Trokuti, klinovi i klasteriranje u usmjerenim grafovima

Markežić, Marko

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Marko Markežić

TROKUTI, KLINOVI I KLASTERIRANJE U USMJERENIM GRAFOVIMA

Diplomski rad

Voditelj rada:
prof.dr.sc. Zlatko Drmač

Zagreb, 09.2017.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

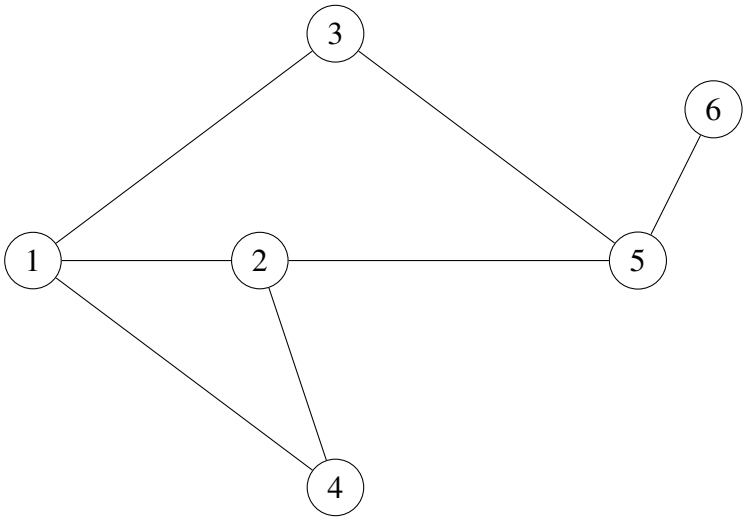
Sadržaj

Uvod

Mrežne strukture nalaze se svugdje oko nas. Možemo ih naći u web stranicama, društvenim mrežama, fizičkim interakcijama, molekularnoj povezanosti, epidemijama, bankovnim transakcijama, kupovinama, citiranjima i mnogim drugima. Iako je većina mreža koje su nam zanimljive usmjerenog tipa, razni algoritmi, zbog jednostavnosti i brzine, prvo pretvore takve mreže u neusmjerene grafove te zatim rade razne analize nad njima. Takav pristup se do sada pokazao dovoljno dobrim jer se redukcijom kompleksnosti grafa smanjila kompleksnost data mining problema, a pritom se i dalje otkrilo puno korisnih informacija o samoj mreži.

Mreže se mogu sastojati od desetaka, stotina pa i milijardi različitih čvorova, a bez obzira na njihovu veličinu, javlja se potreba da se na efikasan način pristupi cijeloj mreži i otkriju korisne podstrukture koje ćemo u nastavku rada nazivati "zajednicama". Postavlja se pitanje koja je definicija tih zajednica i kako mjeriti njihovu kvalitetu, međutim za sada ne postoji univerzalan odgovor na ovo pitanje. Intuitivno gledajući, zajednicu možemo promatrati kao skup čvorova koji su međusobno "dobro" povezani, a ta povezanost može se iskazati matematičkim mjerama baziranim na pojmovima klinova i trokuta, gdje se klin definira kao put unutar grafa duljine 2, dok je trokut zatvoreni put duljine 3. Trokuti su dobri indikatori zajednica jer predstavljaju homofilnost (ljudi postaju prijatelji s onima sličnima sebi) i tranzitivnosti (prijatelji prijatelja postaju prijatelji). Primjer klina na Slici 0.1 centriranog u vrhu 2 je 1-2-3, dok je jedan primjer trokuta 1-2-4.

U ovom ću radu prvo predstaviti teorijsku pozadinu računanja broja trokuta i klinova te raznih matematičkih mjera baziranih na tim pojmovima u neusmjerenom grafu koristeći algoritam uniformnog uzorkovanja klinova, zatim ću definirati pojam zajednica, predstaviti matematičke teoreme o zajednicama te predstaviti BTER (Block Two-Level Erdo-Renyi) model koji dobro opisuje realne grafove i sadrži korisne podstrukture. Nakon toga slijedi predstavljanje algoritma brojanja trokuta u usmjerenom grafu i komentar kako bi se BTER model mogao prilagoditi na usmjereni slučaj. Dodatno, sve opisane algoritme isprobat ću na realnom i javno dostupnom skupu podataka.



Slika 0.1: Prikaz neusmjerenog grafa koji sadrži 6 vrhova, 7 bridova, 10 klinova i jedan trokut.

Poglavlje 1

Neusmjereni grafovi

Notacija

Neka su n, m, W i T redom broj vrhova, bridova, klinova i trokuta u grafu G. Naš je zadatak aproksimirati broj trokuta T, definirati mjere vezane uz broj trokuta i na temelju tih mjera napraviti algoritam klasteriranja pomoću kojeg ćemo pronaći zajednice unutar mreže. Definirajmo prvo sljedeće matematičke mjere nad neusmjerenim grafovima:

$\kappa = \frac{3T}{W}$	tranzitivnost
$C_v = \frac{T_v}{W_v}$	koeficijent klasteriranja vrha v
$C = \frac{\sum_v C_v}{n}$	lokalni koeficijent klasteriranja
$C_d = \frac{\sum_{v \in V_d} C_v}{n_d}$	koeficijent klasteriranja po stupnjevima

Tablica 1.1: Mjere na neusmjerenim grafovima

n	broj vrhova	n_d	broj vrhova stupnja d
m	broj bridova	d_v	stupanj vrha v
V_d	skup vrhova stupnja d	W	ukupan broj klinova
W_v	broj klinova centriranih u vrhu v	T	ukupan broj trokuta
T_v	broj trokuta koji sadrže vrh v	T_d	broj trokuta koji sadrže vrh stupnja d

Tablica 1.2: Notacija

Na grafu sa Slike 0.1 možemo izračunati npr tranzitivnost $\kappa = \frac{3T}{W} = \frac{3}{7}$. Intuitivno gledajući, tranzitivnost nam pokazuje koliko su često prijatelji prijatelja i međusobno prijatelji te je ona pokazatelj "dobre povezanosti" unutar grafa. Općenito je najpreciznija

metoda računanja gore navedenih mjera iz Tablice 1.1 ručni izračun, ali jasno je da je kod većih mreža skoro pa i nemoguće računati na taj način. Stoga je potreban drugačiji pristup računanja broja bridova i trokuta unutar grafa, pristup koji će aproksimirati njihov broj s relativno malom pogreškom, a s druge strane dovoljno brzo izračunati tražene vrijednosti. Jedan od takvih algoritama je **algoritam uzorkovanja klinova** koji se može koristiti za procjenu svih mjera navedenih u Tablici 1.1. Primjerice, ovim algoritmom za procjenu tranzitivnosti κ s pogreškom $\epsilon \leq 0.1$ (s točnošću 99.9%) trebamo samo 380 slučajno odabranih bridova, a procjena tranzitivnosti je neovisna o ukupnoj veličini mreže. Također, ovaj se algoritam može primijeniti i za procjenu broja trokuta koji sadrži barem jedan vrh stupnja d (u oznaci T_d s Tablice 1.2) te za procjenu broja trokuta u usmjerenim grafovima. U usmjerenim grafovima postoji 7 različitih vrsta trokuta pa ih je samim time i teže prebrojati.

1.1 Algoritam uzorkovanja klinova

Prije predstavljanja samog algoritma, navedimo notaciju vezanu uz klinove. Kažemo da je klin zatvoren ako je dio trokuta, u suprotnom je otvoren. Srednji vrh klina zove se *centar klina*, primjerice klinovi sa Slike 0.1 (3)-(5)-(6) i (2)-(5)-(6) su centrirani u vrhu (5). Fiksirajmo neku distribuciju klinova i neka je w proizvoljan klin. Neka je X slučajna varijabla definirana na sljedeći način:

$$X(w) = \begin{cases} 1, & \text{ako je } w \text{ zatvoren} \\ 0, & \text{ako je } w \text{ otvoren} \end{cases}$$

Neka je $\mu = \mathbf{E}[X]$ i pretpostavimo da želimo procijeniti μ. Možemo jednostavno uzeti k nezavisnih klinova $w_1, w_2, \ldots, w_k$ s odgovarajućim slučajnim varijablama $X_1, X_2, \ldots, X_k$ i izračunati $\bar{X} = \frac{1}{k}\sum_{i=1}^{k} X_i$. Sljedeći teorem nam govori koliko je $\bar{X}$ dobra procjena za μ:

Teorem 1.1.1 (Hoeffdingov teorem)**.** *Neka su $X_1, X_2, \ldots, X_k$ nezavisne slučajne varijable, $X_i \in [0,1]$, $\forall i = 1, 2, \ldots, k$. Neka je $\bar{X} = \frac{1}{k}\sum_{i=1}^{k} X_i$ i neka je $\mu = \boldsymbol{E}[\bar{X}]$. Tada $\forall \epsilon \in \langle 0, 1\rangle$ vrijedi:*

$$P\left\{\left|(\bar{X} - \mu)\right| \geq \epsilon\right\} \leq 2exp\left(-2k\epsilon^2\right).$$

Nadalje, ako stavimo da je $k = \lceil 0.5\epsilon^{-2} ln(2/\delta)\rceil$, slijedi da je $\boldsymbol{P}\left\{\left|(\bar{X} - \mu)\right| \geq \epsilon\right\} < \delta$. Drugim riječima, za k uzoraka, s pouzdanošću od $1 - \delta$ greška naše procjene je najviše ϵ.

1.2 Tranzitivnost i broj neusmjerenih trokuta

Pretpostavimo da su klinovi uniformno distribuirani što dobijemo na način da svakom vrhu v uniformno slučajno pridružimo njegov par susjeda. **E**[X] možemo interpretirati kao vje-

rojatnost da je uniformno slučajno odabran klin zatvoren. Da bismo generirali uniformno slučajni klin, primijetimo prvo da je broj klinova centriranih u vrhu v jednak $W_v = \binom{d_v}{2}$, a ukupni broj klinova jednak je $W = \sum_v W_v$. Stavimo sada $p_v = \frac{W_v}{W}$ da dobijemo distribuciju po vrhovima. Možemo vidjeti da je vjerojatnost odabira vrha v proporcionalna broju klinova centriranih u v, a uniformno slučajan klin centriran u v možemo dobiti tako da na slučajan način izaberemo 2 susjeda vrha v.

Slutnja 1.2.1. *Izaberimo vrh v s vjerojatnošću p_v i uzmimo uniformno slučajan par susjeda od v. Ovime smo dobili uniformno slučajan klin.*

Dokaz. Neka je klin w centriran u vrhu v. Vjerojatnost da je vrh v izabran je $p_v = \frac{W_v}{W}$. Nadalje, vjerojatnost odabira slučajnog para susjeda od v je $1/\binom{d_v}{2} = \frac{1}{W_v}$. Slijedi da je vjerojatnost odabira klina w jednaka $\frac{1}{W}$ čime smo pokazali da je klin w uniformno generiran. □

Sada slijedi algoritam za računanje tranzitivnosti κ u grafu G prema [5] i [6]. Primijetimo da smo u prvom koraku algoritma pretpostavili da unaprijed znamo stupnjeve svih vrhova.

Algoritam 1 Računanje tranzitivnosti κ (κ-klin uzorkovanje) prema [6]

1. Uzmi k slučajnih vrhova (s mogućim ponavljanjima) prema vjerojatnosnoj distribuciji definiranoj sa p_v gdje je $p_v = \frac{W_v}{W}$.
2. Za svaki izabrani vrh v, izaberi dva susjeda od v (uniformno slučajno bez ponavljanja) da bismo dobili slučajan klin.
3. Izračunaj procjenu tranzitivnosti κ kao udio zatvorenih klinova u skupu klinova iz koraka 2.

Slijedi i primjer implementacije Algoritma 1 u programskom jeziku R-u za koji je potrebno instalirati paket *igraph*. Detaljnije o programskom jeziku može vidjeti na [1]. Funkcija koja računa procjenu tranzitivnosti kao argumente prima:

- veličinu uzorka,
- graf u obliku prilagođenom *igraph* paketu u R-u,
- skup svih vrhova grafa u obliku vektora,
- vjerojatnosnu distribuciju odabira vrhova u oznaci p_v.

Sve te vrijednosti potrebno je unaprijed izračunati, međutim taj dio nije prikazan u kodu jer se jednostavno dobije koristeći gotove funkcije iz *igraph* paketa. Dodatno, ova funkcija vraća i ukupno vrijeme potrebno za izvršenje R koda pa se tako može pratiti i efikasnost algoritma.

```
##tranzitivnost
#### k = velicina uzorka
#### g = graf u obliku graph.data.frame
####objekta iz igraph paketa
#### V = skup svih vrhova grafa g
#### p_v = vjerojatnosna distribucija biranja
####vrhova, p_v = W_v / W
procjena_kappa <- function(k, g, V, p_v) {
  start.time <- Sys.time()
  cnt <- 0
  n <- length(V)
  if (k > n){
    k <- n
  }
  uzorak <- sample(V, size=k, replace=TRUE, prob=p_v)
  for (i in uzorak){
    susjedi<-sample(unique(neighbors(g, i)),2,
    replace=FALSE)
    if(are.connected(g,susjedi[1], susjedi[2])){
      cnt <- cnt +1
    }
  }
  end.time <- Sys.time()
  time.taken <- end.time - start.time
  return (list(cnt/k, time.taken))
}
```

Kombinirajući Teorem 1.1.1 i Slutnju 1.2.1 dobijemo sljedeći teorem:

Teorem 1.2.2. *Neka je* $k = \left\lceil 0.5\epsilon^{-2}ln(2/\delta)\right\rceil$. *Algoritam 1 daje procjenu* $\bar{X}$ *za tranzitivnost* κ *tako da vrijedi:*

$$P\left\{\left|\bar{X} - \kappa\right| < \epsilon\right\} > 1 - \delta.$$

Procjenu za broj trokuta T u grafu G možemo jednostavno izračunati na sljedeći način:

$$T \approx \bar{X} \cdot \frac{W}{3},$$

pri čemu iz Teorema 1.2.2 slijedi da grešku procjene broja trokuta možemo iskazati kao:

$$\mathrm{P}\left\{\left|\bar{X}\cdot\frac{W}{3}-T\right|<\epsilon\cdot\frac{W}{3}\right\}>1-\delta.$$

1.3 Lokalni koeficijent klasteriranja

U ovom ću odjeljku prikazati kako s malom modifikacijom početne distribucije klinova možemo izračunati koeficijent klasteriranja C. Jedina razlika između algoritma koji računa koeficijent klasteriranja C od algoritma koji računa tranzitivnost κ je način izbora vrhova u koraku 1. U Algoritmu 1 su vrhovi birani prema vjerojatnosnoj distribuciji $\{p_v\}$ dok će se u Algoritmu 2 vrhovi birati uniformno slučajno.

