Algoritam za prepoznavanje grupa vrhova u kompleksnoj mreži

Poljak, Denis

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Department of Mathematics / Sveučilište Josipa Jurja Strossmayera u Osijeku, Odjel za matematiku**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:126:234365

Sveučilište Josipa Jurja Strossmayera u Osijeku

Odjel za matematiku

Sveučilišni diplomski studij Matematika i računarstvo

Algoritam za prepoznavanje gustih grupa vrhova u kompleksnoj mreži

Diplomski rad

Mentor:

izv. prof. dr. sc.
Snježana Majstorović

Kandidat:

Denis Poljak

Osijek, 2022

Sadržaj

1 | Uvod

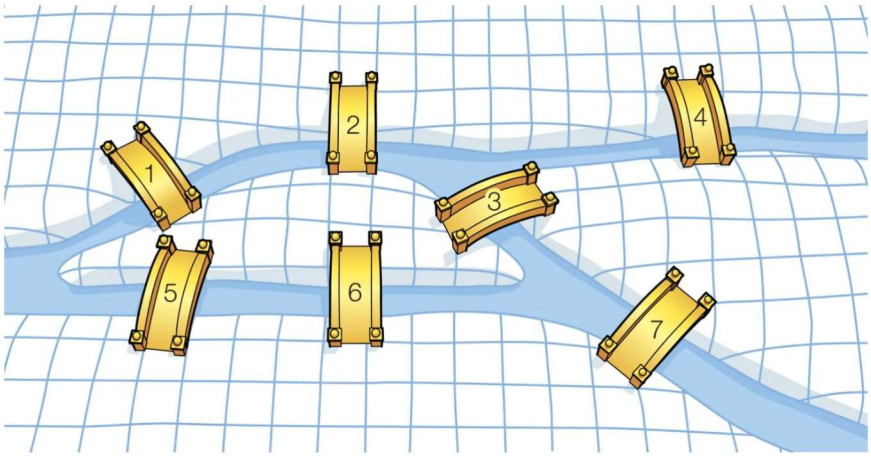
U sadašnjem digitalnom vremenu mreže su prikladan način predstavljanja podataka jer je većina sustava dobro povezana. Značajniji i napredniji način proučavanja mreža i interakcija među njima su grafovi. Analiza kompleksnih mreža ima važan istraživački značaj u primijenjenoj matematici, biologiji, poslovnoj analizi i mnogim drugim područjima. Cilj ovoga rada je objasniti osnovne pojmove teorije grafova, odnosno mreža te predstaviti detekciju zajednica u mrežama koristeći Louvain algoritam.

Louvain algoritam je brzi algoritam za otkrivanje zajednica s pouzdanim rezultatima. U ovom radu ćemo predstaviti Louvain algoritam, a zatim i njegovu poboljšanu verziju koja se zove Louvain Prune algoritam. Dokazano je da poboljšana verzija značajno smanjuje vrijeme izvršavanja algoritma i zadržava gotovo istu kvalitetu kao izvorni Louvain algoritam.

2 | Kompleksne mreže

2.1 Nastanak teorije kompleksnih mreža

Jedan od prvih matematičara koji je razmišljao o grafovima i mrežama bio je Leonhard Euler. Eulera je zaintrigirao stari problem u vezi grada Königsberga u blizini Baltičkog mora. Rijeka Pregel dijeli Königsberg na četiri odvojena dijela koji su povezani sa sedam mostova. Građani Königsberga su se pitali je li moguće obići sva područja grada prolazeći svakim od mostova točno jednom. Leonhard Euler je problem sedam Königsberških mostova riješio 1735. godine. Na postavljeno pitanje odgovorio je negativno.

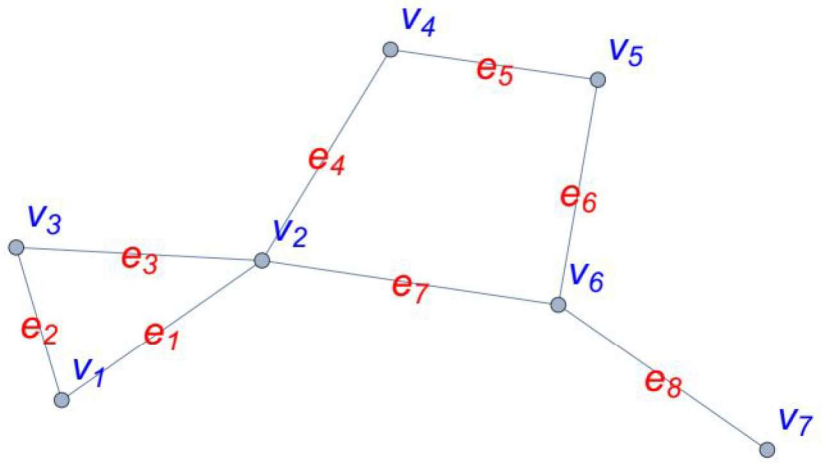


Slika 2.1: Mostovi u Königsbergu

Euler je pokazao da rješenje problema ovisi isključivo o načinu na koji mostovi povezuju različite dijelove grada, odnosno o topološkim svojstvima mreže. Eulerov znanstveni rad je prvi rad koji koristi koncepte i tehnike moderne teorije mreža pa je 1736. godine utemeljena nova grana matematike, teorija grafova. Danas pojam mreže postupno zamijenjuje pojam grafa pa možemo govoriti o teoriji mreža umjesto o teoriji grafova.

2.2 Neusmjerena kompleksna mreža

Definicija 1. *Neusmjerena kompleksna mreža G je uređena trojka $G = (V(G), E(G), \Psi_G)$ koja se sastoji od nepraznog skupa $V = V(G)$, čiji su elementi vrhovi od G, skupa $E = E(G)$ disjunktnog s $V(G)$, čiji su elementi bridovi od G i funkcije incidencije Ψ_G koja svakom bridu od G pridružuje neuređeni par (ne nužno različitih) vrhova od G.*

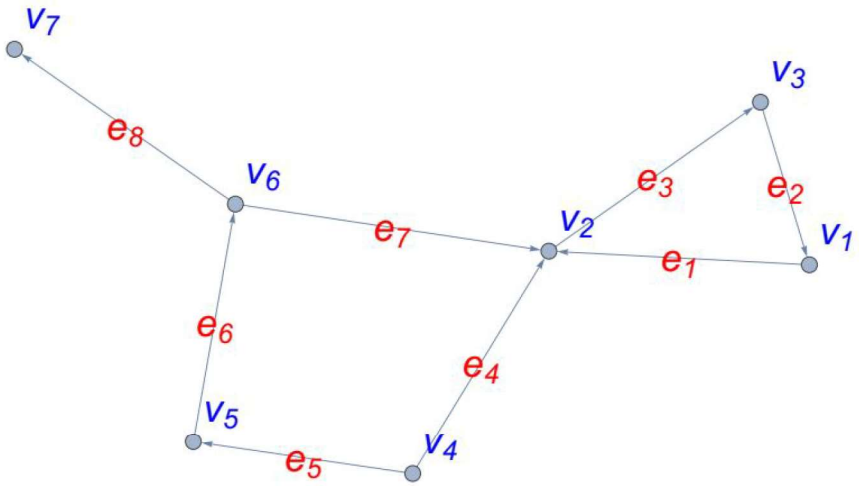


Slika 2.2: Primjer neusmjerene jednostavne mreže

Skup vrhova grafa sa slike 2.2 je $V = \{v_i : i = 1, 2, \ldots, 7\}$, dok je skup bridova $E = \{e_i : i = 1, 2, \ldots, 8\}$ te vrijedi $\psi(e_1) = \{v_1, v_2\}$, $\psi(e_2) = \{v_1, v_3\}$, $\psi(e_2) = \{v_2, v_3\}$ itd.

