Klasteriranje višestrukih grafova

Marija Nikolić, Maja Soldo, Anamarija Sršen

5.12.2024.

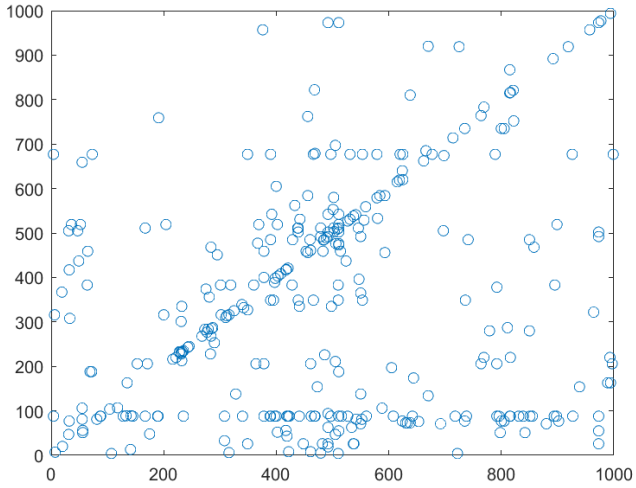
1 Uvod

Klasteriranje višestrukih grafova je grupiranje podataka koji su međusobno povezani na više načina. Na primjer, u društvenim mrežama, korisnici mogu biti povezani na različite načine, poput izravne komunikacije, dijeljenja sadržaja ili zajedničkih interesa. Cilj klasteriranja je pronaći skupine vrhova (korisnika) koje imaju slične uzorke povezanosti u svim analiziranim mrežama. Metoda koju ćemo promatrati za ovaj zadatak je Linked Matrix Factorization (LMF), koja omogućuje istovremenu analizu više različitih vrsta povezanosti između vrhova. Ova metoda omogućuje identifikaciju zajedničkih i specifičnih struktura unutar podataka te na temelju toga formira klastere.

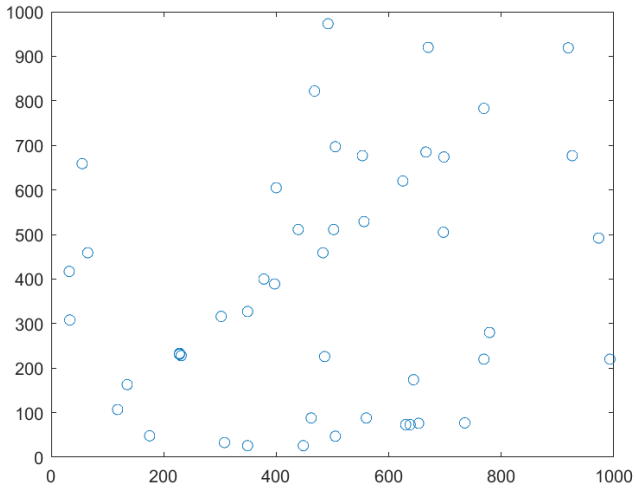
Primjer: Razmotrimo komunikaciju između korisnika na Twitteru. Oni mogu međusobno komunicirati na tri različita načina: spominjanje korisnika (mention), dijeljenje objava korisnika (retweet), odgovaranje na objave (reply). Analizom ovih oblika interakcije želimo grupirati korisnike u klastere. Na primjer, jedna grupa korisnika može predstavljati aktivne suradnike koji često odgovaraju jedni drugima, dok druga grupa može predstavljati pasivne promatrače koji isključivo dijele objave bez direktne komunikacije.