Algoritam 2 Računanje lokalnog koeficijenta klasteriranja C (C-klin uzorkovanje) prema [6]

---

1. Uzmi k uniformno slučajnih vrhova (s mogućim ponavljanjima)
2. Za svaki izabrani vrh v, izaberi dva susjeda od v (uniformno slučajno bez ponavljanja) da bismo dobili slučajan klin.
3. Izračunaj procjenu koeficijenta klasteriranja C kao udio zatvorenih klinova u skupu klinova iz koraka 2.

Slijedi i primjer implementacije Algoritma 2 u R-u u obliku funkcije koja prima sljedeće argumente:

- veličinu uzorka,
- graf u obliku prilagođenom *igraph* paketu u R-u,
- ukupan broj vrhova u obliku vektora.

Da bi ovaj kod bilo moguće uspješno pokrenuti potrebno je prethodno instalirati R paket *igraph*, a dodatno se, kao i u prethodnom algoritmu, prati vrijeme izvršavanja algoritma za praćenje efikasnosti koda:

```
#### racunanje lokalnog koeficijenta klasteriranja C
#### k = velicina uzorka
#### g = graf u obliku graph.data.frame
###objekta iz igraph paketa
#### V = skup svih vrhova grafa g
```

```
procjena_C <- function(k, g, V){
  start.time <- Sys.time()
  cnt <- 0
  n <- length(V)
  if (k > n){
    k <- n
  }
  uzorak <- sample(V, size=k, replace=TRUE)
  for (i in uzorak){
    susjedi<-sample(unique(neighbors(g, i)),2,
    replace=FALSE)
    if(are.connected(g,susjedi[1], susjedi[2])){
      cnt <- cnt +1
    }
  }
  end.time <- Sys.time()
  time.taken <- end.time - start.time
  return (list(cnt/k, time.taken))
}
```

Teorem 1.3.1. *Neka je* $k = \left\lceil 0.5\epsilon^{-2}ln(2/\delta)\right\rceil$. *Algoritam 2 daje procjenu* $\bar{X}$ *za lokalni koeficijent klasteriranja C tako da vrijedi:*

$$P\left\{\left|\bar{X} - C\right| < \epsilon\right\} > 1 - \delta.$$

Dokaz. Neka je w proizvoljan klin i $X(w)$ slučajna varijabla takva da vrijedi:

$$X(w) = \begin{cases} 1, & \text{ako je } w \text{ zatvoren} \\ 0, & \text{ako je } w \text{ otvoren.} \end{cases}$$

Neka je $\mathcal{V}$ uniformna distribucija bridova. Za svaki vrh v, neka je $\mathcal{N}_v$ uniformna distribucija parova susjeda od vrha v. Primijetimo da je:

$$\mathrm{E}[X] = \mathop{\mathrm{P}}_{v\sim\mathcal{V}}\left[\mathop{\mathrm{P}}_{(u,u')\sim\mathcal{N}_v}\left\{\text{klin}\left\{(u,v),(u',v)\right\}\text{je zatvoren}\right\}\right]$$

Raspišimo sada čemu je koeficijent klasteriranja C jednak:

$$\begin{aligned} C &= n^{-1} \sum_{v} C_v \\ &= \mathrm{E}_{v\sim\mathcal{V}}[C_v] \\ &= \mathrm{E}_{v\sim\mathcal{V}}[\text{udio zatvorenih klinova centriranih u vrhu v}] \\ &= \mathrm{E}_{v\sim\mathcal{V}}[\mathop{\mathrm{P}}_{(u,u')\sim\mathcal{N}_v}\{\text{klin}\{(u,v),(u',v)\}\text{ je zatvoren}\}] \\ &= \mathop{\mathrm{P}}_{v\sim\mathcal{V}}[\mathop{\mathrm{P}}_{(u,u')\sim\mathcal{N}_v}\{\text{klin}\{(u,v),(u',v)\}\text{ je zatvoren}\}] \\ &= \mathrm{E}[X] \end{aligned} \tag{1.1}$$

Prema Hoeffdingovom teoremu 1.1.1 slijedi tvrdnja teorema. □

1.4 Koeficijent klasteriranja po stupnjevima

U ovom će se odjeljku predstaviti algoritam računanja koeficijenta klasteriranja po stupnjevima. Radi se o lokalnom koeficijentu klasteriranja koji se računa na podskupu vrhova fiksiranog stupnja d.

Teorem 1.4.1. *Neka je* $k = \left\lceil 0.5\epsilon^{-2} ln(2/\delta)\right\rceil$. *Algoritam 3 (*$C_d$*-klin uzorkovanje) daje procjenu* $\bar{X}$ *za lokalni koeficijent klasteriranja po stupnjevima* C_d *tako da vrijedi:*

$$P\left\{\left|\bar{X} - C_d\right| < \epsilon\right\} > 1 - \delta.$$

Dokaz. Dokaz je analogan dokazu Teorema 1.3.1. □

Algoritam iz prethodnog odjeljka daje procjenu koeficijenta klasteriranja vrhova uz poznate stupnjeve vrhova. U praksi, dovoljno je računati koeficijent klasteriranja nad nekim skupom stupnjeva. Algoritmi "klin uzorkovanja" mogu, uz malo prilagodbe, funkcionirati i na manjim skupovima stupnjeva. Unutar svakog skupa dajemo svakom vrhu određenu težinu, sukladno broj klinova koju je proizveo. Ovo nam garantira da će svaki klin iz skupa biti jednako vjerojatno izabran. Primjerice, ako stavimo u isti skup vrhove stupnja 3 i 4, vrhovima koji imaju stupanj 4 dodijelit ćemo duplo veću težinu od vrhova stupnja 3 jer vrhovi stupnja 3 generiraju $\binom{3}{2} = 3$ klina dok vrhovi stupnja 4 generiraju $\binom{4}{2} = 6$ klinova.

Algoritam 3 Računanje koeficijenta klasteriranja po stupnjevima (C_d-klin uzorkovanje) prema [6]

1. Uzmi k uniformno slučajnih vrhova stupnja d (s mogućim ponavljanjima)
2. Za svaki izabrani vrh v, izaberi dva susjeda od v (uniformno slučajno bez ponavljanja) da bismo dobili slučajan klin.
3. Izračunaj procjenu koeficijenta klasteriranja C_d kao udio zatvorenih klinova u skupu klinova iz koraka 2.

Kao i u prethodnim algoritmima, slijedi implementacija Algoritma 3 u R-u za koji je potrebna instalacija paketa *igraph*. Ovoga puta se u R funkciju prosljeđuju malo izmijenjeni argumenti u odnosu na prethodno implementirane algoritme, a to su:

- veličina uzorka,
- graf u *igraph* obliku,
- distribucija stupnjeva po vrhovima u obliku vektora,
- stupanj d za koji računamo koeficijent klasteriranja C_d.

I ovdje se također prati brzina izvršavanja R koda da bi se pratila efikasnost algoritma.

```
#### racunanje koeficijenta klasteriranja po stupnjevima C_d
#### k = velicina uzorka
#### g = graf u obliku graph.data.frame
####objekta iz igraph paketa
#### d_v = distribucija stupnjeva po vrhovima
#### d = proizvoljan stupanj za koji racunamo algoritam
procjena_C_d <- function(k, g, d_v, d){
  start.time <- Sys.time()
  cnt <- 0
  V_d <- d_v[d_v==d]
  ## ukupan broj vrhova stupnja d u grafu g
  n_d <- sum(d_v == d)
  if ( k > n_d) {
    k <- n_d
  }
  uzorak <- sample(names(V_d), size=k, replace=TRUE)
  for (i in uzorak){
```

```
    susjedi<-sample(unique(neighbors(g, i)),2,
    replace=FALSE)
    if(are.connected(g,susjedi[1], susjedi[2])){
      cnt <- cnt +1
    }
  }
  end.time <- Sys.time()
  time.taken <- end.time - start.time
  return (list(cnt/k, time.taken))
}
```

1.5 Broj neusmjerenih trokuta po stupnjevima

Ako modificiramo prethodno navedene algoritme, možemo dobiti procjenu za T_d (broj trokuta vezanih uz vrhove stupnja d) na način da umjesto računanja udjela zatvorenih klinova, računamo težinsku sumu. Algoritam računanja neusmjerenih trokuta po stupnjevima prikazan je sljedećom shemom:

Algoritam 4 Računanje neusmjerenih trokuta po stupnjevima (T_d-klin uzorkovanje) prema [6]

---

1. Uzmi k uniformno slučajnih vrhova stupnja d (s mogućim ponavljanjima)
2. Za svaki izabrani vrh v, izaberi dva susjeda od v (uniformno slučajno bez ponavljanja) da bismo dobili slučajan klin.
3. Za svaki klin w_i iz prethodnog koraka algoritma definiraj slučajnu varijablu Y_i kao:

$$Y_i = \begin{cases} 0, & \text{ako je w otvoren klin,} \\ 1/3, & \text{ako je w zatvoren i ima 3 vrha stupnja d,} \\ 1/2, & \text{ako je w zatvoren i ima 2 vrha stupnja d,} \\ 1, & \text{ako je w zatvoren i ima 1 vrh stupnja d.} \end{cases}$$

4. Neka je $\bar{Y} = \frac{1}{k} \sum_i Y_i$.
5. Izračunaj $W_d \cdot \bar{Y}$ kao procjenu za T_d.

Slijedi implementacija Algoritma 4 u R-u za koji je potrebno instalirati paket *igraph*. Funk-

ciji koja računa broj neusmjerenih trokuta po stupnjevima potrebno je proslijediti iste argumente kao i u Algoritmu 3:

- veličinu uzorka,
- graf u obliku *igraph* objekta,
- distribucija stupnjeva po vrhovima u obliku vektora,
- stupanj d za koji računamo T_d.

Kao i u svim prethodnim algoritmima, prati se vrijeme izvršavanja R koda.

```
#### racunanje neusmjerenih trokuta po stupnjevima T_d
#### k = velicina uzorka
#### g = graf u obliku graph.data.frame
####objekta iz igraph paketa
#### d_v = distribucija stupnjeva po vrhovima
#### d = proizvoljan stupanj za koji racunamo algoritam
procjena_T_d <- function(k, g, d_v, d){
  start.time <- Sys.time()
  y <- 0
  ## ukupan broj vrhova stupnja d u grafu g
  n_d <- sum(d_v == d)
  ## ukupan broj klinova centriranih u vrhovima stupnja d
  W_d <- n_d * choose(d,2)
  V_d <- d_v[d_v==d]
  if (k > n_d){
    k <- n_d
  }
  uzorak <- sample(names(V_d), size=k, replace=TRUE)
  for (i in uzorak){
    susjedi<-sample(unique(neighbors(g, i)),2,
    replace=FALSE)
    if(are.connected(g,susjedi[1], susjedi[2])){
      if(sum(d_v[susjedi] == d) == 2){
        y <- y + 1/3
      } else if (sum(d_v[susjedi] == d) == 1){
        y <- y + 1/2
      } else {
        y <- y + 1
```

```
        }
      }
    }

    y_avg <- y/k
    end.time <- Sys.time()
    time.taken <- end.time - start.time
    return (list(W_d * y_avg, time.taken))
}
```

Teorem 1.5.1. *Neka je* $k = \left\lceil 0.5\epsilon^{-2} ln(2/\delta) \right\rceil$ *i* $W_d = n_d \cdot \binom{d}{2}$ *ukupan broj klinova centriranih u vrhovima stupnja d. Algoritam 4 (*T_d*-klin uzorkovanje) daje procjenu* $W_d \cdot \bar{Y}$ *za broj trokuta vezanih uz vrhove stupnja d (u oznaci* T_d*) tako da vrijedi:*

$$P\left\{\left|W_d \cdot \bar{Y} - T_d\right| < \epsilon W_d\right\} > 1 - \delta.$$

Dokaz. Za svaki klin w_i definirajmo Y_i. Pokazat ćemo da je očekivana vrijednost E[Y] = $\frac{T_d}{W_d}$. Kada to dokažemo, prema Hoeffdingovom teoremu 1.1.1, nakon množenja nejednakosti s W_d, dobijemo da vrijedi $\left|W_d \cdot \bar{Y} - T_d\right| < \epsilon W_d$ s vjerojatnošću većom od $1 - \delta$.