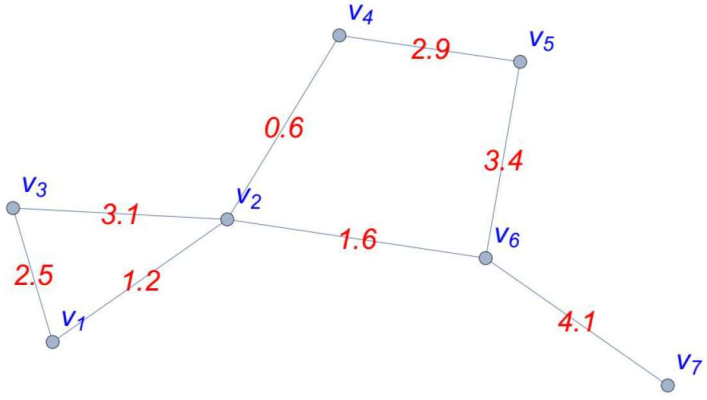
2.3 Usmjerena kompleksna mreža

Definicija 2. *Usmjerena kompleksna mreža D je uređena trojka $(V(D), A(D), \Psi_D)$ koja se sastoji od nepraznog skupa $V(D)$ vrhova, skupa $A(D)$ lukova (ili usmjerenih bridova) i funkcije incidencije Ψ_D koja svakom luku a pridružuje uređeni par (ne nužno različitih) vrhova u i v. Vrh u je početni, a v krajnji vrh luka a.*



Slika 2.3: Primjer usmjerene mreže.

Definicija 3. *Težinska mreža G^α je mreža nastala od grafa G tako da su joj bridovima pridruženi realni brojevi, odnosno postoji težinska funkcija $\alpha : E(G) \to \mathbb{R}$, pri čemu broj $\alpha(e)$ zovemo težinom brida $e \in E(G)$.*



Slika 2.4: Primjer težinske mreže.

Grafovi, odnosno mreže često se koriste za modeliranje različitih aspekata našeg života i okruženja iz stvarnog svijeta. Značajna istraživanja i evaluacije struktura mreža provedena su vezano za probleme u društvenim znanostima, biologiji, računarstvu i primijenjenoj matematici.

Razna područja znanosti identificirale su vlastite potrebe za razvojem metoda za analiziranjem i detektiranjem specijalnih struktura u mrežama. Problemi particioniranja grafova usko su povezani s brojnim realnim problemima i zato su razvijeni brojni algoritmi za otkrivanje strukture mreže i njeno particioniranje.

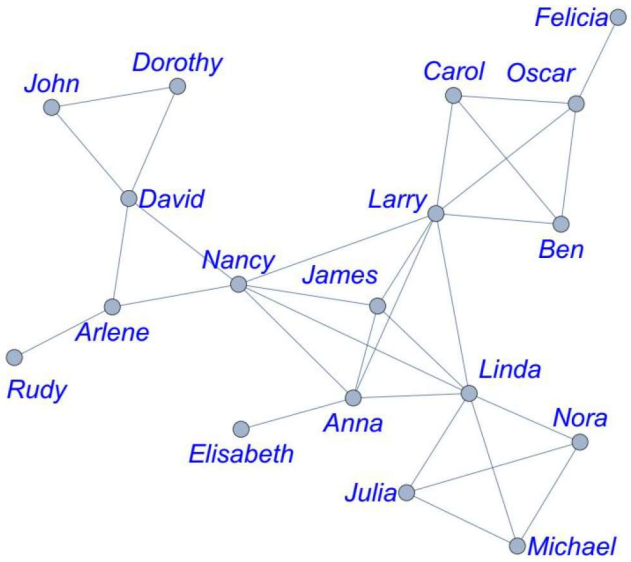
Navedimo neke primjere uporabe kompleksnih mreža.

- Društvene mreže.
  Korisnici se smatraju vrhovima, a ukoliko su neka dva korisnika prijatelji, tada su odgovarajući vrhovi spojeni bridom. Ovo je korisno u preporukama prijatelja.

- Preporuke za proizvode.
  Proizvodi su vrhovi mreže, a s obzirom na količinu kupljenih proizvoda mogu se stvoriti odnosi između proizvoda koji postaju bridovi grafa. Ti odnosi pokreću preporuku proizvoda za nove kupce.

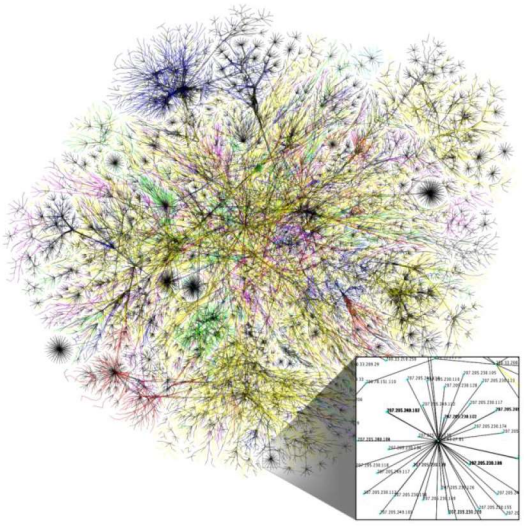
- Zemljopisne karte.
  Svaka aplikacija koja koristi karte koristi mreže (grafove) za računanje najkraćeg puta između dvije lokacije (vrhovi) i stvarne cestovne udaljenosti (bridovi).

Primjer 1. *U društvenim mrežama vrhovi su ljudi ili skupine ljudi, a bridovi socijalna interakcija između njih: prijateljstvo, rodbinski odnosi, poslovni odnosi itd.*



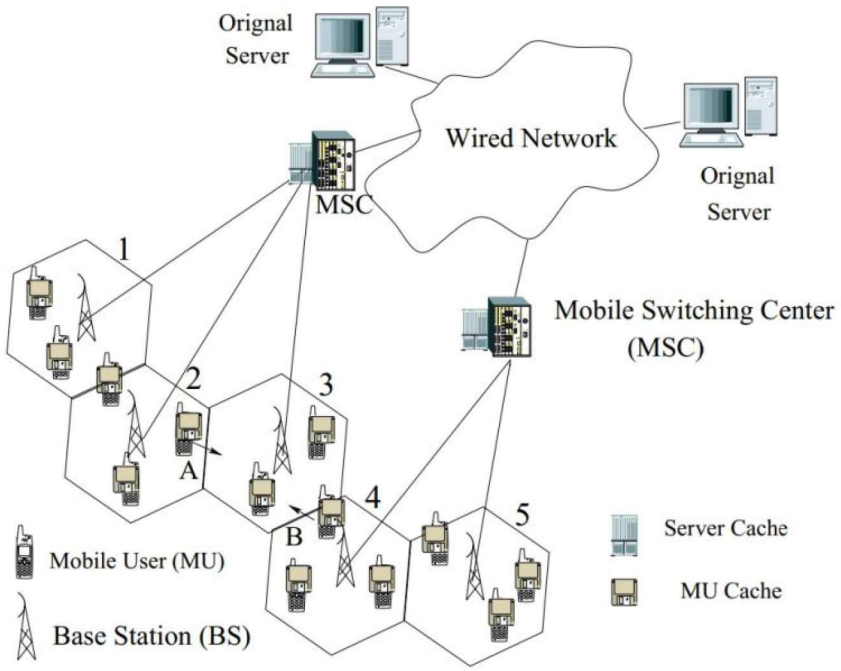
Slika 2.5: Mreža obiteljskog okupljanja.

