Mere kompleksnih mrež
(angl. Network Statistics)

Seminarska naloga pri predmetu Izbrana poglavja iz diskretne matematike

Ajda Pirnat, Julia Cafnik in Živa Mitar

Fakulteta za matematiko in fiziko

April 2012

Uvod

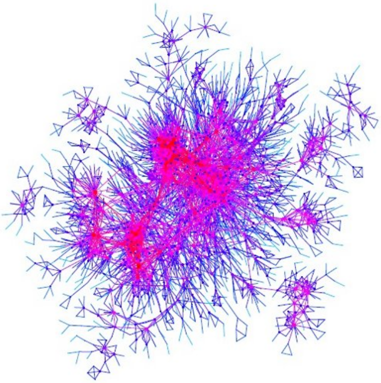
- Zakaj potrebujemo mere za omrežja?
- Kakšne naj bodo bistvene lastnosti omrežnih mer?
  - Opisujejo naj bistvene značilnosti omrežja.
  - Razlikujejo naj med različnimi razredi omrežij.
  - Uporabna naj bodo v algoritmih in aplikacijah.

Porazdelitev stopenj vozlišč

- To je najpogostejša in računsko nezahtevna mera.
- V naključnem grafu $G_{n,p}$ je stopnja vozlišča porazdeljena:
  - Binomsko - za majhno število vozlišč: $\binom{n-1}{k}p^k(1-p)^{(n-1-k)}$
  - Poissonovo - za veliko število vozlišč: $\frac{(np)^k}{k!}e^{-np}$
- V realnosti so vozlišča pogosto porazdeljena po potenčnem zakonu, tj. ck^{γ} za $\gamma > 0$ in $c > 0$.

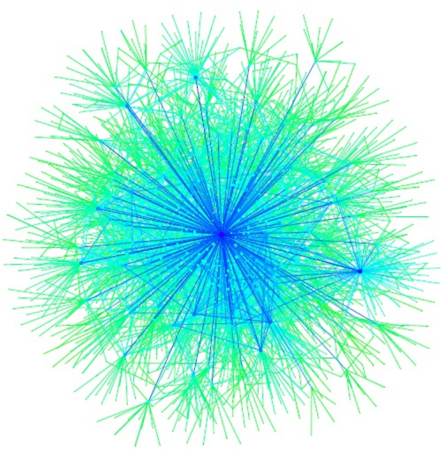
Primeri grafov in omrežij, kjer je stopnja vozlišča porazdeljena po potenčnem zakonu:

- Graf sodelovanja med igralci (hollywood graf),$\gamma \approx 2.3$



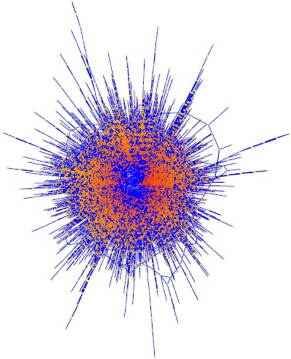
Slika : Graf sodelovanj med igralci z 10000 vozlišči.

- Graf sodelovanj med matematiki: Ima približno 401 000 vozlišč. Ta graf ima $\gamma \approx 2.4$.



Slika : Graf sodelovanj med matematiki.

- Grafi socialnih omrežij kot so Facebook, MySpace, Twitter, Yahoo! Instant messaging, MSN etc., imajo praviloma $2 < \gamma < 3$:



Slika : **Graf Yahoo! Instant messaging grafa.** To je primer grafa socialnega omrežja, ki ima 29000 vozlišč in 39000 povezav.

Razdalja

- Računsko bolj kompleksna mera
- Razdalja med dvema vozliščema grafa:
  $d(u, v) = \min\{P | P \text{je najkrajša pot med u in v}\}$
- Če razdalje zložimo v matriko dobimo matriko velikosti $V \times V$, indeksirano z vozlišči grafa, tj.

$$D = (d(u, v))_{u,v \in V} \quad \text{Matrika razdalje.}$$

- **Povprečna oz. karakteristična razdalja**
  - To je aritmetrična sredina vseh razdalj v grafu:
    $\overline{d} = \frac{1}{|V^2| - |V|} \sum_{i \neq v \in V} d(u, v)$.
  - Če poznamo matriko razdalje D, potem lahko izračunamo povprečno razdaljo v $O(n^2)$ času.