Pokažimo sada da vrijedi E[Y] = $\frac{T_d}{W_d}$. Particionirajmo skup svih klinova centriranih u vrhu d na četiri skupa S_0, S_1, S_2, S_3 tako da skup S_i, i=1,2,3, sadrži sve zatvorene klinove koji sadrže točno i vrhova stupnja d. Skup S_0 sadrži preostale otvorene klinove. Za proizvoljan klin w vrijedi:

$$Y_i = \begin{cases} 0, & w \in S_0, \\ \frac{1}{i}, & w \in S_i. \end{cases}$$

Klin w je uniformno slučajno izabran klin iz skupa svih klinova centriranih u vrhovima stupnja d pa slijedi:

$$\mathrm{E[Y]} = \frac{(|S_1| + |S_2|/2 + |S_3|/3)}{W_d}.$$

Particionirajmo sada skup trokuta koji sadrže barem jedan vrh stupnja d na skupove S_1', S_2', S_3', gdje je S_i' skup trokuta koji sadrži točno i vrhova stupnja d. Ako trokut sadrži točno i vrhova stupnja d, onda u tom trokutu postoji točno i klinova centriranih u vrhu stupnja d. Iz toga slijedi da je $|S_i| = i \cdot |S_i'|$. Konačno vrijedi:

$$(|S_1| + |S_2|/2 + |S_3|/3) = (|S_1'| + |S_2'| + |S_3'|) = T_d.$$

Dijeljenjem gornje jednažbe s W_d slijedi E[Y] = $\frac{T_d}{W_d}$ čime smo dokazali tvrdnju. □

1.6 Implementacija algoritama klin uzorkovanja nad stvarnim podacima

U ovom ću odjeljku testirati prethodno navedene algoritme nad stvarnim, javno dostupnim podacima. Prilikom traženja odgovarajućih skupova podataka bile su bitne dvije stvari: da se podaci mogu prikazati u obliku dovoljno velikog grafa (tu mislim na barem 50 000 vrhova po grafu) te da podaci dolaze iz interakcijskih društevnih mreža. Drugi uvjet je bio bitan zbog potencijalnih primjena ovog rada na razna područja u kojima se promatraju ljudske interakcije kao što je primjerice detektiranje prijevara u bankovnim transakcijama, a dodatno se taj uvjet uklapa u priču oko *zajednica* koje se spominju u nastavka rada. Također, poželjno je da su barem neki od podataka usmjerenog tipa da bi se mogli isprobati i algoritmi klin uzorkovanja nad usmjerenim grafovima.
Podaci su preuzeti sa *Stanford Large Network Dataset Collection* stranice označene u literaturi brojem [2]. Radi se o sljedećim podacima:

1. **soc-Slashdot0922** je usmjerena mreža sa 82,168 vrhova i 948,464 bridova preuzeta u veljači 2009. sa Slashdot stranice. Slashdot je web stranica usmjerena tehnološkim novostima koja omogućava da korisnici međusobno označavaju (”tagiraju”) jedni druge kao prijatelje. Mreža sadrži poveznice između korisnika Slashdot stranice.

2. **email-EuAll** je usmjerena mreža mailova koja se sastoji od 265,214 vrhova i 420,045 bridova. Mailovi su anonimizirani i prikupljeni u periodu od listopada 2003. do svibnja 2005. od strane velikog europskog istraživačkog centra i sadrže podatke o primateljima i pošiljateljima mailova.

3. **wiki-Talk** je usmjerena mreža koja se sastoji od 2,394,385 vrhova i 5,021,410 bridova. Wikipedia je besplatna online enciklopedija koju su napisali volonteri diljem svijeta. Svaki registrirani korisnik ima svoju stranicu na kojoj on/ona ili bilo koji drugi korisnik mogu komentirati novosti oko raznih članaka s Wikipedije. Skup podataka koji koristim sadrži sve korisnike i njihove diskusije od osnivanja Wikipedije do siječnja 2008. Vrhovi predstavljaju korisnike Wikipedije, a usmjereni brid iz vrha i prema vrhu j govori da je korisnik i barem jednom uređivao stranicu korisnika j.

Bitno je za naglasiti da su se algoritmi vrtjeli na dosta slabom 32-bitnom računalu s 2GB RAM-a i procesorom od 2.20GHz pa je i vrijeme izvršavanja algoritama nešto sporije nego očekivano. Također, nije nužno da je vrijeme izvršavanje kraće na manjem uzorku zbog načina na koji algoritam radi. Naime, za svaki slučajno izabrani vrh algoritam na slučajan način traži njegova dva susjeda i gleda jesu li oni međusobno povezani. Međutim, ako su ti susjedi vrhovi velikog stupnja onda će samim time traženje poveznice između susjeda trajati više. U sljedećim tablicama mogu se vidjeti rezultati algoritma za računanje tranzitivnosti i koeficijenta klasteriranja ovisno o početnom broju uzorka. Jedna stvar koja se

može primijetiti je da koeficijent klasteriranja i nije baš dobro aproksimiran ovim algoritmom. Moguće je da se drugačije tumači koeficijent klasteriranja u [2].

Podaci	Broj vrhova	Broj bridova	Koeficijent klasteriranja C	Tranzitivnost κ
soc-Slashdot0922	82168	948464	0.0603	0.02410896
email-EuAll	265214	420045	0.0671	0.004106431
wiki-Talk	2394385	5021410	0.0526	0.002192441

Tablica 1.3: Glavna obilježja podataka

Veličina uzorka	$k = 380$	$k = 1000$	$k = 2000$	$k = 4000$
Koeficijent klasteriranja C	0.5921053	0.5700000	0.5775000	0.5720000
Tranzitivnost κ	0.02894737	0.04500000	0.03450000	0.03625000

Tablica 1.4: Implementacija algoritma klin uzorkovanja nad podacima *soc-Slashdot0922*

Veličina uzorka	$k = 380$	$k = 1000$	$k = 2000$	$k = 4000$
Koeficijent klasteriranja C	21.486229 sec	1.987714 min	52.147983 sec	3.018656 min
Tranzitivnost κ	5.043289 sec	26.838535 sec	41.630382 sec	1.110930 min

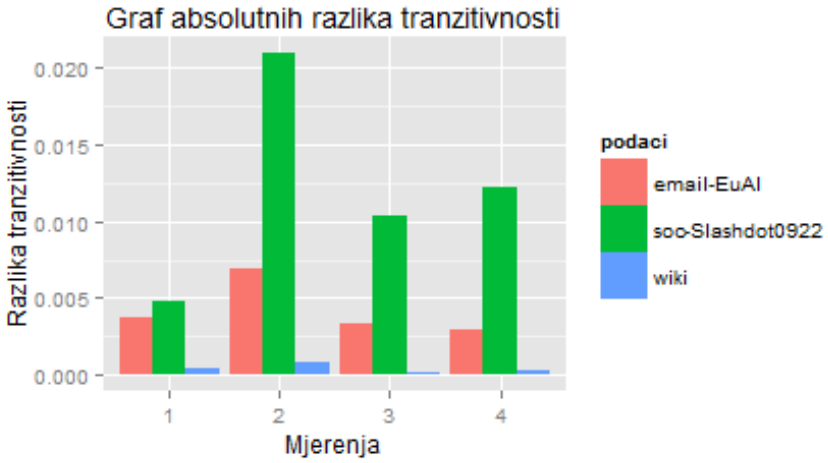
Tablica 1.5: Vrijeme izvršavanja algoritma nad podacima *soc-Slashdot0922*

Veličina uzorka	$k = 380$	$k = 1000$	$k = 2000$	$k = 4000$
Koeficijent klasteriranja C	0.07894737	0.07900000	0.07600000	0.08400000
Tranzitivnost κ	0.007894737	0.011000000	0.007500000	0.007000000

Tablica 1.6: Implementacija algoritma klin uzorkovanja nad podacima *email-EuAll*

Veličina uzorka	$k = 380$	$k = 1000$	$k = 2000$	$k = 4000$
Koeficijent klasteriranja C	8.522488 sec	20.158153 sec	39.231244 sec	1.340677 min
Tranzitivnost κ	35.277018 sec	28.502631 sec	1.026109 min	1.875241 min

Tablica 1.7: Vrijeme izvršavanja algoritma nad podacima *email-EuAll*



Slika 1.1: Prikaz absolutnih razlika simuliranih i stvarnih tranzitivnosti za svako mjerenje.

Veličina uzorka	$k = 380$	$k = 1000$	$k = 2000$
Koeficijent klasteriranja C	0.5710526	0.587	0.5755
Tranzitivnost κ	0.002631579	0.003	0.002

Tablica 1.8: Implementacija algoritma klin uzorkovanja nad podacima *wiki-Talk*

Veličina uzorka	$k = 380$	$k = 1000$	$k = 2000$
Koeficijent klasteriranja C	14.65784 sec	24.81642 sec	35.20001 sec
Tranzitivnost κ	1.787152 min	4.576412 min	10.41851 min

Tablica 1.9: Vrijeme izvršavanja algoritma nad podacima *wiki-Talk*

1.7 Teorija zajednica

U ovom ćemo se odjeljku usredotočiti na pojam zajednica, vidjeti zašto su nam one važne u razumijevanju samih mreža i predstaviti matematičke teoreme koji nam govore što je to zajednica. Intuitivno rečeno, zajednica je podskup vrhova grafa koji su međusobno ”dovoljno dobro povezani”, a veliki koeficijent klasteriranja unutar grafa je jedan od dobrih indikatora povezanosti vrhova. Također, možemo očekivati da će zajednice biti usko povezane s pojmovima trokuta koje smo definirali u prethodnim odjeljcima ovog rada. Zajednice

mogu biti različitih veličina, ali najčešće je najveća zajednica manja od samog grafa.
Krenimo prvo s notacijom. Neka imamo neusmjereni graf G s n vrhova sa stupnjevima $d_1, d_2, \ldots, d_n$. Neka je $m = \frac{1}{2}\sum_{i=1}^{n} d_i$ broj bridova grafa G. Kažemo da podgraf S ima veliku modularnost ako S sadržava mnogo više internih bridova nego što bi imao prema *null-modelu* koji kaže da su vrhovi i i j povezani s vjerojatnošću $d_i d_j/2m$ (ovo je zapravo samo točno ako pretpostavimo da vrijedi tvrdnja $d_i^2 \leq 2m,\ \forall i$, ali i jedan i drugi slučaj ćemo detaljnije obraditi u nastavku rada kod predstavljanja detaljnijih matematičkih teorema). Kada spominjemo *null-model* referiramo se na CL model koji je formaliziran od strane dvojca Chung i Lu i detaljnije opisan u [3].
Kažemo da je podgraf velike modularnosti S *modul* ako je sam po sebi dobro modeliran prema CL modelu. Formalno, pretpostavimo da S ima r vrhova sa stupnjevima $\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_r$ i neka je $s = \frac{1}{2}\sum_{i=1}^{r} \hat{d}_i$ broj bridova podgrafa S. Prema CL modelu, vjerojatnost da su vrhovi $i, j \in S$ međusobno povezani je $\hat{d}_i\hat{d}_j/2s$. Reći ćemo da je S *modul* ako je izvedeni graf od S (prema CL modelu) dobro modeliran. U suprotnom, S sadrži podskup vrhova koji se može izdvojiti u novi podgraf. *Modul* se može promatrati kao "atomska" substruktura unutar grafa. U ovom smislu, prepoznavanje zajednica u grafu se zapravo svodi na razdvajanje grafa na module. Vidjet ćemo da priča nije baš ovako jednostavna jer zajednice nisu isto što i moduli, ali su također vrhovi unutar zajednice dobro povezani.
Sada prema [3] formalno definiramo **zajednicu** kao modul koji ima veliki interni koeficijent klasteriranja. Ovo je malo drugačija definicija zajednica od onih koje možemo naći u literaturi gdje se zajednica općenito definira kao skup vrhova koji je više povezan interno, nego eksterno. Prema našoj definiciji, zajednica je gusto interno povezana pa stoga sadrži i veliki broj trokuta. Graf ima *strukturu zajednica* ako se on, ili dovoljno veliki dio grafa, može particionirati na zajednice. Prednost ovakve definicije je da sada možemo pokušati razumijeti kako izgledaju grafovi koji imaju strukturu zajednica.
Fokusirajmo se prvo na jednu zajednicu. Intuitivno se čini da zajednica u društvenim mrežama ne može biti istovremeno velika i sastavljena samo od vrhova malog stupnja, ali isto tako ne može biti sastavljena od jednog vrha velikog stupnja koji je povezan s drugim vrhovima stupnja jedan (struktura zvijezde). Erdos-Renyi (ER) graf s n vrhova i vjerojatnošću povezanosti p je graf takav da je svaki par vrhova nezavisno povezan s vjerojatnošću p. Ako je p konstanta onda se radi o gustom ER grafu, ako je $p = O(1/n)$ onda kažemo da se je to rijetki ER graf. Za gusti ER graf vrijedi sljedeći teorem, pri čemu oznaka $\Omega(\cdot)$ označava konstantni faktor, odnosno ako vrijedi da graf sadrži $\Omega(\sqrt{s})$ vrhova za neki broj bridova s to znači da je sadrži $c * \sqrt{s}$ vrhova za neku pozitivnu konstantu c:

Teorem 1.7.1. *Ako zajednica ima s bridova, tada mora sadržavati $\Omega(\sqrt{s})$ vrhova stupnja $\Omega(\sqrt{s})$.*