Primjer 2. *Opte Project: vizualizacija ruterskih putova kroz dio Interneta.*



Slika 2.6: Ruterski putevi.

Primjer 3. *Tehnološke mreže: komunikacijska mreža.*



Slika 2.7: Bežična mreža.

3 | Osnovni pojmovi iz teorije mreža

U ovom poglavlju ćemo navesti osnovne pojmove iz teorije mreže koji su nužni za razumijevanje Louvani algoritma.

3.1 Osnovna svojstva mreža

Definiciju grafa, odnosno mreže smo naveli u poglavlju 2. Spomenimo da vrhove mreže najčešće označavamo s $v_1, v_2, \ldots, v_n$, a bridove s $e_1, e_2, \ldots, e_n$. Ako brid e_1 spaja vrhove v_1 i v_2, onda pišemo $e_1 = v_1v_2$ ili $e_1 = v_2v_1$. Za mrežu G kažemo da je konačna ako su $V(G)$ i $E(G)$ konačni skupovi.

Ako su vrhovi v_1 i v_2 mreže G spojeni s dva ili više bridova, onda kažemo da postoji višestruki brid između tih vrhova, a mrežu G tada zovemo *multimrežom*. Brid koji spaja vrh sa samim sobom zovemo *petljom*.
Za mrežu kažemo da je jednostavna ako su joj svaka dva vrha spojena s najviše jednim bridom i nema petlji.

Stupanj vrha v u mreži G je broj bridova koji su incidentni s vrhom v i označavamo ga s $d(v)$. Ako postoji petlja nad vrhom v, onda ju računamo kao dva brida. Vrh stupnja 0 je *izolirani vrh*, a vrh stupnja 1 je *list*.
Intuitivno, stupanj vrha u mreži je broj sjecišta male kružnice oko vrha s linijama koje izlaze iz njega.

- Najveći stupanj mreže G označavamo s $\Delta(G)$:

  $$\Delta(G) = \max_{v \in V(G)} d_G(v).$$

- Najmanji stupanj mreže G označavamo s $\delta(G)$:

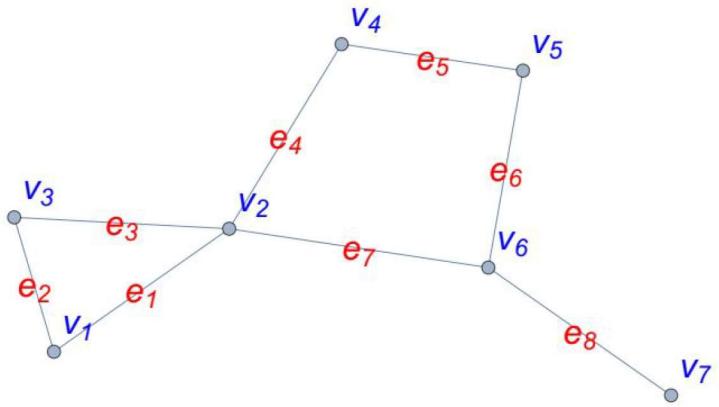
  $$\delta(G) = \min_{v \in V(G)} d_G(v).$$

- Prosječan stupanj mreže G označavamo s $d(G)$:

  $$d(G) = \frac{1}{|V(G)|} \sum_{v \in V(G)} d_G(v).$$

Jasno je da vrijedi $\delta(G) \leq d(G) \leq \Delta(G)$.

Za svaku mrežu G vrijedi $\sum_{v \in V(G)} d(v) = 2|E(G)|$.

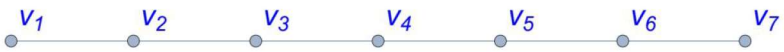


Slika 3.1: Stupnjevi vrhova v_i, $i = 1, \ldots, 7$ u mreži su redom 2,4,2,2,2,3,1.

Mreža sa samo jednim vrhom je trivijalna, u suprotnom je netrivijalna. Mreža G je prazna ako je $E(G) = \emptyset$. U praznoj mreži svaki vrh je izoliran, odnosno stupanj svakog vrha jednak je nuli.

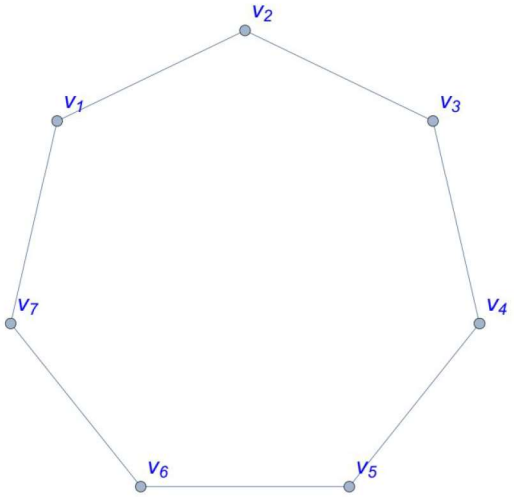
U mreži G mogu postojati neki disjunktni podskupovi A i B skupa vrhova V između kojih ne postoji niti jedan brid. Takva mreža je nepovezana, a njezini dijelovi sa skupovima vrhova A i B su komponente povezanosti mreže G.

Definicija 4. *Put P_n s n vrhova je jednostavna mreža sa skupom vrhova i bridova redom* $V(P_n) = \{v_1, v_2, \ldots, v_n\},\ E(P_n) = \{v_i v_{i+1} : i = 1, \ldots, n-1\}$.



Slika 3.2: Put P_7

Definicija 5. *Ciklus C_n s n vrhova je jednostavna mreža sa skupom vrhova i bridova redom* $V(C_n) = \{v_1, v_2, \ldots, v_n\},\ E(C_n) = \{v_1v_2, v_2v_3, \ldots, v_{n-1}v_n, v_n, v_1\}$.



