathcal{D}_1$:	0	2	0
$\mathcal{D}_2$:	0	0	2
$\mathcal{W}_0$: snow, let, christma, merri, dai			
$\mathcal{W}_1$: sea, love, annabel, lee, kingdom			
$\mathcal{W}_2$: macav, cat, rum, tum, curious			

Tablica 4: Particija pjesama (s greškom)

	Christmas	*E.A.Poe*	*Cats*
$\mathcal{D}_0$:	3	2	0
$\mathcal{D}_1$:	0	1	0
$\mathcal{D}_2$:	0	0	2
$\mathcal{W}_0$: snow, christma, love, sea, annabel			
$\mathcal{W}_1$: childhood, other, same, lovd, hour			
$\mathcal{W}_2$: macav, cat, rum, tum, curious			

Tablica 5: Particija pjesama (s greškom)

Primjer 6 i 7

U ovom primjeru smo koristile 500 članaka o sportu i 400 o tehnologiji. Nakon biparticioniranja su *Tech* dokumenti grupirani skupa, dok su *Sport* dokumenti rascjepkani. Zanimljivo je što smo povećanjem broja klastera na 4 dobile bolji rezultat, što se vidi u Tablici 7. Ovim primjerom dolazi do izražaja sama obrada podataka i primjedba s početka da taj proces predstavlja teoriju za sebe i ovisi o skupu podataka budući da smo dobili relativno zadovoljavajuće grupirane dokumente u klasterima $\mathcal{D}_1$ i $\mathcal{D}_2$, a sadržaj klastera $\mathcal{W}_0$ i $\mathcal{W}_3$ nije "progutao" ključne riječi vezane uz sport i tehnologiju.

	sport	*tech*
$\mathcal{D}_0$:	172	400
$\mathcal{D}_1$:	328	0
$\mathcal{W}_0$: olymp, world, athlet, athen		
$\mathcal{W}_1$: year, new, mobil, peopl		

Tablica 6: *Tech* dokumenti su odlično raspoređeni, dok su *Sport* dokumenti rascjepkani

	sport	*tech*
$\mathcal{D}_0$:	4	0
$\mathcal{D}_1$:	23	400
$\mathcal{D}_2$:	471	0
$\mathcal{D}_3$:	2	0
$\mathcal{W}_0$: harrier, ac, stade, treviso		
$\mathcal{W}_1$: peopl, game, mobil, phon		
$\mathcal{W}_2$: game, win, plai, against		
$\mathcal{W}_3$: republ, ireland, franc, faro		

Tablica 7: Uvođenjem dodatnih klastera, dobile smo točniju particiju originalnih dokumenata

Primjer 8

Koristeći primjere iz svih kategorija vijesti, nismo uspjele dobiti dobru particiju. Kao razlog naslućujemo preklapanje u ključnim riječima između većine kategorija. Particiju smo pokušale poboljšati povećanjem broja dokumenata, korištenjem dužih verzija dokumenata (podsjetnik: u prethodnim primjerima su korišteni sažeci vijesti) te variranjem broja klastera, no nismo uspjele povećati točnost.

	sport	*entertainment*	*politics*	*business*	*tech*
$\mathcal{D}_0$:	4	19	0	0	1
$\mathcal{D}_1$:	0	1	0	0	0
$\mathcal{D}_2$:	0	2	0	0	0
$\mathcal{D}_3$:	45	23	48	50	49
$\mathcal{D}_4$:	1	5	2	0	0
$\mathcal{W}_0$: best, award, film, book, year, winner					
$\mathcal{W}_1$: la, fenic, viotti, opera, director, includ					
$\mathcal{W}_2$: christian, andersen, han, prize, author, booker					
$\mathcal{W}_3$: mr, year, new, govern, peopl, world					
$\mathcal{W}_4$: film, famili, border, ballet, white, year					

Tablica 8: Primjer loše particije

Literatura

[1] Inderjit S. Dhillon, *Co-clustering documents and words using Bipartite Spectral Graph Partitioning, 2001.*

[2] Zlatko Drmač *Uvod u složeno pretraživanje podataka, predavanja 2020-2021*

[3] Zlatko Drmač *Numerička analiza 1, predavanja 2020-2021*

[4] 'Text Analytics Toolbox' dokumentacija *www.mathworks.com*