Ovaj teorem je zanimljiv jer iako je dobro poznato da ER grafovi nisu dobri modeli za interakcijske mreže, oni su ipak važan dio izgradnje zajednica. Ovaj teorem interpretiramo

na način da kažemo da je najjednostavnija moguća zajednica zapravo gusti ER graf. U ovom smislu, interakcijske mreže možemo promatrati kao veliki skup gustih ER grafova. Ovakvo razmišljanje nas prirodno vodi do pitanja distribucije veličine tih ER komponenti. Posljedica ove teorije je da će ER zajednica s $d + 1$ vrhova imati ρd^2 bridova za neku konstantu ρ. Za hipotezu o očekivanom broju trokuta, ρ će biti približno jednak 1. Nadalje, može se pokazati da interakcijski grafovi imaju *power-law* distribuciju teškog repa što je za i očekivati jer velike interakcijske mreže sadrže mali broj vrhova velikog stupnja, a veliki broj vrhova niskog stupnja. Ta distribucija je prikazana Slikom 1.2 i takva je da vrijedi:

$$X_d \propto d^{-\gamma},$$

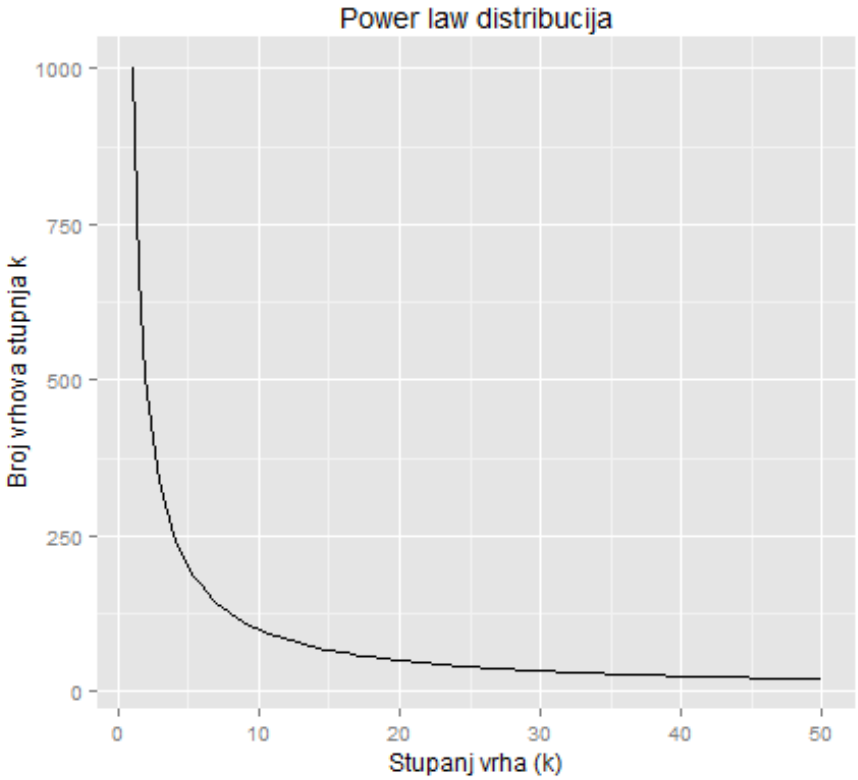
gdje je X_d broj vrhova stupnja d, γ *power-law* eksponent, a $\propto$ oznaka koja označava proporcionalnost.
Ako pretpostavimo da su svi vrhovi unutar zajednice jednakog stupnja, tada vrhovi stupnja d tvore $X_d/(d + 1)$ zajednica. Stoga, ako definiramo Y_d kao broj zajednica veličine $(d + 1)$ (s vrhovima stupnja d), vrijedi:

$$Y_d \propto \frac{X_d}{d+1} \propto d^{-(\gamma+1)}.$$

Ovo tvori *scale-free* distribuciju zajednica, distribuciju koja prati *power-law*, barem asimptotski. Zbog toga ćemo pretpostaviti da stvarne interakcijske mreže sadrže *scale-free* kolekciju gustih Erdos-Renyi grafova, odnosno da je distribucija veličine ER komponenti također distribucija teškog repa.
U literaturi je empirijski pokazano da postoji malo relativno velikih zajednica i puno malih zajednica. Uočeno je na velikom broju različitih grafova da najveća viđena zajednica ima oko 100 vrhova. Ovo je u skladu s našom hipotezom: ako pretpostavimo da je $X_d = n/d^\gamma$ i da postoji zajednica veličine d tada mora vrijediti $n/d^{\gamma+1} \geq 1$. Stoga je maksmimalna veličina zajednica $\bar{d}$ približno jednaka $\bar{d} \approx n^{1/(\gamma+1)}$. Za $n = 1000000$ i $\gamma = 2$ dobijemo procjenu veličine zajednice na 100 vrhova.
Dodatno, Teorem 1.7.1 također dokazuje da CL model sam po sebi nije dobar za interakcijske mreže. Pretpostavimo da se cijeli graf G s m vrhova može prikazati kao CL graf. Pošto G ima veliki koeficijent klasteriranja, G je modul. Stoga, G mora imati $\Omega(\sqrt{m})$ vrhova stupnja $\Omega(\sqrt{m})$, ali to nije u skladu s repnim ponašanjem distribucije stupnjeva.

Matematički detalji o zajednicama

U ovom će se odjeljku prvo predstaviti skica dokaza Teorema 1.7.1, a zatim i potpuni dokaz. Analiza je fundamentalno asimptotska tako da ćemo koristiti oznake $O(\cdot)$, $\Omega(\cdot)$ i $\Theta(\cdot)$ da prikažemo konstantne faktore. Oznaka $A \ll B$ nam govori da postoji konstanta c takva da vrijedi $A \leq cB$. Sa S označimo zajednicu koju promatramo i pretpostavimo da je



Slika 1.2: Prikaz power-law distribucije čvorova u mreži za $\gamma = 1$ prema funkciji $f(x) = 1000x^{-1}$ gdje je veliki broj čvorova malog stupnja, a mali broj čvorova velikog stupnja.

interna distribucija vrhova jednaka $\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_r$ takva da je $\hat{d}_i \leq \hat{d}_j, \ \forall i < j$. Broj bridova grafa S označavamo sa $s = \sum_{i=1}^{r} \hat{d}_i$.
Za danu distribuciju, neka je T očekivani broj trokuta u grafu S. Pošto se radi o zajednici, zahtijevamo da je očekivani broj trokuta T barem $\kappa/3$ puta veći od broja klinova za neku konstantu κ, odnosno vrijedi:

$$T \geq \frac{\kappa}{3} \sum_{i} \binom{\hat{d}_i}{2}. \tag{1.2}$$

Bez smanjenja općenitosti možemo pretpostaviti da je $\hat{d}_i > 1, \ \forall i$ jer vrhovi stupnja 1 ne mogu biti dijelovi trokuta. Ključna stvar koju koristimo je *Kruskal-Katona* teorem koji

kaže da ako graf ima T trokuta i s bridova, tada vrijedi $T \leq s^{3/2}$. Kombinirajući taj teorem s jednadžbom (1.2) dobijemo sljedeći izraz:

$$\sum_i \hat{d}_i^2 \ll s^{3/2}. \tag{1.3}$$

Pogledajmo sada koliki je očekivani broj trokuta prema CL distribuciji. Za svaku trojku (i, j, k), neka je slučajna varijabla X_{ijk} indikator da je (i, j, k) trokut, odnosno $\forall i, j, k \in V$ vrijedi:

$$X_{ijk} = \begin{cases} 1, & \text{ako je } (i, j, k) \text{ trokut,} \\ 0, & \text{inače.} \end{cases}$$

Trojka (i, j, k) čini trokut ako sadrži bridove (i, j), (j, k) i (k, i), a zbog nezavisnosti je vjerojatnost toga događaja jednaka $p_{ij}p_{jk}p_{ki}$, gdje je $p_{ij} = \hat{d}_i\hat{d}_j/2m$, $\forall i, j \in V$. Očekivani broj trokuta se može iskazati kao $E\left[\sum_{i<j<k} X_{ijk}\right]$ što je po linearnosti očekivanja jednako izrazu $\sum_{i<j<k} E\left[X_{ijk}\right]$. Prema tome vrijedi:

$$T = \sum_{i<j<k} \frac{\hat{d}_i\hat{d}_j}{2s}\frac{\hat{d}_j\hat{d}_k}{2s}\frac{\hat{d}_i\hat{d}_k}{2s} \leq \frac{\left(\sum_i \hat{d}_i^2\right)^3}{8s^3}. \tag{1.4}$$

Ranije je spomenuto da je $T = \Omega\left(\sum_i \hat{d}_i^2\right)$. Ako ovo uključimo u izraz (1.4) dobijemo:

$$s^{3/2} \ll \sum_i \hat{d}_i^2, \tag{1.5}$$

što je obrnuti izraz od (1.3). To znači da su lijeva i desna strana iz (1.5) jednake do na konstantni faktor. Možemo si sada postaviti pitanje kada će ti izrazi biti jednaki, a to se događa u slučaju kada se zajednica sastoji od ukupno $\sqrt{s}$ vrhova stupnjeva $\sqrt{s}$. Tada vrijedi $\sum_i \hat{d}_i^2 = \sum_i s = s^{3/2}$. Intuitivno, da bi se zadovoljile nejednakosti (1.3) i (1.5), mora biti $\Theta(\sqrt{s})$ vrhova stupnja $\Theta(\sqrt{s})$. Takvi vrhovi unutar zajednice tvore gusti ER graf što dokazuje da svaka zajednica sadrži konstantni omjer bridova iz ER grafa.

Ovime je završena skica dokaza Teorema 1.7.1, sada ćemo predstaviti detaljan dokaz. Dokaz je asimptotski proveden za broj bridova u zajednici što znači da je dokaz valjan za dovoljno veliki broj bridova s.
Definirajmo za svaki uređeni par (i, j), vjerojatost odabira brida (i, j) kao $p_{ij} = \hat{d}_i\hat{d}_j/(2s)^2$. Primijetimo da smo tako definirali vjerojatnosnu distribuciju po parovima jer vrijedi $\sum_{i,j} \hat{d}_i\hat{d}_j/(2s)^2 = \left(\sum_i \hat{d}_i\right)^2/(2s)^2 = 1$. Generiramo s nezavisnih uzoraka iz ove distribucije da dobijemo graf. Konačan graf je neusmjeren i jednostavan tako da maknemo sve paralelne bridove. Ovo je jedna od standardnih metoda dobivanja grafa preko modela s bridovima.

Teorem 1.7.2. *(Preformulirana tvrdnja Teorema 1.7.1) Neka imamo CL graf G sa stupnjevima vrhova* $1 < \hat{d}_1 \leq \hat{d}_2 \leq \ldots \leq \hat{d}_r$ *i neka je* $s = \sum_i \hat{d}_i/2$ *te* $c > 0$ *i* $\kappa \in < 0, 1 >$ *konstante neovisne o odabiru s. Pretpostavimo da je očekivani broj trokuta u grafu G barem* $(\kappa/3)\sum_i \binom{\hat{d}_i}{2}$. *Tada za dovoljno veliki s postoji skup indeksa* $U \subseteq \{1, \ldots, r\}$ *takav da vrijedi* $|U| = \Omega\left(\sqrt{s}\right)$ *i* $\hat{d}_k = \Omega\left(\sqrt{s}\right)$, $\forall k \in U$. *(Konstante skrivene u izrazu* $\Omega(\cdot)$ *sadrže samo ovisnosti o c i* κ*.)*

Da bismo dokazali ovaj teorem potrebne su nam određene tvrdnje iz kombinatorike i vjerojatnosti. Važan alat za dokazivanje teorema je *Kruskal − Katona* teorem koji nam daje gornju ogradu za broj trokuta u grafu za fiksan broj bridova.

Teorem 1.7.3 (Kruskal-Katona)**.** *Ako graf ima t trokuta i m bridova tada vrijedi* $t \leq m^{3/2}$.

Vjerojatnost da su i i j povezani je dobro aproksimirana izrazom $\hat{d}_i\hat{d}_j/2s$ kada je ovaj izraz značajno manji od 1. Međutim, možemo ga uvijek koristiti kao gornju ogradu što pokazuje sljedeća Slutnja:

Slutnja 1.7.4. *Neka imamo tri vrha* $i \neq j \neq k$. *Vjerojatnost da je formiran trokut* (i, j, k) *u CL grafu je:*

$$O\left(min\left(\frac{\hat{d}_i\hat{d}_j}{2s}, 1\right) min\left(\frac{\hat{d}_i\hat{d}_k}{2s}, 1\right) min\left(\frac{\hat{d}_j\hat{d}_k}{2s}, 1\right)\right).$$

Dokaz. Fiksirajmo par (i, j). Vjerojatnost da izaberemo brid (i, j) u CL graf G je $q_{ij} := 2\hat{d}_i\hat{d}_j/(2s)^2$. Primijetimo da je $q_{ij} < 0.5$. Vjerojatnost da brid nije izabran u prvih s odabira bridova je $(1 - q_{ij})^s \geq exp(-q_{ij}s/(1 - q_{ij}))$. Pretpostavimo da je $q_{ij}s \leq 0.5$ pa je izraz iz eksponenta strogo manji od 1. Aproksimacijom dobijemo $(1 - q_{ij})^s \geq 1 - q_{ij}s/(1 - q_{ij}) \geq 1 - 2q_{ij}s$. Stoga je vjerojatnost da je taj brid izabran najviše $2q_{ij}s = O(\hat{d}_i\hat{d}_j/2s)$. Kada je $q_{ij} \geq 0.5$ tada je $\hat{d}_i\hat{d}_j/2s = \Omega(1)$. Trivijalno je maksimalna vjerojatnost događaja da je brid izabran jednaka 1 pa je također u ovome slučaju vjerojatnost odabira brida jednaka $O(\hat{d}_i\hat{d}_j/2s)$. Prema tome, kombinirajući oba slučaja, dobijemo da je vjerojatnost odabira brida najviše $O\left(min\left(\frac{\hat{d}_i\hat{d}_j}{2s}, 1\right)\right)$. Događaji odabira bridova (i, j), (j, k) i (k, i) su nezavisni kada $s \to \infty$ pa slijedi tvrdnja Slutnje. □

Sada ćemo dokazati neke tvrdnje o očekivanom broju trokuta i stupnjeva vrha.

Slutnja 1.7.5. *Neka T označava očekivani broj trokuta. Tada postoje konstante* β *i* c'*, koje ovise samo o c i* κ *takve da vrijedi:*

1. $T \geq \beta \sum_i \hat{d}_i^2$
2. $\sum_i \hat{d}_i^2 \leq c' s^{3/2}$.