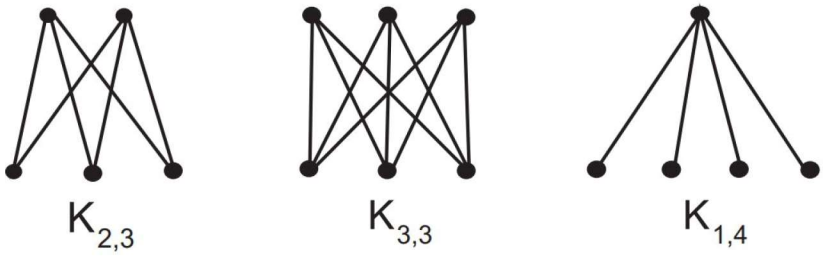
Slika 3.3: Ciklus C_7

Definicija 6. *Potpuna mreža K_n s n vrhova je jednostavna mreža u kojoj su svaka dva vrha spojena bridom.*

Definicija 7. *Mreža G je bipartitna (ili dvodijelna) ako joj se skup vrhova može particionirati u dva skupa X i Y tako da svaki brid ima jedan kraj u X, a drugi u Y. Particiju (X, Y) zovemo biparticijom mreže G. Bipartitna mreža s biparticijom (X, Y) označavamo s $G(X, Y)$.*

Definicija 8. *Potpuna bipartitna mreža je jednostavna bipartitna mreža s biparticijom (X, Y) u kojem je svaki vrh iz X spojen sa svakim vrhom iz Y. Ako je $|X| = m$ i $|Y| = n$, onda takvu mrežu označavamo s $K_{m,n}$.*

Bipartitne mreže obično opisuju pripadnost vrhova nekim grupama. Primjerice, jedan skup biparticije mogu činiti nogometni igrači, a drugi skup nogometni klubovi. Brid spaja igrača i klub ako je igrač igrao za taj klub (ili npr. glumci i filmovi u kojima su glumili).



Slika 3.4: Primjeri bipartitnih mreža.

3.2 Podmreže

Definicija 9. *Mreža H je podmreža od G, u oznaci $H \subseteq G$, ako je $V(H) \subseteq V(G)$, $E(H) \subseteq E(G)$, a $\Psi_H = \Psi_G|_{E(H)}$ (tj. Ψ_H je restrikcija od Ψ_G na $E(H)$.)*

- Ako je $H \subseteq G$ i $H \neq$G, pišemo$H \subset G$ i H zovemo pravom podmrežom od G.
- Ako je H podmreža od G, onda je G nadmreža od H.

Najjednostavniji tipovi podmreža od G su one koje su dobivene izbacivanjem jednog vrha ili jednog brida iz G.

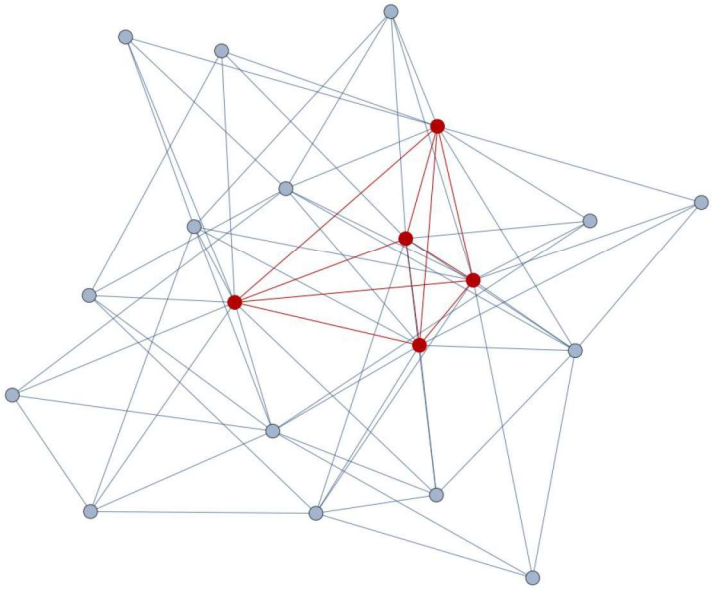
Definicija 10. *Podmreža H od G za koju je $V(G) = V(H)$ zove se podmreža koja razapinje G ili razapinjujuća podmreža od G.*

Posebni tipovi podmreža neke mreže su klike. Klika u mreži G je podmreža od G sa skupom vrhova $C \subseteq V(G)$, pri čemu su svaka dva vrha iz C spojena bridom.

Drugim riječima, klika je podmreža sa skupom vrhova $C \subseteq V(G)$ koja je potpuna. Maksimalna klika je klika koja nije sadržana u niti jednoj kliki sa većim brojem vrhova, tj. dodavanjem nekog vrha, ona prestaje biti klika. Općenito, ne postoji učinkovit algoritam za određivanje najvećih klika u mreži.

Identifikacija (najvećih) klika u mreži je važna jer često implicira postojanje klastera odnosno da je mreža heterogena. U kontestu društvenih mreža, klika implicira postojanje grupe ljudi s međusobno snažnim vezama (poslovnim, prijateljskim, itd.)

Primjer 4. *Na slici 3.5 prikazana je mreža s* 20 *vrhova. Crvenom bojom je istaknuta najveća klika* K_5.



Slika 3.5: Klika K_5.

3.3 Povezanost i udaljenost vrhova u mreži

Definicija 11. *Udaljenost $d_G(u,v)$ dvaju vrhova $u,v \in V(G)$ u mreži G je duljina najkraćeg $(u,v)-$puta u G. Ako takav put ne postoji, stavljamo $d_G(u,v) = \infty$.*

Mreža G je povezana ako $d_G(u,v) < \infty$, $\forall$ u,v $\in V(G)$. U suprotnom kažemo da je G nepovezana.

Svakoj mreži G možemo particionirati skup vrhova V na skupove $V_1, V_2, \ldots, V_k$ tako da su dva vrha $u,v \in V$ povezana u G ako i samo ako postoji $i \in \{1,\ldots,k\}$ takav da vrijedi $u,v \in V_i$.

Mreže sa skupovima vrhova $V_1, V_2, \ldots, V_k$ zovemo komponentama povezanosti od G. Povezane mreže imaju samo jednu komponentu povezanosti. Broj komponenata povezanosti u mreži označavamo s $c(G)$.

Komponenta povezanosti koja ima puno veći broj vrhova od ostalih komponenata zove se gigantska komponenta. Mnoge realne mreže nisu povezane, odnosno sadrže gigantsku komponentu.

Ukoliko je mreža G povezana, važno je ispitati koliko "jaka" je njena povezanost, tj. hoće li prestati biti povezana nakon što joj uklonimo samo jedan vrh, jedan brid ili više vrhova i bridova. S tim u vezi definiramo rezni vrh i rezni brid. Vrh v je rezni vrh mreže G ako vrijedi da je $c(G-v) > c(G)$, odnosno brid e je rezni brid ili most mreže G ako vrijedi $c(G-e) > c(G)$.

Vrijede sljedeće tvrdnje.

Teorem 1. *Neka je $|V(G)| \geq 3$. Vrh v mreže G je rezni vrh ako i samo ako postoje vrhovi u i w $(v \neq u,w)$ tako da v pripada svakom $(u,w)-$putu u G.*

Teorem 2. *Brid e mreže G je most ako i samo ako postoje vrhovi u i w takvi da e pripada svakom $(u,w)-$putu od G.*

Postoje mreže u kojima treba ukloniti više vrhova ili bridova kako bi se one raspale, odnosno prestale biti funkcionalne. Slijedom toga nužno je definirati vršni rez i bridni rez u mreži.

Definicija 12. *Vršni rez (separator, separacijski skup) mreže G je podskup* $V' \subseteq V(G)$ *tako da je* $G - V'$ *nepovezana ili trivijalna.* $K-$*vršni rez je vršni rez s k elemenata, tj.* $|V'| = k$.