2 Podaci

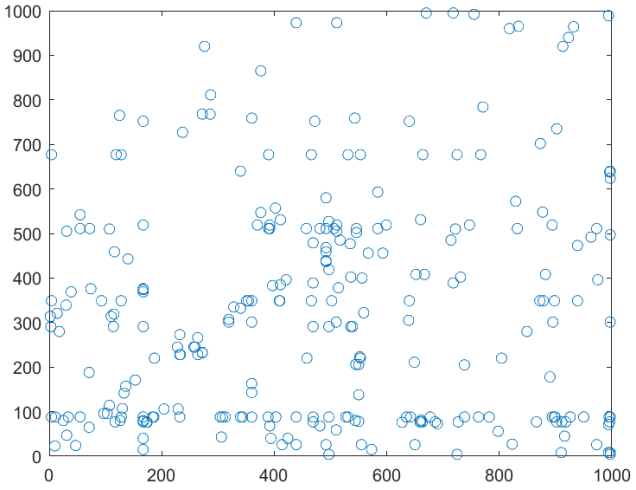
U ovom radu promatrat ćemo komunikaciju između 1000 korisnika na Twitteru. Imamo tri kategorije komunikacije, spominjanje drugog korisnika u objavi ili komentaru, dijeljenje objave drugog korisnika, i odgovaranje na objavu ili komentar drugog korisnika. Želimo vidjeti možemo li grupirati korisnike na temelju ovih triju kategorija komunikacije. Prvo smo sa spy funkcijom prikazale matricu spominjanje, matricu dijeljenje i matricu odgovaranje. U svakoj matrici element $a_{i,j}$ označava koliko je puta $i-$ti korisnik spomenuo, odgovorio ili podijelio objavu, komentar $j-$tog korisnika. Na slikama 1, 2 i 3 vidimo redom grafički prikaz podataka matrice spominjanja, matrice dijeljenja i matrice odgovaranja. Radimo klasteriranje za sve tri matrice podataka (Listing 1) i dobivamo grafičke prikaze na slikama 4, 5 i 6. Na temelju dobivenih slika, možemo uvidjeti grupiranje.



Slika 1: Grafički prikaz nenul podataka spominjanja drugog korisnika.



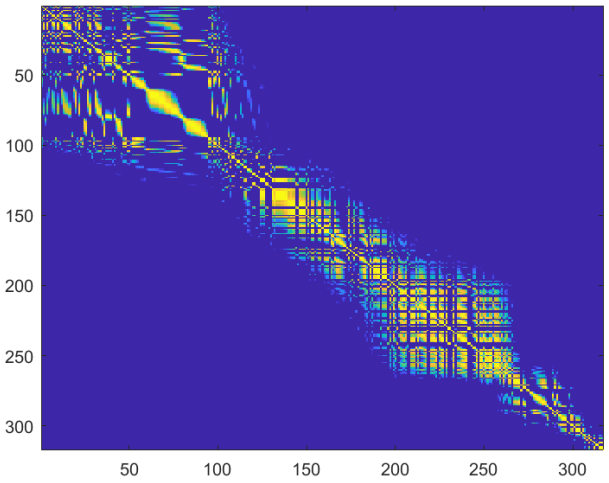
Slika 2: Grafički prikaz nenul podataka dijeljenja objave drugog korisnika.



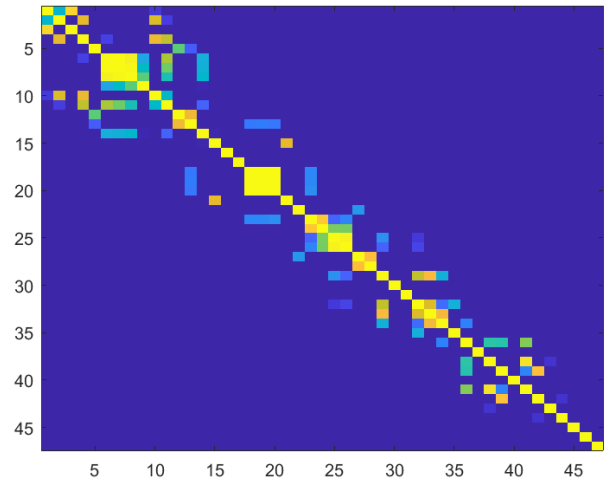
Slika 3: Grafički prikaz nenul podataka odgovaranje na objavu ili komentar drugog korisnika.

Listing 1: MATLAB kod za klasteriranje

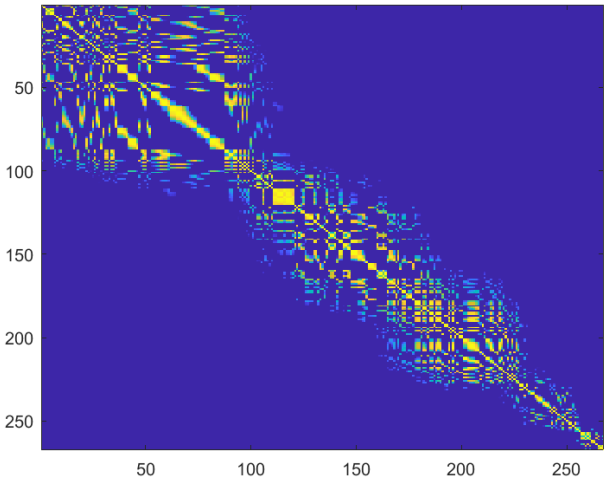
```
V=zeros(nnz_retweet,2);
for i=1:nnz_retweet
    V(i, 1)=rows_retweet(i);
    V(i, 2)=cols_retweet(i);
end
%plot (V(:, 1), V(:,2), 'o');
W=NaN(nnz_retweet, nnz_retweet);
e=exp(1);
s=4;
for i=1:nnz_retweet
    for j=i:nnz_retweet
        W(i,j)=e^((-(V(i,1)-V(j,1))^2-(V(i,2)-V(j,2))^2)/(2*s^2)); %s
        W(j,i)=W(i,j);
    end
end
imagesc(log(W));
```