Dokaz. Po pretpostavci Teorema 1.7.2 vrijedi $T \geq (\kappa/3)\sum_i \binom{\hat{d}_i}{2}$. Za $\hat{d}_i > 1$ vrijedi da je $\binom{\hat{d}_i}{2} \geq \hat{d}_i^2/4$ (za velike $\hat{d}_i$, ovaj izraz je zapravo puno bliži $\hat{d}_i^2/2$. Stoga, $T \geq (\kappa/12)\sum_i \hat{d}_i^2$ i definiranjem β kao $\beta = \kappa/12$ dokazujemo prvu tvrdnju Slutnje.
Pretpostavimo da smo generirali slučajan CL graf. Neka su t broj trokuta i E broj bridova (obje su slučajne varijable). Po Teoremu 1.7.3 vrijedi $t \leq E^{3/2}$. Ako primijenimo očekivanje na ovaj izraz i uzmemo u obzir da je $E \leq s$ dobijemo da vrijedi $T \leq \mathbf{E}[E^{3/2}] \leq s^{3/2}$. Kombinirajući ovo s prvim dijelom tvrdnje dobijemo $\sum_i \hat{d}_i^2 \leq (1/\beta)s^{3/2}$. Ako definiramo $c' = 1/\beta$ dokazali smo drugu tvrdnju Slutnje. □

Sada dolazimo do dokaza Teorema 1.7.2:

Dokaz. Neka je b dovoljno velika konstanta i γ dovoljno mala. Neka je l najmanji indeks takav da vrijedi $\hat{d}_l > b\sqrt{s}$. Za uređenu trojku vrhova (i, j, k) neka je slučajna varijabla X_{ijk} indikator tvori li (i, j, k) trokut. Primijetimo da je $T = \mathbf{E}\left[\sum_{i<j<k} X_{ijk}\right]$ te iskoristimo gornju ogradu za $\mathbf{E}[X_{ijk}]$ iz Slutnje 1.7.4 pri čemu koristimo oznaku $\ll$ umjesto O notacije:

$$\begin{aligned}
\mathbf{E}\left[\sum_{i<j<k} X_{ijk}\right] &= \sum_{i<j<k} \mathbf{E}\left[X_{ijk}\right] \\
&\ll \sum_{i<j<k} \min\left(\frac{\hat{d}_i\hat{d}_j}{2s}, 1\right)\min\left(\frac{\hat{d}_i\hat{d}_k}{2s}, 1\right)\min\left(\frac{\hat{d}_j\hat{d}_k}{2s}, 1\right) \\
&\leq \sum_{i<j<k} \frac{\hat{d}_i\hat{d}_j}{2s}\frac{\hat{d}_i\hat{d}_k}{2s}\min\left(\frac{\hat{d}_j\hat{d}_k}{2s}, 1\right) \\
&\leq \sum_i \hat{d}_i^2 \sum_{j<k} \frac{\hat{d}_j\hat{d}_k}{4s^2}\min\left(\frac{\hat{d}_j\hat{d}_k}{2s}, 1\right).
\end{aligned}$$

Sada ćemo razdvojiti drugu sumu u zadnjem retku na slučajeve $j \leq l$ i $j > l$. U prvom slučaju ćemo ograničiti *min* izraz sa $\hat{d}_j\hat{d}_k/2s$, a u drugom slučaju sa 1. Primijetimo da je u drugoj sumi u izrazu ispod $k \geq l$ jer je $k > j$:

$$\begin{aligned}
\mathbf{E}\left[\sum_{i<j<k} X_{ijk}\right] &\ll \sum_i \hat{d}_i^2\left[\sum_{j,k:\, j\leq l} \frac{\hat{d}_j^2\hat{d}_k^2}{8s^3} + \sum_{j<k:\, j>l} \frac{\hat{d}_j\hat{d}_k}{4s^2}\right] \\
&\leq \left(\sum_i \hat{d}_i^2\right)\left[\frac{\left(\sum_{j\leq l}\hat{d}_j^2\right)\left(\sum_k \hat{d}_k^2\right)}{8s^3} + \frac{\left(\sum_{j\geq l}\hat{d}_j\right)^2}{4s^2}\right].
\end{aligned}$$

Po prvom dijelu Slutnje 1.7.5 vrijedi $T \geq \beta \sum_i \hat{d}_i^2$. Jednostavnosti radi, zamijenit ćemo sve nezavisne indekse s indeksom i te tada vrijedi:

$$\begin{aligned} \beta &\ll \frac{\left(\sum_i \hat{d}_i^2\right)\left(\sum_{i\leq l} \hat{d}_i^2\right)}{8s^3} + \frac{\left(\sum_{i\geq l} \hat{d}_i\right)^2}{4s^2} \\ \Longrightarrow \beta' &\leq \frac{\left(\sum_i \hat{d}_i^2\right)\left(\sum_{i\leq l} \hat{d}_i^2\right)}{8s^3} + \frac{\left(\sum_{i\geq l} \hat{d}_i\right)^2}{4s^2}. \end{aligned} \tag{1.6}$$

Koristimo β' da bismo označili konstantu koja dolazi iz << izraza. Prema Slutnji 1.7.5 vrijedi $\sum_i \hat{d}_i^2 \leq c' s^{3/2}$ i nadalje vrijedi $\sum_i \hat{d}_i^2 \geq b\sqrt{s} \sum_{i\geq l} \hat{d}_i$ (jer za $i \geq l$ vrijedi $\hat{d}_i \geq b\sqrt{s}$). Kombinirajući obje ograde dobijemo $\sum_{i\geq l} \hat{d}_i \leq (c'/b)s$. Ako koristimo ove ograde u Jednadžbi (1.6) i ako dobro namjestimo konstantu τ, dobijemo:

$$\begin{aligned} \beta' &\leq \frac{c' \sum_{i\leq l} \hat{d}_i^2}{8s^{3/2}} + (c'/2b)^2 \\ \Longrightarrow \sum_{i\leq l} \hat{d}_i^2 &\geq (8/c')(\beta' - (c'/2b)^2)s^{3/2} = \tau s^{3/2}. \end{aligned}$$

Primijetimo da ako u početku postavimo b dovoljno velikim, osigurat ćemo da je τ pozitivna konstanta. Neka je l' najmanji indeks takav da vrijedi $\hat{d}_l \geq \gamma\sqrt{m}$ i neka je $s' = \sum_{l'\leq i\leq l} \hat{d}_i$. Tada vrijedi:

$$\begin{aligned} \tau s^{3/2} &\leq \sum_{i\leq l} \hat{d}_i^2 \leq \sum_{i<l'} \hat{d}_i^2 + \sum_{l'\leq i\leq l} \hat{d}_i^2 \\ &\leq \gamma\sqrt{s} \sum_{i<l'} \hat{d}_i + b\sqrt{s} \sum_{l'\leq i\leq l} \hat{d}_i \\ &\leq \gamma(s - s')\sqrt{s} + bs'\sqrt{a} \\ \Longrightarrow \tau s &\leq s'(b - \gamma) + \gamma s \\ \Longrightarrow s' &\geq s(\tau - \gamma)/(b - \gamma) = \Omega(s). \end{aligned}$$

Opet, dovoljno mali γ osigurava da je s' pozitivan. Pokazali smo da vrhovi su indeksima iz intervala $[l', l]$ vezani uz barem $\Omega(s)$ bridova. Svi ti vrhovi su stupnja $\Theta(\sqrt{s})$ pa stoga ima $\Theta(\sqrt{s})$ takvih vrhova. □

1.8 BTER model

U ovom ću odjeljku predstaviti Block Two-Level Erdos-Renyi model baziran na ideji grafa koji sadrži ER zajednice. Prednost BTER modela je da ima strukturu zajednice u formi

gustih ER podgrafova i da dobro opisuje realne mreže. Prvo ću ukratko opisati model, a kasnije i napraviti detaljnije analize.
Prva faza BTER modela je izgradnja ER blokova u skladu sa specificiranom distribucijom stupnjeva vrhova. Iako BTER model omogućava izgradnju grafa neovisno o distribuciji vrhova, realne mreže su u pravilu *power-law* mreže, a kada je distribucija jednaka distribuciji teškog repa, onda BTER graf prirodno sadrži *scale-free* ER podgrafove. Internu povezanost ER grafova definira korisnik i može je prilagođavati u odnosu na promatrane podatke.
Druga faza BTER modela povezuje različite blokove. Pretpostavimo da svaki vrh nakon prve faze ima neki neiskorišteni višak stupnjeva. Primjerice, ako vrh i ima stupanj d_i u grafu G, a u ER bloku sudjeluje u d'_i bridova tada je višak stupnjeva jednak $d_i - d'_i$. U tom slučaju, koristimo CL model (koji se može smatrati težinskim oblikom ER modela) da formiramo veze koje povezuju zajednice od viška stupnjeva $d_i - d'_i$.

Detaljan opis BTER modela

BTER model se sastoji od međusobno povezanih zajednica. Intuitivno, konekcije na maloj udaljenosti (prva faza) su guste i imaju veliki koeficijent klasteriranja. Konekcije na velikoj udaljenosti (druga faza) su rijetke i imaju distribuciju teškog repa. Sada ćemo detaljno opisati svaku fazu izrade BTER modela.

Predobrada. U ovom koraku je svaki vrh stupnja 2 ili većeg dodijeljen zajednici. Pretpostavimo da je unaprijed zadana željena distribucija stupnjeva $\{d_i\}$ gdje d_i predstavlja stupanj vrha i. Ugrubo govoreći, $d + 1$ vrhova stupnja d dodjeljuje se zajednici tako da je podgraf induciran tim vrhovima gust. Sortiramo vrhove stupnja ≥ 2 uzlazno tako da imamo $d_1 \leq d_2 \leq d_3 \ldots$ Zamislimo da smo ovim redoslijedom sve vrhove stavili u jedan uređen skup tako da je vrh broj 1 prvi u nizu. S početka niza pročitamo stupanj d i uzmemo $d + 1$ članova niza koji čine jednu zajednicu. Zatim isti postupak ponavljamo dok u uređenom skupu ne ostane više vrhova, a na kraju će svi vrhovi biti particionirani po zajednicama. Primijetimo da ovim postupkom grupiramo vrhove istog stupnja osim par iznimaka gdje postoje križanja u stupnjevima (primjerice ako je zadnji vrh stupnja 2 grupiran s dva vrha stupnja 3) ili slučajeva gdje vrhova velikog stupnja ima jako malo. Ako distribucija stupnjeva ima distribuciju teškog repa, broj zajednica određenih veličina također imaju distribuciju teškog repa. Pošto je distribucija stupnjeva vrhova parametar s kojim ulazimo u model, ovaj korak je relativno jednostavan i prikazan na Slici 1.3. Sa $\mathcal{G}_r$ označavamo r-tu zajednicu, a sa u_i pripa dnost zajednici vrha i.

Faza 1. Lokalna struktura zajednica je modelirana kao ER graf što je prikazano na Slici 1.4. Povezanost svake zajednice je parametar modela, a promatrajući koeficijente klasteriranja nad stvarnim podacima možemo vidjeti da vrhovi manjeg stupnja imaju puno veće koeficijente klasteriranja od vrhova većeg stupnja. Ovo upućuje da su male zajednice

gušće povezane od velikih pa možemo na taj način namjestiti povezanost. Može se koristiti bilo koja formula, ali empirijski je pokazano da sljedeća formula koja označava vjerojatnost brida za zajednicu r funkcionira dobro u praksi:

$$\rho_r = \rho \left[1 - \eta \left(\frac{\log(\bar{d}_r + 1)}{\log(d_{max} + 1)} \right)^2 \right], \tag{1.7}$$

gdje su $\bar{d}_r = \min\{d_i \mid i \in \mathcal{G}_r\}$, d_{max} maksimalni stupanj u grafu, a ρ i η parametri izabrani tako da najbolje odgovaraju određenom grafu. Zašto je izabrana baš ova formula? Uočeno je u stvarnim mrežama da su koeficijenti klasteriranja nad vrhovima niskog stupnjeva relativno veliki. Kako se stupanj povećava, tako koeficijent klasteriranja opada i postiže najniže vrijednost za velike stupnjeve. Ovo opadanje se odvija u logaritamskoj skali pa zato koristimo formulu (1.7).

Faza 2. Globalna struktura je određena poveznica između zajednica. Primijenit ćemo CL model za *višak stupnjeva* svakog vrha, e_i, definirane na sljedeći način:

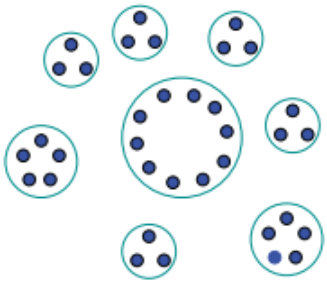
$$e_i = \begin{cases} 1, \text{ ako je } d_i = 1, \\ d_i - \rho_{u_i} \left(|\mathcal{G}_{u_i}| - 1 \right), \text{ inače,} \end{cases} \tag{1.8}$$

gdje $|\mathcal{G}_r$ označava veličinu zajednice r. Za dane e_i-eve, bridovi se generiraju birajući dvije krajnje točke na slučajan način. Specijalno, vjerojatnost odabira čvora i je $e_i / \sum_j e_j$. Faza 2 je prikazana Slikom 1.5. Potrebno je i napomenuti da su Slike 1.3, 1.4 i 1.5 preuzete iz [3]

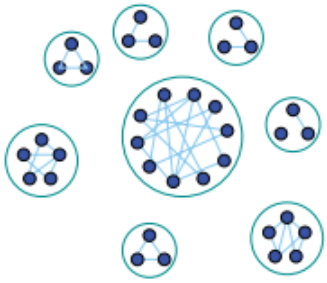
Asortativnost. Asortativnost je težnja čvorova mreže da se međusobno povežu sa sebi sličnima na određeni način. Faza 1 BTER modela ima veliku asortativnost, a Faza 2 nije asortativna. Parametar ρ_r kontrolira udio bridova Faze 1 u odnosu na Fazu 2. Općenito, BTER je prigodan model za mreže velikog koeficijenta klasteriranja jer to odgovara našoj teorijskoj podlozi koja pretpostavlja postojanje mnogo trokuta. Kao posljedica, BTER nije dobar model za grafove koji nisu asortativni.