Slično je definiran bridni rez i $k-$bridni rez od G.

U literaturi se osnovni problem particioniranja mreže zove MIN-CUT problem, a ideja je pronaći onaj bridni rez u mreži čija je suma težina bridova najmanja moguća.

3.4 Komponente povezanosti

Mnoge realne neusmjerene mreže nisu povezane, nego u njima postoji jedna velika komponenta povezanosti koja sadrži većinu ili nerijetko više od 90% vrhova (a time i jako puno bridova) dok se ostatak mreže sastoji od izuzetno malih komponenata povezanosti.

Postoje i mreže koje nemaju veliku komponentu povezanosti, ali one se niti ne koriste kao model nekog realnog problema jer to nije od interesa. Treba razlikovati veliku komponentu povezanosti od gigantske komponente povezanosti, a koja je od velikog interesa u proučavanju kompleksnih mreža.

Gigantska komponenta je komponenta povezanosti u mreži čiji broj vrhova raste proporcionalno s ukupnim brojem n vrhova u mreži. Glavni cilj u proučavanju gigantske komponente je procijeniti broj vrhova koji joj pripadaju.

Detekcija vrhova mreže sa "sličnim" svojstvima od velike je važnosti, posebice u velikim mrežama, gdje je ključno što ranije identificirati specifične strukture. Proces grupiranja ili particioniranja je postupak organiziranja vrhova u disjunktne grupe takve da su vrhovi unutar jedne grupe međusobno slični na temelju neke mjere sličnosti.

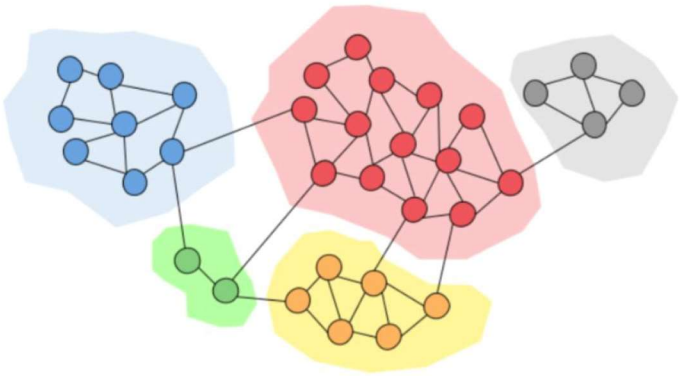
Mjera sličnosti se obično izračunava na temelju nekih topoloških kriterija, npr. strukture grafa ili na temelju lokacije vrhova u mreži itd. Vrhovi koji se smatraju sličnim grupiraju se u takozvane klastere. Drugim riječima, svaki klaster sastoji se od vrhova koji imaju zajednička svojstva.

Razlikujemo problem particioniranja grafa (eng. graph partitioning) i problem detektiranja zajednica (eng. community detection). Oba problema se odnose na podjelu vrhova mreže u izvjestan broj grupa uzimajući u obzir kriterij podjele koji se odnosi na bridove. Najčešće je cilj minimizirati broj (sumu težina) bridova koji spajaju vrhove iz različitih grupa.

4 | Particioniranje kompleksnih mreža

Particioniranje mreža je važno za otkrivanje i razumijevanje njene strukture. Neke mreže su homogene pa primjena bilo kakvog particijskog algoritma neće dati smislene rezultate. Smisleno je koristiti metode particioniranja na mrežama za koje postoji sumnja na heterogenost.

Problem detektiranja zajednica nastao je razvijanjem teorije kompleksnih mreža u svrhu razumijevanja strukture mreže, a odnosi se na podjelu ili particiju skupa vrhova grafa u izvjestan broj grupa tako da je broj (suma težina) bridova između grupa najmanja moguća. Broj grupa nije fiksan, a nije ni veličina grupa (broj vrhova). Problem detektiranja zajednica stoga nije jasno definiran, a cilj je napraviti prirodnu podjelu mreže u istaknute grupe vrhova. Neki algoritmi za detekciju zajednica neće uvijek napraviti podjelu mreže u grupe.



Slika 4.1: Podjela mreže.

Modularnost je veličina koja se odnosi na strukturu mreže, a mjeri snagu podjele mreže na izvjesne grupe (klastere ili zajednice). Mreže s visokom modularnošću imaju puno veći broj bridova između vrhova unutar grupe nego između vrhova koji se nalaze u različitim grupama.

Istraživanja su pokazala da modularnost ima ograničenje razlučivosti, tj. nije u mogućnosti otkriti male zajednice.

Homofilija ili asortativno uparivanje u kompleksnoj mreži je tendencija da se vrhovi spajaju bridovima s vrhovima koji su im slični po određenom kriteriju.
U društvenim mrežama često postoji tendencija da se ljudi povezuju sa sličnima

sebi prema npr. socijalnom statusu, rasi, dobi, spolu, nacionalnosti, istovrsnom poslu koji obavljaju, političkim stavovima itd. Također, u mrežama citiranosti znanstveni radovi citiraju radove iz sličnih područja znanosti. Web stranice napisane na nekom jeziku imaju tendenciju povezivanja sa stranicama koje su na istom jeziku. Stoga je razumno očekivati da u brojnim kompleksnim mrežama postoje istaknute grupe vrhova koji su slični prema određenim kriterijima.

Tijekom godina, predloženo je više algoritama za otkrivanje zajednica u kompleksnoj mreži i oni su najčešće temeljeni na modularnosti. Izvorni algoritmi za optimizaciju modularnosti koriste dendograme i zatim pronalaze rezni vrh za koji je modularnost maksimalna. Iz toga razloga algoritam je prespor za velike mreže jer mu je vremenska složenost $O(n^3)$, gdje je n broj vrhova u mreži.

Optimiziranjem modularnosti se često dobiju dobre grupe zajednica. No, problem grupiranja zajednica na temelju modularnosti je NP-potpun problem jer je potrebno računati najveću modularnost za svaki vrh u mreži.

Pohlepni algoritmi se uglavnom koriste za rješavanje problema matematičke optimizacije. Uglavnom se minimizira ili maksimizira funkcija troška. U našem slučaju radi se o funkciji modularnosti. Pohlepni algoritmi neće uvijek dati optimalno rješenj, ali ponekad mogu dati približno globalno rješenje danog problema u razumnom vremenu. Jedan od takvih algoritama je i Louvain algoritam.

Navedimo i heurističke algoritme koji se koriste za rješavanje NP problema. Oni smanjuju vremensku složenost problema davanjem brzih rješenja. Za očekivati je da neće ponuditi najbolje rješenje, ali će sigurno dati skoro optimalno rješenje u kratkom vremenu.

4.1 Koeficijent asortativnosti

Definicija 13. *Neka je G prozvoljna mreža sa skupom vrhova $V(G) = \{1, 2, \ldots, n\}$. Matrica susjedstva $A(G)$ mreže G je kvadratna $n \times n$ matrica, čiji je element a_{ij} jednak broju bridova između vrhova i i j.*

Neka je c_i svojstvo (tip) vrha i u mreži G, $c_i \in \{1, \ldots, n_c\}$, pri čemu je n_c ukupan broj svojstava koje mogu imati vrhovi mreže. Ukupan broj bridova koji spajaju vrhove istog tipa dan je formulom

$$\sum_{\{i,j\} \in E(G)} \delta(c_i, c_j) = \frac{1}{2} \sum_{i,j} a_{ij} \delta(c_i, c_j),$$

pri čemu je δ Kroneckerova delta funkcija.