Razdalja (nadaljevanje)

- Ekscentričnost: $\epsilon(u) := \max\{d(u,v) | v \in V\}$
- Radij: $\mathrm{rad}(G) = \min\{\epsilon(u) | u \in V\}$.
- Diameter: $\mathrm{diam}(G) = \max\{d(u,v) | u, v \in V\}$.
- Okolice:
  - h-okolica: $\mathrm{Neigh}_h(v) := \{u \in V | d(u,v) \leq h\}$.
  - Velikost h-okolice označimo z: $N(v,h) = |\mathrm{Neigh}_h(v)|$.
- Hop plot:
  $P(h) = |\{(u,v) \in V^2 | d(u, v \leq h)\}| = \sum_{v \in V} N(v,h)$.
- Povprečno velikost h-okolice:
  $\overline{\mathrm{Neigh}}(h) = \frac{1}{n} \sum_{v \in V} N(v,h) = \frac{P(h)}{n}$.

Algoritmi za iskanje najkrajše poti

- Iskanje najkrajše poti med dvema vozliščema v omrežju s poljubnimi utežmi $\omega : E \to \mathbb{R}$ na povezavah je NP-težak problem. Če se omejimo na omrežja brez negativnih ciklov, lahko problem rešimo v polinomskem času.
- **SSSP problem (angl. Single source shortest path)**
  - Če rešimo SSSP problem, lahko izračunamo matriko razdalje za vsak izvor posebej - za vsako vozlišče.
  - Glede na vrsto utežne funkcije poznamo različne algoritme:

Algoritmi za iskanje najkrajše poti (nadaljevanje)

- Če imamo neuteženo omrežje, potem rešimo SSSP preko iskanja v širino (angl. Breadth-first-search) v O(n+m) času.
- Če imamo omrežje, ki ima samo nenegativne uteži, potem rešimo SSSP preko Dijkstra algoritma v O(m+nlogn) času.
- Če imamo omrežje brez negativnih ciklov pa uporabimo Bellman-Ford algoritem v O(mn) času.

Algoritmi za iskanje najkrajše poti (nadaljevanje)

- **APSP problem (angl. All pairs shortest paths):**
- Preko Floyd-Warshall algoritma, v času $O(n^3.)$
- Preko SSSP algoritmov, za vsako vozlišče posebej:
    - Neutežen graf: $O(nm)$
    - Nenegativne uteži na grafu: $O(nm + n^2 logn)$
    - Graf brez negativnih ciklov: $O(n^2 m)$

O številu najkrajših poti

- Število različnih najkrajših poti med vozliščema u in v
  - $c(u, v) := |\{P; P \text{ je najkrajša pot od } u \text{ do } v\}|$
  - Floyd - Warshall algoritem

Algorithm 1 Štetje različnih najkrajših pot

Require: Z utežmi w utežen graf $G = (V, E)$, kateri je brez ciklov negativne in ničelne dolžine.
Ensure: Razdalja $d(u, v)$ in število različnih najkrajših poti med vsakim parom vozlišč u, v.

for all v ∈ V **do**
  for all $u \in V$ **do**

$$d(u,v) = \begin{cases} 0 & \text{če } u = v \\ w(u,v) & \text{če } uv \in E \\ \infty & \text{sicer} \end{cases}$$

$$c(u,v) = \begin{cases} 1 & \text{če } uv \in E \\ 0 & \text{sicer} \end{cases}$$

  end for
end for
for all v' ∈ V **do**
  for all u ∈ V **do**
    for all v ∈ V **do**
      if d(u,v') + d(v',v) = d(u,v) **then**
        c(u, v) = c(u,v) + c(u,v')c(v',v) (Najdemo pot enake dolžine.)
      else
        if $d(u, v') + d(v', v) < d(u, v)$ **then**
          d(u,v) = d(u,v') + d(v',v)
          c(u,v) = c(u,v')c(v',v) (Najdemo krajšo pot.)
        end if
      end if
    end for
  end for
end for