1.9 Detalji implementacije BTER modela

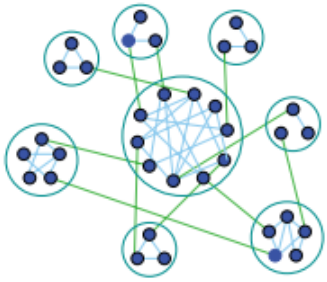
Pokazalo se da je u Fazi 1 dosta praktično postaviti parametar $\rho_r = 0$ za zadnju zajednicu pošto se ona sastoji od par vrhova koji su otpatci. Izračun za vrhove Faze 2 možemo podijeliti u 3 podfaze tako da posebno možemo raditi s vrhovima stupnja 1. Varijanca vrhova stupnja 1 je u CL modelu velika, tako da ćemo dio tih vrhova ostaviti sa strane i ručno ih pregledati. Neka w označava broj vrhova stupnja 1 i pretpostavimo da su vrhovi indeksirani uzlazno sortirano od vrhova najmanjeg do najvećeg stupnja. Početno je postavljeno da se 75% vrhova stupnja 1 promatra "ručno" iako je ovaj postotak podložan promjenama. Neka



Slika 1.3: Predobrada: distribucija čvorova po zajednicama



Slika 1.4: Faza 1: Poveznice unutar zajednica



Slika 1.5: Faza 2: Globalne poveznice između zajednica

$p = [0.75w]$ označava tu vrijednost gdje $[\cdot]$ označava najbliži cijeli broj. Ažuriramo e_i na sljedeći način:

$$e_i \leftarrow \begin{cases} 0, & 1 \leq i \leq p, \\ 1.10, & p+1 \leq i \leq w, \\ e_i, & \text{inače.} \end{cases}$$

Ovaj korak izbacuje prvih p vrhova iz CL dijela te također malo podiže vjerojatnost odabira brida za preostalih $w-p$ vrhova stupnja 1. Ova modifikacija pomaže izbalansirati činjenicu da neki vrhovi stupnja većeg od 1 vrlo vjerojatno postaju vrhovi stupnja 1 u završnom grafu pa želimo da u konačnici i vrhovi stupnja 1 postanu vrhovi većeg stupnja.
U Fazi 2a ostavimo sa strane $q \leq p$ vrhove stupnja 1 da se povežu s ostalim vrhovima stupnja 1. Ovu vrijednost korisnik može posebno specificirati ili se vrijednost definira kao:

$$q = 2 \left\lfloor \frac{p^2}{2 \sum_i d_i} \right\rfloor,$$

što predstavlja očekivani broj bridova CL modela koji spajaju po dva vrha stupnja 1. Ovo se može napraviti i na način da se slučajno upare takvi vrhovi. U našim eksperimentima ćemo koristiti $q = 0$.
U Fazi 2b, ručno spajamo preostalih $(p-q)$ vrhova s ostatkom grafa. Za svaki vrh stupnja 1, određujemo krajnju točku proporcijonalnu vrijednosti e_i.
U Fazi 2c konačno kreiramo CL model. Izmijenit ćemo očekivani broj stupnjeva na način da uračunamo duplikate i bridove korištene u Fazi 2b. Ažurirat ćemo e_i s vrijednošću ηe_i gdje je

$$\eta = 1 - 2\frac{p-q}{p-q+\sum_i e_i} + \beta,$$

gdje je β udio duplikata. U eksperimentima u [3] koristi se $\beta = 0.10$. Ukupan broj bridova generiranih u Fazi 2c (uključujući izbačene duplikate i bridove koji imaju jednak početni i završni vrh) je $\lfloor \sum_i e_i/2 \rfloor$.

Poglavlje 2

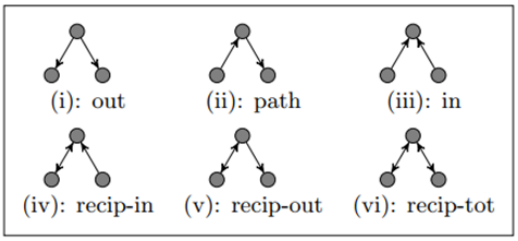
Usmjereni grafovi

Sada ćemo se fokusirati na usmjerene grafove i algoritme računanja broja usmjerenih klinova i trokuta. Iako je većina mreža koje su nam zanimljive usmjerenog tipa, razni algoritmi, zbog jednostavnosti i brzine, prvo pretvore takve mreže u neusmjerene grafove te zatim rade razne analize nad njima. Takav pristup se do sada pokazao dovoljno dobrim jer se redukcijom kompleksnosti grafa smanjila kompleksnost data mining problema, a pritom se i dalje otkrilo puno korisnih informacija o samoj mreži. Međutim, najčešći atribut u bridovima grafa je upravo usmjerenost, a mreže kao što su društvene, internetske ili transakcijske su po svojoj prirodi usmjerene mreže. Štoviše, usmjerene mreže često imaju značajan udio recipročnih bridova za koje se može pokazati da su važni u prepoznavanju širenja raznih virusa ili vijesti i pomažu nam bolje razumijeti sastav mreže. Također, trokuti koji sadrže recipročne i usmjerene bridove daju nam važne informacije o raznim podstrukturama mreže. S druge strane, izazovno je uspoređivati usmjerene mreže i dati smisao svim informacijama koje dobijemo analizom usmjerenih bridova. Naravno, i kompleksnost brojanja usmjerenih trokuta u grafu uvelike raste porastom broja čvorova mreže.

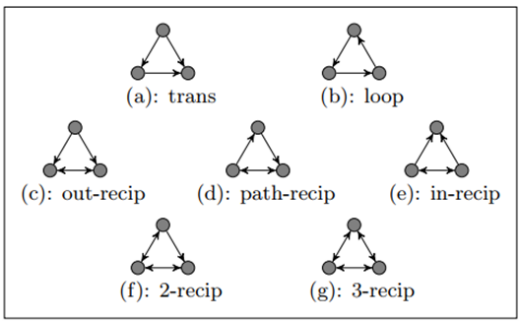
Notacija

Promatrat ćemo usmjereni graf u oznaci $G = (V, E)$ na način da prema [4] usmjerene bridove podijelimo u dva skupa: obične i recipročne. Recipročni brid je zapravo par bridova $\{(i, j), (j, i)\}$ koji ćemo spojiti u zajednički brid, a na slikama ćemo ga prikazivati kao dvostranu strelicu. Nadalje, definiramo recipročnost grafa, u oznaci τ, kao udio recipročnih bridova u grafu G. Klin i trokut se definiraju isto kao i u neusmjerenom slučaju, ali u usmjerenom slučaju postoji 6 različitih vrsta klinova i 7 trokuta što je detaljnije prikazano na Slikama 2.1 i 2.2 preuzetih iz [4]. Vrstu klina označit ćemo slovom ψ, a vrstu trokuta slovom τ i svakom tipu klina, odnosno trokuta, dodijelit ćemo imena koja su također prikazana na Slikama 2.1 i 2.2. Primijetimo da smo time napravili particiju svih vrsta us-

mjerenih klinova i trokuta. Pošto recipročne bridove gledamo zasebno, nećemo reći da npr. *recip-out* klin sadrži *out* klin.
Nadalje, za svaki vrh v u usmjerenom grafu imamo tri pripadajuća stupnja: ulazni, izlazni i recipročni stupanj. Oni se redom označavaju $d_v^{\leftarrow}$, $d_v^{\rightarrow}$ i $d_v^{\leftrightarrow}$, te vrijedi da je ukupni stupanj vrha v jednak $d_v = d_v^{\leftarrow} + d_v^{\rightarrow} + d_v^{\leftrightarrow}$.



Slika 2.1: Usmjereni klinovi



Slika 2.2: Usmjereni trokuti

Na Slikama 2.1 i 2.2, preuzetih iz [4], se u prvom redu nalaze klinovi, odnosno trokuti, koji su sastavljeni isključivo od nerecipročnih bridova. Postoji samo 3 različita klina i 2 različita trokuta takvog tipa. Trokuti (b), (d), (f) i (g) sadrže ciklus i to redom imaju 0, 1, 2 i 3 recipročna brida. Također, možemo primijetiti da različiti trokuti sadrže različite vrste klinova, a tu povezanost možemo prikazati funkcijom $\chi(\psi, \tau)$ koju definiramo kao broj klinova tipa ψ u trokutu tipa τ. Popis svih vrijednosti funkcije χ nalazi se u Tablici 2.2 koja sadrži informaciju koliko je klinova tipa i sadržano u trokutu tipa j, za svaki $i \in \{(i), (ii), (iii), (iv), (v), (vi)\}$ i svaki $j \in \{(a), (b), (c), (d), (e), (f), (g)\}$. Tablica ukupno ima 15 pozitivnih vrijednosti.
Nadalje, za svaki vrh v možemo računati vrijednost funkcije $W_{v,\psi}$ koja se definira kao broj klinova tipa ψ centriranih u vrhu v. Prikaz svih vrijednosti funkcije $W_{v,\psi}$ nalaze se u Tablici 2.1. Naravno, vrijednosti funkcije W ovise o stupnjevima vrha v: $d_v^{\leftarrow}$, $d_v^{\rightarrow}$ i $d_v^{\leftrightarrow}$.

ψ	i	ii	iii	iv	v	vi
$W_{v,\psi}$	$\binom{d_v^{\rightarrow}}{2}$	$d_v^{\leftarrow} d_v^{\rightarrow}$	$\binom{d_v^{\leftarrow}}{2}$	$d_v^{\leftarrow} d_v^{\leftrightarrow}$	$d_v^{\rightarrow} d_v^{\leftrightarrow}$	$\binom{d_v^{\leftrightarrow}}{2}$

Tablica 2.1: Broj klinova tipa ψ centriranih u vrhu v

τ \ ψ	i	ii	iii	iv	v	vi
a	1	1	1			
b		3				
c	1				2	
d		1		1	1	
e			1	2		
f				1	1	1
g						3

Tablica 2.2: Zastupljenost svake vrste klina ψ u različitim vrstama trokuta τ u oznaci $\kappa(\psi, \tau)$

2.1 (ψ, τ) - zatvaranje

Tranzitivnost κ se u neusmjerenom grafu definira kao udio klinova koji sudjeluju u trokutima, $\kappa = \frac{|3T|}{|W|}$, gdje je T skup trokuta, a W skup klinova. Također, u neusmjerenom grafu je klin zatvoren ako tvori trokut, a otvoren je u suprotnom slučaju. U usmjerenom grafu ćemo reći da je ψ-klin τ-zatvoren ako je klin dio trokuta tipa τ. (ψ, τ) **- zatvaranje**, u oznaci $\kappa_{\psi,\tau}$, definira se kao omjer ψ-klinova koji su τ-zatvoreni. Formalno, neka je W_ψ skup svih ψ-klinova i T_τ skup svih τ-trokuta. Tada je:

$$\kappa_{\psi,\tau} = \frac{\chi(\psi, \tau)\,|T_\tau|}{\left|W_\psi\right|}.$$

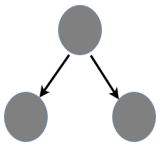
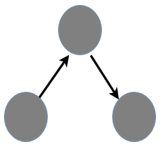
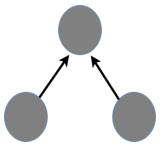
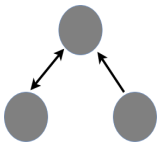
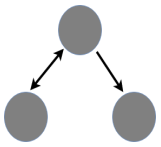
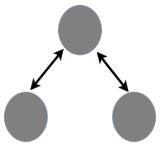
Primijetimo da je broj ψ-klinova u τ-trokutima jednak $\chi(\psi, \tau)\,|T_\tau|$. Također, ako τ-trokut ne sadrži klin tipa ψ, tada je vrijednost $\kappa_{\psi,\tau}$ jednaka nuli jer je $\chi(\psi, \tau) = 0$. Kao što je spomenuto ranije u radu, postoji 15 netrivijalnih (ψ, τ) - zatvaranja.

Nul-model za (ψ, τ) - zatvaranja

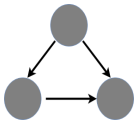
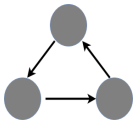
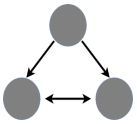
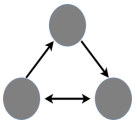
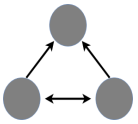
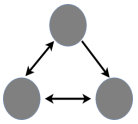
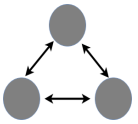
U ovom ćemo odjeljku predstaviti nul-hipotezu baziranu na dodjeljivanju tipa svakom bridu isključivo na osnovi recipročnosti grafa. Prvo ćemo graf pretvoriti u neusmjereni te zatim dodati smjer i recipročnost na sljedeći način:

1. ako je (u, v) neusmjeren brid, pretvorit ćemo ga u recipročni s vjerojatnošću r,
2. usmjerit ćemo ga iz vrha u u vrh v s vjerojatnošću $(1 - r)/2$,
3. usmjerit ćemo ga iz vrha v u vrh u s vjerojatnošću $(1 - r)/2$.

S ova tri slučaja dobili smo potpuni sustav događaja u kojem je suma vjerojatnosti svih događaja jednaka 1. Prema ovakvom modelu, jednostavno možemo izračunati kolika je vjerojatnost da neusmjereni klinovi/trokuti pripadaju određenoj vrsti klinova/trokuta. Rezultati tih izračuna prikazani su u Tablicama 2.3 i 2.4.

					
$\frac{(1-r)^2}{4}$	$\frac{(1-r)^2}{2}$	$\frac{(1-r)^2}{4}$	$r(1-r)$	$r(1-r)$	r^2

Tablica 2.3: Vjerojatnost da neusmjereni klin postane određena vrsta usmjerenog klina

						
$\frac{3(1-r)^3}{4}$	$\frac{(1-r)^3}{4}$	$\frac{3(1-r)^2}{4}$	$\frac{3(1-r)^2}{2}$	$\frac{3(1-r)^2}{4}$	$3r^2(1-r)$	r^3

Tablica 2.4: Vjerojatnost da neusmjereni trokut postane određena vrsta usmjerenog trokuta

U Tablici 2.4 možemo na primjer usporediti očekivane udjele zadnjih dvaju trokuta *2-recip* i *3-recip* te bi prema našem nul-modelu *2-recip* trokuti trebali biti zastupljeniji ako pretpostavimo da je recipročnost $r < 0.75$. Također, očekivani udio *trans*-trokuta je prema nul-modelu tri puta veći od udjela *loop*-trokuta.

2.2 Usmjereni trokuti

U prijašnjim odjeljcima smo prepoznali važnost računanja direktnih zatvaranja u usmjerenim grafovima, a sada ćemo pokazati kako na efikasan način doći do tih rezultata. Opisat ćemo aproksimacijske algoritme za procjenu različitih koeficijenata klasteriranja i ukupnog broja τ-trokuta tako da prilagodimo algoritam *klin-uzorkovanja* koji smo koristili nad neusmjerenim grafovima.
Započnimo s osnovnom notacijom, definirajmo prvo sedam različitih skupova $W_\psi(\tau)$ kao:

$$W_\psi(\tau) = \{w \in W_\psi \mid \text{w je } \tau - \text{zatvoren}\}.$$

Primijetimo da je $\kappa_{\psi,\tau} = \left|W_\psi(\tau)\right| / \left|W_\psi\right|$ te se ovaj omjer može procijeniti sljedećim algoritmom:

Algoritam 5 Procjena $\kappa_{\psi,\tau}$ mjere ((ψ,τ)-zatvaranja)

1. Uzmi k uniformno slučajnih ψ-klinova (s ponavljanjima)
2. Izračunaj k', broj τ-zatvorenih klinova iz uzorka dobivenog u koraku 1
3. Izračunaj procjenu $\hat{\kappa}_{\psi,\tau} = k'/k$.