Slika 4: Grafički prikaz klasteriranja podataka o spominjanju drugog korisnika.



Slika 5: Grafički prikaz klasteriranja podataka o dijeljenju objave drugog korisnika.



Slika 6: Grafički prikaz klasteriranja podataka o odgovaranje na objavu ili komentar drugog korisnika.

3 Metode

3.1 Linked Matrix Factorizaiton

3.1.1 Uvod

Neke od metoda za klasteriranje višestrukih grafova su zbroj grafova, zbroj spektralnih jezgri, konsenzus klasteriranje, linked matrix factorization. U metodi Linked Matrix Factorization želimo pronaći zajednički faktor svih grafova. Matricu grafa želimo aproksimirati matricom nižeg ranga $A \approx P\Lambda P^T$, gdje je P matrica reda $n \times d$, a Λ je simetrična matrica reda $d \times d$. S obzirom da radimo s više grafova kojima imaju zajedničke osnovne entitete, želimo pronaći matica koja je zajednički faktor svih matrica. Stoga je cilj minimizirati funkciju

$$G(P, \Lambda) = \frac{1}{2} \sum_{m=1}^{M} ||A^{(m)} - P\Lambda^{(m)} P^T||_F^2 + \frac{\alpha}{2} \left(\sum_{m=1}^{M} ||\Lambda^{(m)}||_F^2 + ||P||_F^2 \right),$$

gdje je P zajednički faktor svih grafova, $\Lambda^{(m)}$ je karakteristični faktor grafa, α je parametar regularizacije i $A^{(m)}$ je neusmjereni simetrični graf.

3.1.2 Optimizacija i algoritam

Budući da ciljna funkcija nije zajednički konveksna u P i $\Lambda^{(m)}$, koristimo izmjeničnu minimizaciju za pronalaženje lokalno optimalnog rješenja. Postupak se odvija kroz sljedeće korake:

1. Optimizira se matrica P, dok su $\Lambda^{(m)}$ fiksirane.
2. Zatim se optimiziraju matrice $\Lambda^{(m)}$, dok je P fiksirana.

3. Ovaj se proces ponavlja dok algoritam ne konvergira.

Za optimizaciju P i $\Lambda^{(m)}$ koristi se metoda kvazi-Newtona, točnije ograničena memorijska BFGS metoda (L-BFGS).

3.1.3 Gradijent ciljne funkcije

Bottleneck algoritma L-BFGS je procjena ciljne funkcije G i njezinog gradijenta prema P i $\Lambda^{(m)}$. Gradijenti se izračunavaju na sljedeći način:

$$\frac{\partial G}{\partial P} = -2\sum_{m=1}^{M}\big(A^{(m)} - P\Lambda^{(m)}P^\top\big)P\Lambda^{(m)} + \alpha P, \tag{1}$$

$$\frac{\partial G}{\partial \Lambda^{(m)}} = -P^\top\big(A^{(m)} - P\Lambda^{(m)}P^\top\big)P + \alpha\Lambda^{(m)}. \tag{2}$$

3.1.4 Računska složenost

Prilikom implementacije iskorištava se rijetkost matrica $A^{(m)}$. Ciljna funkcija može se reformulirati kao:

$$G' = \frac{1}{2}\sum_{m=1}^{M}\|A^{(m)}\|_F^2 - 2\mathrm{Tr}\big(\Lambda^{(m)}P^\top A^{(m)}P\big) + \mathrm{Tr}\big((P^\top P\Lambda^{(m)})^2\big), \tag{3}$$

gdje se prvi član računa u vremenskoj složenosti $O\big(d(\mathrm{nnz} + Nd)\big)$ za svaki graf. Ovdje nnz označava broj nenultih elemenata u matrici, dok d predstavlja dimenzionalnost podataka.