Očekivani broj bridova između vrhova istog tipa jednak je

$$\frac{1}{2} \sum_{i,j} \frac{d_i d_j}{2m} \delta(c_i, c_j),$$

gdje je $m = |E(G)|$.

Slijedi da je razlika između postojećeg i očekivanog broja bridova koji spajaju vrhove istog tipa

$$\frac{1}{2} \sum_{i,j} a_{i,j} \delta(c_i, c_j) - \frac{1}{2} \sum_{i,j} \frac{d_i d_j}{2m} \delta(c_i, c_j) = \frac{1}{2} \sum_{i,j} \left(a_{i,j} - \frac{d_i d_j}{2m} \right) \delta(c_i, c_j).$$

Dijeljenjem te razlike s m dobivamo

$$Q = \frac{1}{2m} \sum_{i,j} \left(a_{i,j} - \frac{d_i d_j}{2m} \right) \delta(c_i, c_j).$$

Veličina Q je modularnost mreže i pomoću iste izražavamo mjeru tendencije da su vrhovi istog tipa spojeni bridovima.
Vrijedi $Q < 1$. Ako je $0 < Q < 1$, mreža je asortativna, a ako je $1 < Q < 0$, mreža je disasortativna. Ako je $Q = 0$, mreža se ponaša kao slučajna mreža. Kada je $Q > 0.3$, smatra se da mreža ima očitu strukturu zajednice. Što je bliže 1, to je struktura zajednica očitija.

U raznim problemima koristi se tzv. modularna matrica s elementima

$$B_{ij} = a_{ij} - \frac{d_i d_j}{2m}.$$

Kao što je vidljivo iz formule, modularnost je usmjerena samo na računanje razlike stvarnog i očekivanog broja bridova unutar svake zajednice. Kada se određeni vrh pomakne, promjena modularnosti događa se samo između dviju zajednica u kojima se vrhovi pomiču. Stoga je u tom slučaju potrebno računati samo prirast modularnosti i razliku, bez ukupne modularnosti prije i poslije pomicanja.

U sljedećem poglavlju ćemo objasniti Louvain algoritam temeljen na optimizaciji modularnosti.

5 | Louvain algoritam

U teoriji kompleksnih mreža od interesa je grupirati vrhove u klastere ili zajednice (eng. communities). Otkrivanjem tih zajednica uočavaju se razna svojstva mreža. Radi primjene u različitim područjima znanosti, predloženo je mnogo algoritama za otkrivanje zajednica. Algoritmi često koriste optimizaciju modularnosti, jednu od najpopularnijih metoda za otkrivanje zajednice. No, često su ti algoritmi prespori za velike mreže jer im je složenost $O(n^3)$, gdje je n broj vrhova mreže. Zato su preloženi aproksimacijski algoritmi, a jedan od njih je Louvain algoritam, čija je složenost $O(m)$, gdje je m broj bridova u mreži.

Louvain algoritam je pohlepni algoritam za maksimiziranje modularnosti i dobro je poznat kao jedan od najbržih i najučinkovitijih algoritama za otkrivanje zajednice. Ulazna mreža za algoritam je mreža $G = (V, E)$ sa skupom vrhova V i skupom bridova E. Otkrivanje zajednica se provodi tako da dijelimo G u zajednice $C = \{V_1, V_2, \ldots, V_n\}$. Pritom vrijedi da vrh mreže može pripadati samo jednoj zajednici, tj. $V_i \cap V_j = \emptyset$, $\forall i \neq j$.

Glavni koraci Louvainovog algoritma :

1. korak: Inicijalizacija zajednice i postavljanje svakog vrha kao zasebne zajednice.

2. korak: Pronalazak svih zajednica povezanih s vrhom vrh_1 i izračunavanje promjene modularnosti nakon premještanja vrh_1 u svaku susjednu zajednicu. Premještanje vrh_1 u zajednicu koje rezultira povećanjem modularnosti.

3. korak: Prolazak kroz sve vrhove i izvršenje 2. koraka dok se ne nađu vrhovi za premještanje.

4. korak: Spajanje svake zajednicu u 3. koraku u novi vrh koji se zove super vrh. Povratak na 1. korak dok se svi vrhovi ne spoje u jednu zajednicu.
   Rezultat je višerazinska particija zajednica, a particija s najvećom modularnošću odabire se kao konačni rezultat.

5.1 Particioniranje mreže

Algoritam počinje od inicijalizacije tako da svakom vrhu dodijeli različitu zajednicu. Odnosno svaki vrh je zasebna zajednica. Za svaki vrh i algoritam će:

- Izračunati promjenu modularnosti $\triangle Q$ kada vrh premještamo u zajednicu nekog susjednog vrha.
- Premjestiti vrh v_i u zajednicu kojoj pripada vrh v_j tako da je $\triangle Q$ najveća moguća.

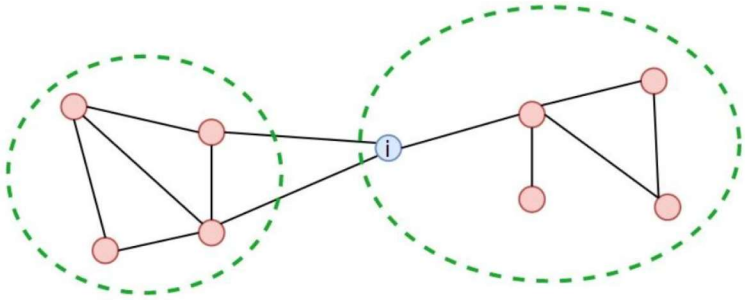
Particioniranje je prva faza algoritma. Prva faza prestaje kada se postigne lokalni maksimum modularnosti, odnosno kada se pomicanjem bilo kojeg vrha ne poboljšava modularnost. Primjetimo da rezultat algoritma ovisi o redoslijedu u kojem se vrhovi razmatraju. Istraživanja pokazuju da redoslijed vrhova nema značajan utjecaj na ukupnu modularnost. U prvoj fazi za svaki vrh u mreži računa se promjena modularnosti $\triangle Q$.

$$\triangle Q(i \to C) = \left[\frac{\sum_{in} + \sum_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]$$

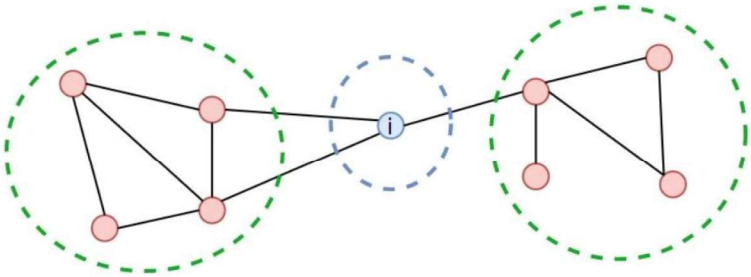
Gdje je:

- $\sum_{in}$ je zbroj stupnjeva vrhova unutar zajednice u kojem se nalazi vrh i ne računajući bridove koji nisu u toj zajednici.
- $\sum_{tot}$ je zbroj stupnjeva vrhova unutar zajednice.
- $k_{i,in}$ je zbroj težina bridova između vrha i i zajednice V_i.
- k_i je zbroj svih težina bridova s krajem u vrhu i.