Sljedeći teorem nam govori da je $\hat{\kappa}_{\psi,\tau}$ dovoljno dobra procjena za $\kappa_{\psi,\tau}$:

Teorem 2.2.1. *Neka su $\epsilon, \delta > 0$ i neka je $k = \left\lceil 0.5\epsilon^{-2} ln(2/\delta)\right\rceil$. Tada vrijedi:*

$$P\left\{\left|\hat{\kappa}_{\psi,\tau} - \kappa_{\psi,\tau}\right| \geq \epsilon\right\} \leq \delta.$$

Dokaz. Definirajmo prvo slučajnu varijablu X_i koja je indikator da je i-ti ψ-klin τ-zatvoren:

$$X_i(w) = \begin{cases} 1, & \text{ako je } \psi\text{-klin } w\ \tau - \text{zatvoren} \\ 0, & \text{ako je } \psi\text{-klin } w\ \tau - \text{otvoren.} \end{cases}$$

Primijetimo da je $E[X_i] = \kappa_{\psi,\tau}$ pa je $E[\sum_{i\leq k} X_i] = k\kappa_{\psi,\tau}$. Pošto je $\hat{\kappa}_{\psi,\tau} = \sum_{i\leq k} /k$, izraz $\left|\hat{\kappa}_{\psi,\tau} - \kappa_{\psi,\tau}\right| \geq \epsilon$ jednak je izrazu $\left|\sum_{i\leq k} X_i - E[\sum_{i\leq k} X_i]\right| \geq k\epsilon$. Prema HOEFFDING teoremu 1.1.1 slijedi da je vjerojatnost ovog događaja najviše $2\exp\left(-2\epsilon^2 k^3\right) \leq 2\exp\left(-2\epsilon^2 k\right) \leq \delta$. □

Direktan korolar ovog teorema daje nam granice za procjenu broja trokuta na način da nejednakost iz teorema 2.2.1 pomnožimo sa $|W_\psi|/\chi_(\psi,\tau)$ i tako dobijemo $|T_\tau| = \kappa_{\psi,\tau}|W_\psi|/\chi(\psi,\tau)$.

Korolar 2.2.2. *Fiksirajmo vrstu klina ψ i vrstu trokuta τ takve da je $\chi(\psi,\tau) \neq 0$. Neka je $k = \left\lceil 0.5\epsilon^{-2} ln(2/\delta)\right\rceil$ i neka je $\hat{T} = \hat{\kappa}_{\psi,\tau}|W_\psi|/\chi(\psi,\tau)$. Tada vrijedi:*

$$P\left[\left|\hat{T} - |T_\tau|\right| \geq \frac{\epsilon|W_\psi|}{\chi\left(\psi,\tau\right)}\right] \leq \delta.$$

Primijetimo ovdje par bitnih stvari:

1. Za istu veličinu uzoraka možemo koristiti različite vrste klinova za računanje broja trokuta iste vrste $|T_\tau|$,

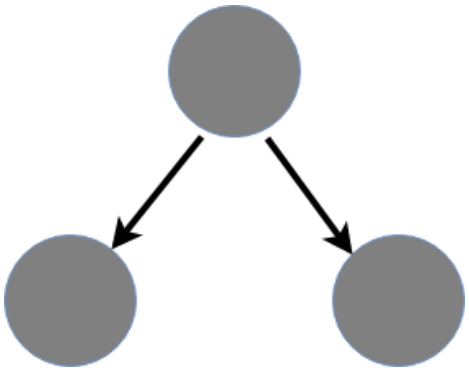
2. Za klin tipa ψ, greška iz korolara je proporcijalna izrazu $|W_\psi|/\chi(\psi,\tau)$ pa korištenjem vrsta klinova koje se pojavljuju rjeđe dobivamo pouzdaniju aproksimaciju za određeni tip trokuta,

3. Samo su četiri vrste klinova dovoljne za računanje ukupnog broja trokuta svake vrste τ (radi se primjerice o klinovima: *path, recip-out, recip-in i recip-tot* sa Slike 2.1, ali isto tako možemo uzeti npr. klinove: *out, path, in i recip-tot* ili bilo koju drugu kombinaciju sa Slike 2.1 tako da su sve vrste trokuta pokrivene).

2.3 Uniformno uzorkovanje usmjerenih klinova

Sada ćemo opisati proceduru uniformnog uzorkovanja usmjerenih klinova. Usmjerene klinove općenito možemo podijeliti u dvije skupine: homogene, one koji sadrže samo jednu vrstu bridova: *out, path, in i recip-tot*, i heterogene, koji sadrže različite vrste bridova: *recip-in, recip-out*. Za sve njih ćemo predstaviti algoritam uniformnog uzorkovanja pomoću kojega ćemo doći do procjene ukupnog broja usmjerenih trokuta bilo koje vrste:

(i) OUT klinovi

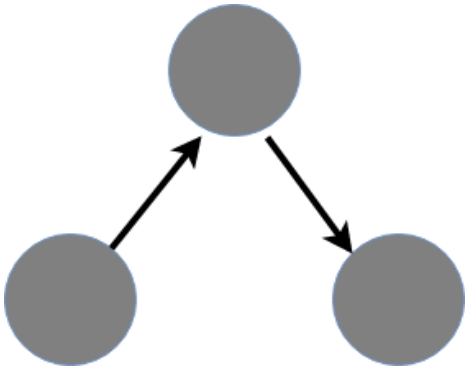


Slika 2.3: Prikaz OUT klina

Odredimo prvo vjerojatnosnu distribuciju $\{p_v\}$ za svaki vrh $v \in V$. Neka je $p_v = |W_{v,(i)}|/|W_{(i)}| = \binom{d_v^{\rightarrow}}{2}/|W_{(i)}|$, gdje je $\binom{d_v^{\rightarrow}}{2}$ broj *out* klinova centriranih u vrhu v, $d_v^{\rightarrow}$ pripadni izlazni stupanj vrha v, te $|W_{(i)}|$ ukupan broj *out* klinova u grafu G. Primijetimo prvo da je $\sum_{v\in V} p_v = 1$ pa je $\{p_v\}$ zaista vjerojatnosna distribucija u grafu G. Nadalje, primijetimo da je za računanje distribucije $\{p_v\}$ potrebno znati izlazni stupanj $d_v^{\rightarrow}$ svakog vrha v te ukupan broj *out* klinova

u grafu G što u velikim grafovima i nije jednostavno izračunati, ali to je pretpostavka bez koje ne možemo krenuti dalje.

(ii) PATH klinovi



Slika 2.4: Prikaz PATH klina

Opet određujemo vjerojatnosnu distribuciju $\{p_v\}$, ali sada na način da je za svaki $v \in V$ $p_v = |W_{v,(ii)}|/|W_{(ii)}| = d_v^{\leftarrow} d_v^{\rightarrow}/|W_{(ii)}|$, gdje je $d_v^{\leftarrow} d_v^{\rightarrow}$ broj *path* klinova centriranih u vrhu v, $d_v^{\rightarrow}$ i $d_v^{\leftarrow}$ pripadni izlazni, odnosno ulazni stupnjevi vrha v, te $|W_{(ii)}|$ ukupan broj *path* klinova u grafu G.

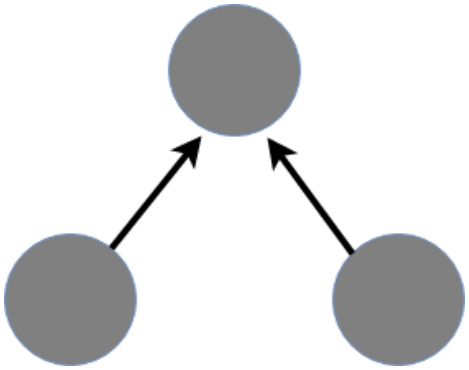
(iii) IN klinovi

U ovom slučaju je za svaki vrh $v \in V$ $p_v = |W_{v,(iii)}|/|W_{(iii)}| = \binom{d_v^{\leftarrow}}{2}/|W_{(iii)}|$, gdje je $\binom{d_v^{\leftarrow}}{2}$ broj *in* klinova centriranih u vrhu v, $d_v^{\leftarrow}$ pripadni ulazni stupanj vrha v, te $|W_{(iii)}|$ ukupan broj *in* klinova u grafu G.

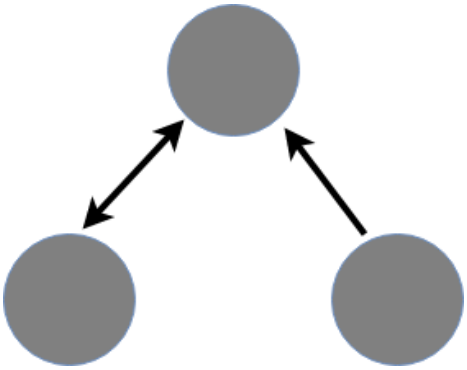
(iv) RECIP-IN klinovi

U *recip-in* slučaju je za svako $v \in V$ $p_v = |W_{v,(iv)}|/|W_{(iv)}| = d_v^{\leftarrow} d_v^{\leftrightarrow}/|W_{(iv)}|$, gdje je $d_v^{\leftarrow} d_v^{\leftrightarrow}$ broj *recip-in* klinova centriranih u vrhu v, $d_v^{\leftarrow}$ i $d_v^{\leftrightarrow}$ pripadni ulazni, odnosno recipročni stupnjevi vrha v, te $|W_{(iv)}|$ ukupan broj *recip-in* klinova u grafu G.

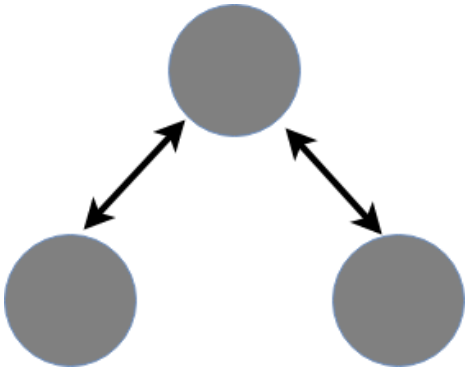
U *recip-out* slučaju je za svako $v \in V$ $p_v = |W_{v,(v)}|/|W_{(v)}| = d_v^{\rightarrow} d_v^{\leftrightarrow}/|W_{(v)}|$, gdje je $d_v^{\rightarrow} d_v^{\leftrightarrow}$ broj *recip-out* klinova centriranih u vrhu v, $d_v^{\rightarrow}$ i $d_v^{\leftrightarrow}$ pripadni izlazni, odnosno recipročni stupnjevi vrha v, te $|W_{(v)}|$ ukupan broj *recip-in* klinova u grafu G.



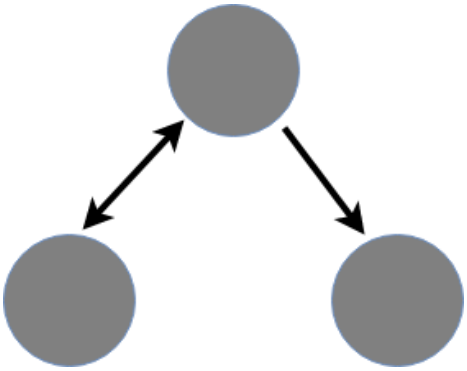
Slika 2.5: Prikaz IN klina



Slika 2.6: Prikaz RECIP-IN klina



Slika 2.7: Prikaz RECIP-TOT klina



Slika 2.8: Prikaz RECIP-OUT klina

(vi) RECIP-TOT klinovi

U ovom slučaju je za svaki vrh $v \in V$ $p_v = |W_{v,(vi)}|/|W_{(vi)}| = \binom{d_v^{\leftrightarrow}}{2}/|W_{(vi)}|$, gdje je $\binom{d_v^{\leftrightarrow}}{2}$ broj *in* klinova centriranih u vrhu v, $d_v^{\leftrightarrow}$ pripadni recipročni stupanj vrha v, te $|W_{(vi)}|$ ukupan broj *in* klinova u grafu G.
Za svaki od prethodno navedenih slučajeva vrijedi sljedeći algoritam za procjenu direktnih zatvaranja i broja usmjerenih trokuta u grafu G:

Algoritam 6 Uniformno uzorkovanje usmjerenih klinova i računanje zatvaranja $\kappa(\psi, \tau)$

1. Uzmi k uniformno slučajnih vrhova (s mogućim ponavljanjima) prema distribuciji $\{p_v\}$, gdje je $p_v = |W_{v,\psi}|/|W_\psi|$.
2. Za svaki izabrani vrh v, izaberi uniformno slučajni ψ klin centriran u vrhu v. Ako su oba brida iste vrste (*out, in i recip-tot* klinovi) onda unifomno uzorkujemo bez ponavljanja. U suprotnom slučaju uzorkujemo s ponavljanjem jer su dvije vrste brida različite i međusobno nezavisne.
3. Izračunaj k', broj τ-zatvorenih klinova iz uzorka dobivenog u koraku 2.
4. Izračunaj procjenu $\hat{\kappa}_{\psi,\tau} = k'/k$ za $\kappa_{\psi,\tau}$.
5. Izračunaj procjenu $\hat{T}_\tau = \hat{\kappa}_{\psi,\tau}|W_\psi|/\chi(\psi, \tau)$ za T_τ.