Na sličan način, gradijent se računa u istoj vremenskoj složenosti, što omogućuje učinkovitu implementaciju algoritma.

3.1.5 Zaključak

Algoritam Linked Matrix Factorization omogućuje istovremenu analizu više povezanih grafova, ali zahtijeva pažljivo podešavanje parametara i korištenje optimizacijskih metoda kao što je L-BFGS kako bi se postigla računska učinkovitost.

3.2 Težinski graf i klasteriranje

3.2.1 Uvod

Neka je $G(V, E)$ težinski graf gdje je V skup vrhova, a E skup bridova. Svaki brid $e \in E$ je uređena trojka (u, v, w), koja znači da su vrhovi u i v povezani s težinom w. Zamijenimo $w \in \mathbb{R}$ s $\overrightarrow{w} \in \mathbb{R}^k$, pri čemu je k broj bridova između u i v. Zatim konstruiramo funkciju f koja višestruki brid $\overrightarrow{w} \in \mathbb{R}^k$ prevodi u realan broj $f(\overrightarrow{w}) = \omega$. Funkcija f većinom je linearna $\omega = f(\overrightarrow{w}) = \sum \alpha w$.

3.2.2 Varijacija informacija u klasterima, metoda za kvantifikaciju udaljenosti između dva klasteriranja

Neka su $C_0 = \langle C_0^1, C_0^2, \dots C_0^K \rangle$ i $C_1 = \langle C_1^1, C_1^2, \dots C_1^K \rangle$ dva klasteriranja nad istim skupom čvorova. C_i je i-to klasteriranje, a C_i^j označava j-ti klaster unutar i-tog klasteriranja. Neka je n ukupan broj čvorova, tada je $\mathbb{P}(C,k) = \frac{|C^k|}{n}$ vjerojatnost da je neki čvor u klasteru C^k. Slično, $\mathbb{P}(C_i, C_j, k, l) = \frac{|C_i^k \cap C_j^l|}{n}$ je vjerojatnost da je neki čvor u klasteru C_i^k u klasteriranju C_i i u klasteru C_j^l u klasteriranju C_j.

Sada je funkcija stanja ili očekivana vrijednost naučenih informacija u C_i definirana kao

$$H(C_i) = -\sum_{k=1}^{K} \mathbb{P}(C_i, k) \log\left(\mathbb{P}(C_i, k)\right),$$

gdje je K broj klastera u C_i.

Međusobna informacija dijeljena između C_i i C_j definirana je na sljedeći način:

$$I(C_i, C_j) = \sum_{k=1}^{K_i} \sum_{l=1}^{K_j} \mathbb{P}(C_i, C_j, k, l) \left(\log \mathbb{P}(C_i, C_j, k, l)\right).$$

S obzirom na te dvije veličine, definiramo metričku varijaciju informacija kao

$$d_{VI}(C_i, C_j) = H(C_i) + H(C_j) - 2I(C_i, C_j),$$

gdje $H(C_i)$ označava prosječnu nesigurnost pozicije čvora u klasteriranju C_i. Ako nam je poznato C_i i C_j, onda $I(C_i, C_j)$ označava prosječno smanjenje nesigurnosti o položaju čvora u C_i. Kada su dva klasteriranja identična, $d_{VI}(C_i, C_j)$ je nula, a maksimalna je kada su dva klasteriranja potpuno neovisna.

3.2.3 Rekonstrukcija grafa na temelju poznatog klasteriranja

Neka nam je dano klasteriranje za graf s višestrukim bridovima, pitamo se možemo li konstruirati shemu agregacije koja najbolje odgovara danim podacima. Ova shema agregacije reducira višestruka mjerenja sličnosti (višestruki težinski bridovi) u jedno mjerenje sličnosti (jedan težinski brid).

Formalno, imamo višestruki graf $G(V, E)$ s višestrukim mjerenjima sličnosti za svaki brid $\langle w_i^1, w_i^2, \dots, w_i^K \rangle \in \mathbb{R}^K$ i klasteriranje C^* za taj graf G. Cilj nam je pronaći vektor težina $\alpha \in \mathbb{R}^K; \omega_i = \sum \alpha_i w_i$, tako da je C^* optimalno klasteriranje za graf G. Za to imamo dva pristupa.