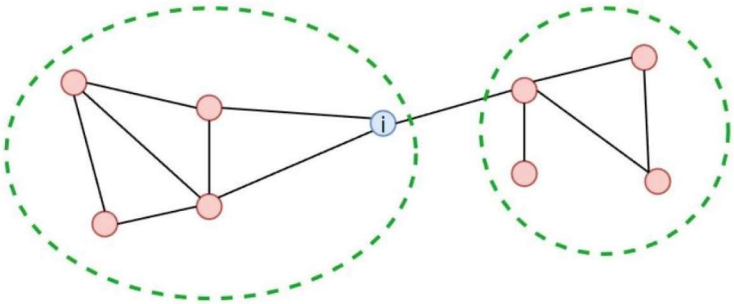
Nakon računanja promjene modularnosti vrh i pridružujemo zajednici C. Neophodno je definirati i kako će se iz neke zajednice D izbaciti vrh i i pridružiti ga drugoj zajednici C, odnosno $\triangle Q(D \to i \to C)$.
Slike u nastavku služe kao pojašnjenje postupka.



Slika 5.1: Inicijalno klasteriranje.

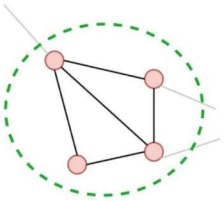


Slika 5.2: $\triangle Q(D \to i)$.

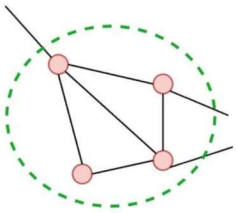


Slika 5.3: $\triangle Q(i \to C)$.

$$Q(C) = \frac{1}{2m} \sum_{i,j \in C} \left[a_{ij} - \frac{d_i d_j}{2m} \right] = \frac{\sum\limits_{i,j \in C} a_{ij}}{2m} - \frac{(\sum_{i \in C} d_i) - (\sum_{j \in C} d_j)}{2m} = \frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m} \right)^2.$$



Slika 5.4: $\sum_{in} = 10$



Slika 5.5: $\sum_{tot} = 13$

Objašnjenje modularnosti Q_{before}.

$$Q_{before} = Q(C) + Q(\{i\}) = \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2\right] + \left[0 - \left(\frac{k_i}{2m}\right)^2\right].$$

Računanje modularnosti nakon pridruživanja vrha i zajednici C.

$$Q_{after} = Q(C + \{i\}) = \left[\frac{\sum_{in} + k_{i,in}}{2m}\right] - \left[\left(\frac{\sum_{tot} + k_i}{2m}\right)^2\right].$$

U konačnici se dobiva

$$\triangle Q(i \to C) = Q_{after} - Q_{before}.$$

Slično se definira $\triangle Q(D \to i)$.

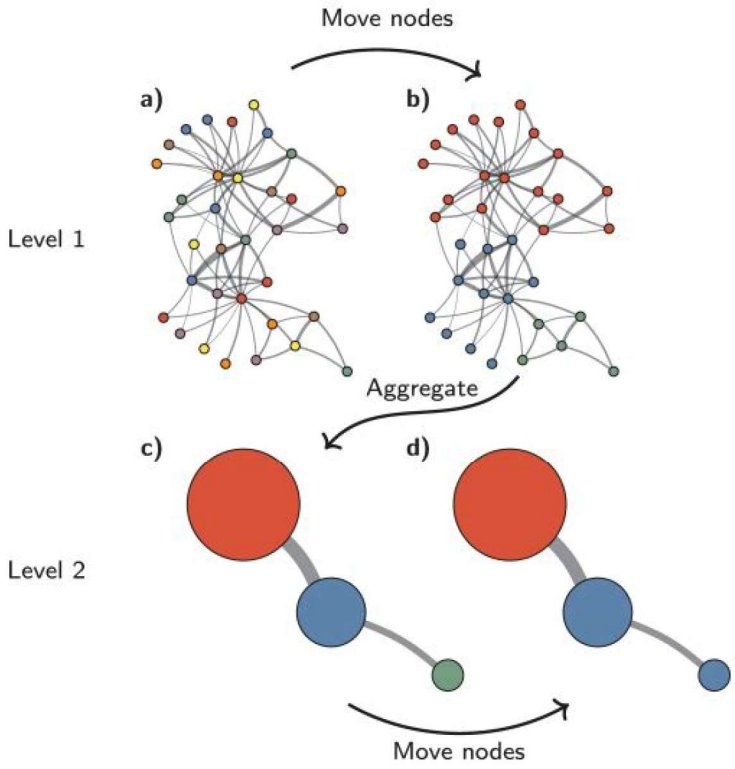
Za svaki vrh $i \in V$ koji se nalazi u nekoj zajednici C tražimo najbolju zajednicu C':

$$C' = \text{argmax}_{C'} \triangle Q(C \to i \to C').$$

Ako vrijedi da je $\triangle Q(C \to i \to C') > 0$, onda se zajednice ažuriraju tako da se od zajednice C oduzme vrh i te se doda zajednici C'.

5.2 Restrukturiranje mreže

Zajednice dobivene u prvoj fazi grupiraju se u "super-vrhove". Super vrhovi su povezani ako postoji barem jedan brid između vrhova odgovarajućih zajednica. Težina brida između dva super-vrha je zbroj težina svih bridova između njihovih zajednica. Prva faza se izvodi iznova na super- vrhovima mreže.



Slika 5.6: Louvain algoritam.

Pojasnimo sliku 5.6. Louvain algoritam počinje od particije u kojoj je svaki vrh u vlastitoj zajednici (a). Algoritam premješta pojedinačne vrhve iz jedne zajednice u drugu kako bi pronašao particiju (b). Na temelju ove particije stvara se agregatna mreža (c). Algoritam zatim pomiče pojedinačne vrhove u agregatnoj mreži (d). Ovi se koraci ponavljaju sve dok se ne može dobiti bolja particija.

5.3 Random Louvain algoritam

Primjetimo da Louvain algoritam troši veliku količinu vremena u prvoj fazi na prvoj iteraciji. Uzrok tome je što ponavlja prvu fazu sve dok nema poboljšanja u modularnosti, a računanje se izvodi tako da se računa $\triangle Q$ za sve susjedne vrhove svih vrhova u mreži. Stoga je potrebno optimizirati algoritam.

Random Neighbor Louvain algoritam je puno brža verzija Louvain algoritma (2 do 3 puta). Ideja na kojoj je zasnovan Random Neighbor algoritam je da se nasumično bira zajednica susjednih vrhova kojoj će neki vrh pripadati umjesto da se razmatraju svi susjedni vrhovi.

Neosporno je i vrlo vjerovatno smanjenje kvalitete algoritma, ali to u konačnici neće utjecati na kalitetu, budući da mnogi susjedi vjerovatno već pripadaju istoj zajednici u početnoj podjeli. To znači da, ako se i nasumično izabere susjedni vrh, on je u istoj zajednici s našim vrhom i to s velikom vjerojatnošću. Ovakav princip značajno smanjuje vrijeme računanja.