Sada slijedi primjer koda u programskom jeziku R. Kod je dosta složeniji od neusmjerenih slučajeva i najveći problem je bio odvojiti stupnjeve recipročnosti vrhova $d^{\leftrightarrow}$ od ulaznih i izlaznih stupnjeva jer paket *igraph* ne podržava tako nešto. Prva opcija je bila da se ručno pretražuje cijeli graf i svakom vrhu dodjeljuje odgovarajući recipročni stupanj, međutim to se čak i za najmanji graf e-mailova nije dalo izračunati u par sati na 32-bitnom kompjuteru s 2GB RAM-a i procesorom od 2.20GHz. Stoga se uvrstio u kod još jedan R paket *slam* koji omogućava izračun recipročnih stupnjeva preko matrice sličnosti. Slijedi prvo kod za računanje recipročnih stupnjeva, a zatim i sam algoritam klin uzorkovanja za procjenu $\kappa(\psi, \tau)$ zatvaranja:

```
##########################################
## izracun reciprocnih stupnjeva
#####################################
install.packages("slam")
library(slam)
email <- read.table("Email-EuAll.txt", header = TRUE)
mail_dir <- graph.data.frame (email, directed=TRUE)
adj_matrix <- as_adjacency_matrix(mail_dir)
d_v_mail_recip <-
row_sums(((adj_matrix+t(adj_matrix))>1)*1)-
diag(adj_matrix)  ##reciprocni stupnjevi
susjedi_recip <- ((adj_matrix+t(adj_matrix))>1)*1
```

Sada slijedi i kod za sam algoritam klin uzorkovanja u obliku funkcije koja prima sljedeće parametre:

- veličinu uzorka,
- graf u obliku prilagođenom *igraph* paketu u R-u,
- naziv klina za koji provodimo algoritam,
- skup svih vrhova grafa u obliku vektora,
- vjerojatnosnu distribuciju odabira vrhova u oznaci p_v,
- distribuciju recipročnih stupnjeva vrhova izračunatu u prethodno navedenom kodu.

```
#Racunanje zatvaranja kappa(psi,tau)
########################
kappa_dir_procjena <-
function (k, g, psi, V, p_v, susjedi_recip){
  start.time <- Sys.time()
  cnt <- 0
  n <- length(V)
  if (k > n){
    k <- n
  }
  uzorak <- sample(V, size=k, replace=TRUE, prob=p_v)
  for (i in uzorak){
    if(psi == "out"){
      susjedi<-sample(unique(neighbors(g, i,
      mode="out")),2, replace=FALSE)
      if(are.connected(g,susjedi[1], susjedi[2])
      | are.connected(g,susjedi[1], susjedi[2])){
        cnt <- cnt +1
      }

    if(psi == "path"){
      susjedi<-c(sample(unique(neighbors(g, i,
      mode="in")),1,
      replace=FALSE),sample(unique(neighbors(g,
      i, mode="out")),1, replace=FALSE))
      if(are.connected(g,susjedi[1], susjedi[2])
      | are.connected(g,susjedi[1], susjedi[2])){
        cnt <- cnt +1
      }
```

```
}

if(psi == "in"){
  susjedi<-sample(unique(neighbors(g, i,
  mode="in")),2, replace=TRUE)
  if(are.connected(g,susjedi[1], susjedi[2])
  | are.connected(g,susjedi[1], susjedi[2])){
    cnt <- cnt +1
  }
}

if(psi == "recip_in"){
  susjedi1 <- as.numeric(neighbors(g, i, mode="in"))
  susjedi_tmp <- susjedi_recip[as.character(i),]
  susjedi2 <- as.numeric(names(susjedi_tmp
  [susjedi_tmp==1]))
  susjedi <-
  sample(unique(c(susjedi1,susjedi2)),2,replace=TRUE)
  if(are.connected(g,susjedi[1], susjedi[2])
  | are.connected(g,susjedi[1], susjedi[2])){
    cnt <- cnt +1
  }
}

if(psi == "recip_out"){
  susjedi1 <- as.numeric(neighbors(g, i, mode="out"))
  susjedi_tmp <- susjedi_recip[as.character(i),]
  susjedi2 <-
  as.numeric(names(susjedi_tmp[susjedi_tmp==1]))
  susjedi <- sample(unique(c(susjedi1,susjedi2)),
  2,replace=TRUE)
  if(are.connected(g,susjedi[1], susjedi[2])
  | are.connected(g,susjedi[1], susjedi[2])){
    cnt <- cnt +1
  }
}

if(psi == "recip_tot"){
  susjedi_tmp <- susjedi_recip[as.character(i),]
```

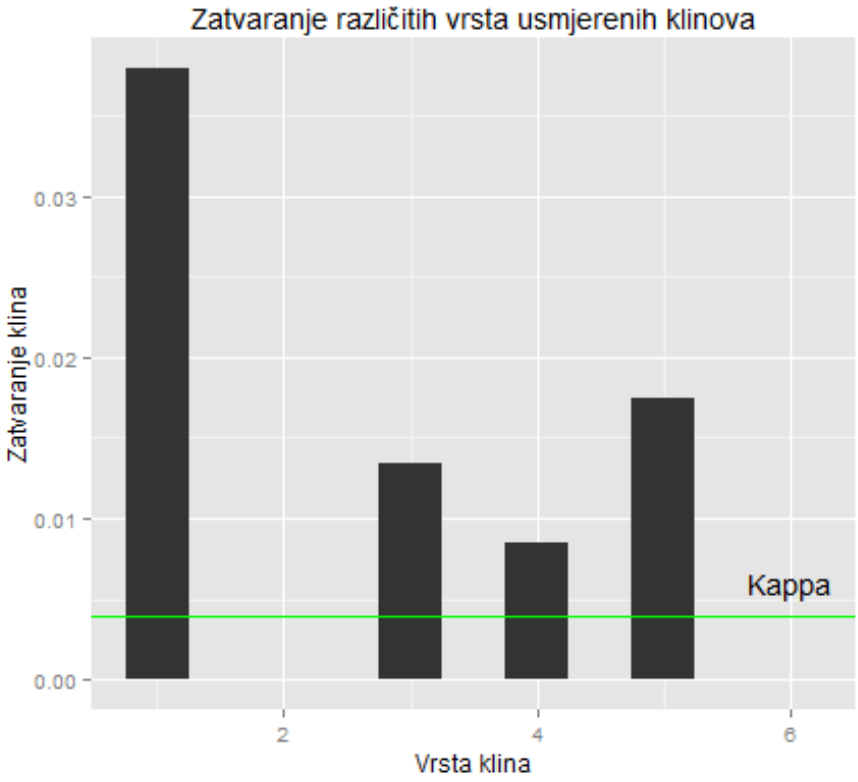
```
        susjedi2 <- as.numeric(names(susjedi_tmp
        [susjedi_tmp==1]))
        susjedi <- sample(susjedi2,2,replace=TRUE)
        if(are.connected(g,susjedi[1], susjedi[2])
        | are.connected(g,susjedi[1], susjedi[2])){
          cnt <- cnt +1
        }
      }

      }
    }
    end.time <- Sys.time()
    time.taken <- end.time - start.time
    return (list(cnt/k, time.taken))
}
```

Primjeri

Algoritam procjene zatvaranja $\kappa(\psi, \tau)$ isproban je na najmanjem skupu podataka zbog problema s performansama računala. Radi se o mreži e-mailova i pritom se promatrao udio svake vrste klina u trokutima. Vidjeli smo da je tranzitivnost κ neusmjerene mreže e-mailova jako mala i iznosi samo 0.024, stoga je za očekivati i da je udio zatvaranja usmjerenih klinova jako mali. Zato je u ovom slučaju algoritam malo modificiran i ne gledaju se zatvaranja usmjerenih klinova za svaku vrstu trokuta, nego općenito da sudjeluju u tvorbi bilo kojeg trokuta.
Na Slici 2.9 možemo vidjeti rezultate algoritma izgeneriranim nad uzorkom od 380 slučajno odabranih čvorova. Zelenom linijom označen je iznos tranzitivnosti u neusmjerenom slučaju, dok stupci redom predstavljaju klinove:

1. out klin,
2. path klin,
3. in klin,
4. recip-in klin,
5. recip-out klin,
6. recip-tot klin.

Mreža sama po sebi ima malu tranzitivnost, a samim time i mali broj trokuta pa i udio zatvaranja usmjerenih klinova nije ništa bolji. Iako u ovom slučaju recipročni bridovi ne sudjeluju češće u tvorbi trokuta, prema [4] su jako bitni i dobri su indikatori tvorbe trokuta, a samim time i podstruktura opisanih BTER modelom. Također, ovdje treba uzeti u obzir i pouzdanost algoritma jer za 380 vrhova ima 99% pouzdanu grešku u iznosu od $\epsilon < 0.1$. Vidimo da je greška procjene tranzitivnosti veća od same tranzitivnosti, stoga ni ne možemo donijeti neke velike zaključke o ovoj mreži.



Slika 2.9: Prikaz zatvaranja usmjerenih klinova u mreži email-EuAll.

Bibliografija

[1] *R tutorial*. https://www.tutorialspoint.com/r/.

[2] Leskovec, Jure i Andrej Krevl: *SNAP Datasets: Stanford Large Network Dataset Collection*. http://snap.stanford.edu/data, lipanj 2014.

[3] Seshadhri, C., Tamara G. Kolda i Ali Pinar: *Community Structure and Scale-free Collections of Erdős-Rényi Graphs*. Physical Review E, 85(5), May 2012.

[4] Seshadhri, C., Ali Pinar, Nurcan Durak i Tamara G. Kolda: *The importance of directed triangles with reciprocity: patterns and algorithms*. CoRR, abs/1302.6220, 2013. http://arxiv.org/abs/1302.6220.

[5] Seshadhri, C., Ali Pinar i Tamara G. Kolda: *Triadic Measures on Graphs: The Power of Wedge Sampling*. U *SDM13: Proceedings of the 2013 SIAM International Conference on Data Mining*, stranice 10–18, 2013.

[6] Seshadhri, C., Ali Pinar i Tamara G. Kolda: *Wedge Sampling for Computing Clustering Coefficients and Triangle Counts on Large Graphs*. Statistical Analysis and Data Mining, 7(4):294–307, August 2014.

Sažetak

Ukratko, u ovom su se radu predstavili pojmovi klinova i trokuta u neusmjerenim i usmjerenim mrežama, koji su temelji prepoznavanja skrivenih mrežnih podstruktura koje nazivamo *zajednicama*. Zajednica se definira kao podstruktura mreže koja je unutar sebe gusto povezana, a trokuti su dobri indikatori povezanosti jer predstavljaju homofilnost i tranzitivnost, svojstva koja nam govore da ljudi postaju prijatelji s onima sličnima sebi, što je korisno primjerice kod istraživanja prijevara u bankovnim transakcijama. Međutim, stvarne interakcijske mreže sadrže milijune pa čak i milijarde čvorova te nije jednostavno izračunati matematičke mjere vezane uz trokute na tako velikim mrežama. Stoga je u ovom radu predstavljen i *algoritam klin uzorkovanja* pomoću kojega se s relativno malim uzorkom slučajno odabranih čvorova mogu dovoljno dobro izračunati ukupan broj trokuta u cijeloj mreži te mjere koje su usko povezane s trokutima i klinovima. Primjerice, da algoritam postigne točnost od $99,9\%$ potrebno je samo 380 slučajno odabranih klinova neovisno o ukupnoj veličini mreže.

Također, predstavljen je i *Block Two-Level Erdos-Renyi* model mreže (BTER) koji se sastoji od međusobno povezanih zajednica različitih veličina. Pokazalo se da takav model ima slična svojstva kao i stvarne interakcijske mreže kod kojih stupnjevi čvorova prate distribuciju teškog repa, odnosno veliki broj čvorova je malog stupnja, a mali je broj čvorova velikog stupnja.

BTER model je u ovom radu predstavljen samo za neusmjerene grafove te bi bilo zanimljivo vidjeti kako bi se mogao primijeniti i na grafove uzimajući u obzir njihovu usmjerenost. Dodatno, iako se na primjeru vezanom za usmjerene grafove nije vidjela važnost recipročnih bridova u tvorbi trokuta, recipročnost je svojstvo koje se pokazalo bitnim kod grafova visoke tranzitivnosti tako da bi to moglo biti jedno od važnijih svojstava za buduća istraživanja primjene BTER modela na usmjerenim grafovima.

Summary

In this work we presented concepts of wedges and triangles in undirected and directed graphs, which are the basis for the recognition of hidden network substructures called *communities*. A community is defined as a substructure of a network that is internally well connected. Triangles are good correlation indicators because they represent homophily and transitivity - indicators that tell us that people become friends with those similar to themselves, which is useful, for example, in fraud detection of banking transactions. However, actual interaction networks contain millions, even billions of nodes, and it is hard to calculate mathematical measures associated with triangles on such large networks. Therefore, we also presented the *wedge sampling* algorithm that can sufficiently accurately calculate total number of triangles on a relatively small sample of randomly selected nodes. For example, it only takes 380 randomly selected wedges for the algorithm to be $99,9\%$ correct.

In addition, *Block Two-Level Erdos-Renyi* model network (BTER) is introduced as a model that has a community structure in the form of dense subgraphs. It has been shown that such model matches well with real-world graphs that are idealized as power laws - in other words, their degree distribution is heavy tailed.

The BTER model is presented in this paper only for undirected graphs and it would be interesting to see how it can be applied for directed graphs. Additionally, the relevance of reciprocal edges in the triangle formation is noted in literature related to directed graphs. This could be one of the most important features for the future research on the BTER model for directed graphs.

Životopis

Rođen sam 9.1.1993. u Zagrebu gdje sam odrastao i trenutačno živim. Završio sam Osnovnu školu Ante Kovačića u Zagrebu, 2007. godine upisujem Gimnaziju Lucijan Vranjanin u Zagrebu, pridoslovno-matematički smjer, te 2011. godine upisujem Prirodoslovno-matematički fakultet u Zagrebu, smjer Matematika. 2014. godine dobivam titulu univ.bacc.math te tada upisujem Diplomski studij Matematičke statistike na Prirodoslovno-matematičkom fakultetu u Zagrebu.
Od 2013. do 2016. aktivni sam član volonterske udruge eSTUDENT, gdje sam 2014./15. bio voditelj studentskog data mining natjecanja Mozgalo, a 2015./16. koordinator svih natjecanja u organizaciji eSTUDENTa. Od kolovoza 2016. godine radim u Zagrebačkoj banci u odjelu Razvoj poslovne inteligencije na poziciji Analitičara za izvještajne baze.