Prvi pristup: Rješavanje inverznog problema Inverzni problemi javljaju se u mnogim aplikacijama, gdje je cilj izvesti nevidljive parametre iz ograničenih opažanja. Rješenja obično uključuju iteracije predviđanja i rješavanje forward problema (problemi kojima je zadatak predvidjeti izlaz sustava na temelju poznatih ulaznih podataka i modela), kako bi se izračunala točnost predviđanja. Naš problem može se smatrati inverznim problemom, jer pokušavamo izračunati funkciju agregacije na temelju danog klasteriranja. Nedostatak ove metode je što se oslanja na točnost algoritma klasteriranja.

Drugi pristup: Maksimiziranje kvalitete klasteriranja Pronalazak funkcije agregacije koja maksimizira kvalitetu klasteriranja. Imamo dva cilja pri izračunavanju α: 1. Opravdati lokaciju svakog vrha. 2. Maksimizirati ukupnu kvalitetu klasteriranja.

3.2.4 Opravdanje lokacija svakog vrha:

Za svaki vrh definiramo privlačnost (pull) prema svakom klasteru C^k u klasteriranju $C = \langle C^1, C^2, \ldots, C^K \rangle$ kao kumulativnu težinu bridova između v i njegovih susjeda u C^k,

$$\mathbb{P}_\alpha(v, C^k) = \sum_{w_i=(u,v)\in E, u\in C^k} w_i(\alpha),$$

gdje $w_i(\alpha)$ označava težinu brida nakon agregacije prema vektoru težina α.

Holding power $H_\alpha(v)$ za svaki vrh definiran je kao razlika između privlačnosti klastera kojem vrh pripada u C^* i sljedeće najveće privlačnosti među preostalim klasterima. Ako je ova razlika pozitivna, tada je vrh čvršće vezan za svoj odgovarajući klaster nego za bilo koji drugi. Maksimiziranjem broja vrhova s pozitivnim $H_\alpha(v)$, možemo povećati broj ispravno grupiranih vrhova $|\{v; H_\alpha(v) > 0\}|$.

3.2.5 Maksimiziranje ukupne kvalitete klasteriranja:

Bilo koja metrička mjera kvalitete klasteriranja može se potencijalno koristiti za ovu svrhu. Otkrili smo da neke strogo linearne funkcije imaju trivijalno rješenje. Razmotrimo funkciju cilja koja mjeri kvalitetu klasteriranja kao zbroj bridova unutar klastera. Kako bismo minimizirali kumulativne težine bridova koji presijecaju klastere (odnosno, maksimizirali težine bridova unutar klastera), rješavamo:

$$\min_{\alpha,|\alpha|=1} \sum_{e_j\in Cut} \sum_{i=1}^{K} \alpha_i w_j^i,$$

gdje je Cut skup bridova koji su krajnji čvorovi u različitim klasterima. Nadalje, označimo s S^k zbroj težina bridova koji povezuju neki vrh s vrhom iz skupa Cut,

$$S^k = \sum_{e_j\in Cut} w_j^k,$$

Tada se funkcija cilja može zapisati kao

$$\min_{\alpha} \sum_{k=1}^{K} \alpha_k S^k,$$

Budući da je ova funkcija linearna, ona ima trivijalno rješenje koje dodjeljuje težinu 1 metričkoj vrijednosti s najvećim S^k, što znači da se uzima u obzir samo jedna metrika sličnosti.

3.3 Pronalaženje latentnih klastera

3.3.1 Uvod

Razmatramo situaciju u kojoj nekoliko vrsta veza dijeli redundantne informacije, ali se kao cjelina kombiniraju kako bi oblikovale širu strukturu. Neki od primjera su znanstveni članci u časopisima koji mogu biti povezani prema sličnosti teksta, sličnosti sažetka,

ključnim riječima, zajedničkim autorima, međusobnim citiranjima itd. Redundantnost se često pojavljuje kod sličnosti teksta, sažetka i ključnih riječi kako bi prenijeli informacije o temi. S druge strane, zajednički autor vjerojatno prenosi informacije i o temi i o lokaciji, jer obično radimo s osobama iz istog područja te s onima iz obližnjih institucija.

Za atribute teme i lokacije kažemo da su latentni jer ne postoje eksplicitno u podacima. Kako bi smanjili broj informacija, vidimo da veći dio varijacija u podacima možemo predstaviti pomoću dva relativno neovisna klasteriranja temeljena na temi dokumenata i njihovoj lokaciji.