Istraživanja su pokazala da Louvain algoritam uvelike ovisi o svojstvima ulazne mreže. Najlošiji rezultati postižu se na mrežama koje imaju visok prosječan stupanj vrha te malen maksimalan stupanj vrha.

Identifikacijom vrhova koji će promijeniti svoju zajednicu u sljedećoj iteraciji prve faze i uzeti u obzir samo njih, vrijeme izračuna će se značajno smanjiti. Iako ih je nemoguće točno identificirati moguće jee smanjiti broj vrhova koji imaju potencijal za promjenu zajednice.
Postoje četiri grupe vrhova koji imaju takav potencijal.

1. Susjedni vrhovi vrha i koji nisu u klasteru Y.
2. Susjedni vrhovi vrha i koji su u klasteru Y.
3. Susjedni vrhovi od klastera X koji nemaju bridove s krajem u vrhu i.
4. Vrhovi od klastera Y koji nemaju bridove s krajem u vrhu i.

Vrhovi koji pripadaju nekoj od grupa će samo utjecati na modularnost. Vrijeme algoritma će se smanjiti, a kvaliteta zadržati.

5.4 Proposed Louvain Prone algoritam

Proposed Louvain Prone algoritam je predloženi algoritam koji smanjuje vrijeme računanja u usporedbi s izvornim algoritmom i zadržava kvalitetu. Vrijeme računanja će se smanjiti ako se modularnost Q računa samo za vrhove koji pripadaju nekoj od četiri grupe opisane gore. Za pronalaženje vrhova iz sve četiri grupe trebat će dosta vremena. Za ubrzanje algoritma razmatraju se samo vrhovi koji pripadaju prvoj grupi jer oni najviše utječu na $\triangle Q$.

U nastavku je prikazan pseudo kod Proposed Louvain Prone algoritma.

Algoritam 1 Louvain Prune Algorithm

```
1:  function LouvainPruneAlgorithm(Graph G)
2:      C the index of communities for each node of G
3:      P = V(G)
4:      while P ≠ ∅ do
5:          v = random node x ∈ P
6:          P = P - {v}
7:          best_q = −∞
8:          best_c = community of v
9:          for all neighboring nodes n of v do
10:             gain_q = △Q between v and n
11:             if best_q < gain_q then
12:                 best_q = gain_q
13:                 best_c = community of n
14:         C = Place v in the best_c
15:         for all neighboring nodes n of v do
16:             if n is not in community of v then
17:                 P = P + {n}
18:     return C
```

Algoritam izračunava $\triangle Q$ samo za vrhove u zajednici P za koje postoji vjerojatnost da promijene svoju zajednicu u budućnosti. U početku, svi vrhovi imaju istu vjerojatnost da bi promijenili svoju zajednicu (redak 3). Zatim je odabran slučajni vrh v i uklonjen iz P (redak 5-6). Samo susjedni vrhovi od v koji su u istoj zajednici kao v dodani su u P. Ovakav odabir značajno smanjuje vrijeme računanja algoritma. Ovi se procesi ponavljaju sve dok P ne ostane bez vrhova, a to znači da niti jedan vrh nema veliku vjerojatnost da promijeni svoju zajednicu (redak 4).

Literatura

[1] V. D. Blondel, J. L. Guillaume, R. Lambiotte & E. Lefebvre , *Fast unfolding of communities in large networks*, Journal of Statistical Mechanics: Theory and Experiment 10008, 12 pp. (2008)

[2] R. Diestel, *Graph Theory*, Electronic Edition, 2000.

[3] E. Estrada, *The Structure of Complex Networks – Theory and Applications*, Oxford University Press, 2012.

[4] D. Fasinoa, F. Tudisco *Generalized modularity matrices*, Linear Algebra and its Applications 502 (2016), 327–345.

[5] S. Fortunato, *Community detection in graphs*, Physics Reports 486 (2010), 75–174.

[6] A. Mishra, *Demystifying Louvain's Algorithm and Its implementation in GPU*, https://medium.com/walmartglobaltech/demystifying-louvains-algorithm-and-its-implementation-in-gpu-9a07cdd3b010

[7] M. E. J. Newman, *An Introduction to Networks*, Oxford University Press, 2010.

[8] M. E. J. Newman, L. A. Barabási, D. J. Watts, *The Structure and Dynamics of Networks*, Princeton University Press, 2011.

[9] P. Van Mieghem, *Graph Spectra for Complex Networks*, Cambridge University Press, 2011.

[10] N. Ozaki, H. Tezuka, M. Inaba, *A Simple Acceleration Method for the Louvain Algorithm*, International Journal of Computer and Electrical Engineering 8 (2016), 207–218.

[11] D. Veljan, *Kombinatorika s teorijom grafova*, Školska knjiga, Zagreb, 1989.

[12] J. Zhang, J. Fei, X. Song, J. Feng, *An Improved Louvain Algorithm for Community Detection*, Mathematical Problems in Engineering 3 (2021), 1–14.

Sažetak

Tema ovog diplomskog rada je detekcija gustih vrhova u kompleksnoj mreži. Analiza kompleksnih mreža ima važan istraživački značaj u raznim znanstvenim i društvenim disciplinama. Prvo ćemo navesti osnovne pojmove o grafovima, odnosno mrežama, a zatim ćemo opisati Louvain algoritam za particioniranje kompleksnih mreža. Njegov princip je postići maksimalnu vrijednost modularnosti u kontinuiranim iteracijama prolaska kroz vrhove grafa i dobiti optimalnu particiju mreže. Predstavit ćemo i algoritam Louvain Prune koji smanjuje vremensku složenost izvornog algoritma, a gotove je iste kvalitete kao izvorni algoritam.

Ključne riječi

kompleksna mreža, zajednica, algoritam za klasteriranje, Louvain algoritam, pohlepni algoritam.

Algorithm for recognizing dense groups of vertices in a complex network

Summary

The subject of this final work is the detection of dense groups of vertices in a complex network. The analysis of complex networks has an important research significance in various scientific and social disciplines. We will first present basic definitions from graph and network theory, and then describe the Louvain algorithm for partitioning complex networks. Its principle is to achieve the maximum modularity value of the community partition result in continuous iterations of passing through the vertices of the graph and obtain the optimal community partition. We will present Louvain Prune algorithm, which reduces the time complexity of the original algorithm and maintains the same qualities as the original algorithm.

Keywords

complex network, community, clustering algorithm, Louvain algorithm, greedy algorithm.

Životopis

Denis Poljak, rođen 11.02.1998. godine u Osijeku. Nakon završene osnovne škole u Tenji, upisuje III. gimnaziju u Osijeku. Obrazovanje nastavlja na preddiplomskom studiju Matematike i računarstva na Odjelu za matematiku 2017. godine te završava 2020. godine uz završni rad "Digitalna obrada slike u Pythonu" kod mentora izv. prof. dr. sc. Domagoja Matijevića. Rad je rađen u suradnji s tvrtkom Orqa. U sklopu preddiplomskog studija odradio je stručnu praksu u tvrtki Mono u Osijeku. Iste godine upisuje diplomski studiji Matematike i računarstva na Odjelu za matematiku. Na diplomskom studiju odradio je stručnu praksu u tvrtki Orqa u kojoj i trenutno radi.