3.3.2 Uzorci prostora klasteriranja

Uzimamo točke u $\alpha_i \in \mathbb{R}^k$ tako da vrijedi $|\alpha_i| = 1$, i na svakoj točki izračunavamo odgovarajući graf i klasteriramo koristeći Graclusov algoritam. Možemo usporediti te klastere koristeći varijacije informacija.

3.3.3 Meta-klasteri: klasteri klasteriranja

Iako je zanimljivo otkriti značajno različita klasteriranja, zbog nedostatka stabilne strukture klasteriranja to nije korisno za primjene poput nenadziranog učenja. Potrebno je dodatno smanjiti ovaj skup klasteriranja. Ovaj problem rješavamo primjenom ideje klasteriranja na sam skup klasteriranja. Ovaj problem nazivamo problemom meta-klasteriranja.

Predstavljamo klasteriranja kao čvorove u grafu i povezujemo ih bridovima čije su težine određene obrnuto proporcionalno metričkoj varijaciji informacija. Klasteriramo graf klasteriranja kako bismo vidjeli postoje li unutar šireg prostora čvrsto povezani klasteri klasteriranja.

Rezultati ovise o konkretnoj instanci problema. Iako ne tvrdimo da se meta-klasteri uvijek mogu pronaći, očekujemo da će postojati u mnogim višetežinskim grafovima, a iskorištavanje strukture meta-klasteriranja može omogućiti učinkovito upravljanje ovim prostorom, što je tema sljedećeg odjeljka.

3.3.4 Prosječno klasteriranje unutar klastera

Prosječno klasteriranje može se izvršiti prema algoritmu CSPA (Cluster-based Similarity Partitioning Algorithm). Svako klasteriranje prikazano je kao blok-dijagonalni graf u kojem su dva čvora povezana ako i samo ako pripadaju istom klasteru. Zatim se formira graf dobiven sumiranjem ostalih grafova. Taj novi graf se zatim klasterira korištenjem tradicionalnog algoritma, a dobiveno klasteriranje vraća se kao reprezentativno klasteriranje.

3.3.5 Redoslijed prema sadržaju informacija skupa

Originalna 3x3 struktura može se rekonstruirati koristeći samo prva dva reprezentativna klasteriranja. Zašto su ta dva klasteriranja odabrana prva? Odabir trećeg i četvrtog reprezentativnog klasteriranja ne bi dao tako ugodan rezultat. Kako bismo trebali odrediti redoslijed skupa reprezentativnih klasteriranja?

Skup reprezentativnih klasteriranja možemo procijeniti prema nekoliko čimbenika:

1. Koliki udio naših uzoraka pripada povezanim meta-klasterima, odnosno koliki dio prostora klasteriranja pokrivaju?

2. Koliko informacija klasteriranja pokrivaju kao skup?

3. Koliko su redundantna klasteriranja? Koliko se informacija preklapa?

Želimo maksimizirati količinu informacija uz minimiziranje redundancije.

3.3.6 Izračunavanje sadržaja informacija u skupu klasteriranja

Međusobna informacija dvaju klasteriranja, C_1 i C_2, prema formuli:

$$I(C_1, C_2) = \sum_{i=1}^{K_1} \sum_{l=1}^{K_2} \mathbb{P}(C_1, C_2, k, l) \log \mathbb{P}(C_1, C_2, k, l)$$

gdje je $\mathbb{P}()$ vjerojatnost da je nasumično odabrani čvor u navedenim klasterima.

Ovo je ekvivalentno vlastitoj informaciji kartezijskog produkta dvaju klasteriranja. Proširenje za skup od n klastera, $I(C_1, C_2, \ldots, C_n)$, daje:

$$I(C_1, C_2, \ldots, C_n) = \sum_{a=1}^{K_1} \sum_{b=1}^{K_2} \cdots \sum_{z=1}^{K_n} \mathbb{P}(C_1, C_2, \ldots, C_n, a, b, \ldots, z) \log \mathbb{P}(C_1, C_2, \ldots, C_n, a, b, \ldots, z)$$

3.3.7 Praktično rješavanje problema

Za veliki broj klasteriranja ili velike vrijednosti K, ovaj izračun brzo postaje nepraktičan. U tim slučajevima redoslijed klasteriranja tako što dodajemo novo klasteriranje u skup maksimizirajući minimalnu parnu udaljenost do svakog klasteriranja koje je već u skupu. Ovakav pristup, iako ne izbjegava preklapanje informacija između triju klasteriranja, pokazuje se učinkovitim u praksi.