O številu najkrajših poti

- Število različnih najkrajših poti med vozliščema u in v
  - $c(u, v) := |\{P; P \text{ je najkrajša pot od } u \text{ do } v\}|$
  - Floyd - Warshall algoritem
- Število različnih najkrajših poti s povezavo e
  - $c(e) := |\{P; P \text{ je najkrajša pot, ki vsebuje povezavo } e\}|$
  - e ima vozlišči u, v

$$c(e) = \sum_{d(u',v')=d(u',u)+1+d(v,v')} c(u', u)c(v, v')$$

Koeficient popačenja

- Graf $G(V, E)$, vpeto drevo T
- Popačenje (distortion):

$$D(T) := \frac{1}{|E|} \sum_{(u,v) \in E} d_T(u, v)$$

Koeficient popačenja nam pove, kolikšna je povprečna najkrajša pot med dvema sosednjima vozliščema grafa G v drevesu T.

- Globalno popačenje:

$$D(G) = \min\{D(T); T \text{ je vpeto drevo grafa } G\}$$

Koeficient gručavosti in tranzitivnost

- Cikel dolžine tri $\Delta = (V_\Delta, E_\Delta)$
  - $\lambda(G)$ ... število ciklov dolžine tri v grafu G
  - $\lambda(v) = |\{\Delta; v \in V_\Delta\}|$
  - $\lambda(G) = \frac{1}{3} \sum_{v \in V} \lambda(v)$
- Triada
  - Triada v vozlišču
  - Število triad v vozlišču v: $\tau(v) = \binom{d(v)}{2} = \frac{d(v)^2 - d(v)}{2}$
  - Število triad v grafu G: $\tau(G) = \sum_{v \in V} \tau(v)$

- Koeficient gručavosti
  - Koeficient gručavosti za vozlišče v $(\tau(v) \neq 0)$

$$c(v) = \frac{\lambda(v)}{\tau(v)}$$

  - $V' = \{v \in V; d(v) > 1\}$
  - Koeficient gručavosti za cel graf

$$C(G) = \frac{1}{|V'|} \sum_{v \in V'} c(v)$$

- Tranzitivnost

$$T(G) = \frac{3\lambda(G)}{\tau(G)}$$

$$0 \leq T(G) \leq 1$$

- Odnos med tranzitivnostjo in koeficientom gručavosti

$$T(G) = \frac{\sum_{v \in V'} \tau(v) c(v)}{\sum_{v \in V'} \tau(v)}$$

- Pri tranzitivnosti so vsi cikli dolžine tri medsebojno enakovredni, pri koeficientu gručavosti pa so enakovredna vozlišča.

- Izračun
  - Node-iterator
  - (Hitro) matrično množenje
  - AYZ Algoritem
    - Alon, Yuster, Zwick (1997)
    - Algoritem poteka tako, da najprej loči vozlišča na dva tipa. Taka z nizko stopnjo in taka z višjo stopnjo. Na vozliščih z nižjo stopnjo išče trikotnike z "node-iteratorjem", na podgrafu induciranem z vozlišči višje stopnje pa s hitrim matričnim množenjem.

$$V_{low} = \{v \in V; d(v) \leq \beta\}; \beta = m^{\frac{y-1}{y+1}}$$

$$V_{high} = \frac{V}{V_{low}}$$

Input: Graph G with adjacency array representation and hashed edge set matrix multiplication parameter γ

Output: number of triangles $\lambda(v)$ for each node

1 $\beta \longleftarrow m^{(\gamma-1)/(\gamma+1)}$

2 **for** $v \in V$ **do**

  $\lambda(v) \leftarrow 0$

  if $d(v) \leq \beta$ **then**

    $V_{\text{low}} \leftarrow V_{\text{low}} \cup \{v\}$

  else

    $V_{\text{high}} \leftarrow V_{\text{high}} \cup \{v\}$

3 **for** $v \in V_{low}$ **do**

  for *all pairs of neighbors* $\{u, w\}$ *of* v **do**

4   **if** *edge between* u *and* w *exists* **then**

5   **if** $u, w \in V_{low}$ **then**

        for $z \in \{v, u, w\}$ **do**

          $\lambda(z) \leftarrow \lambda(z) + 1/3$

6   **else if** $u, w \in V_{high}$ **then**

        for $z \in \{v, u, w\}$ **do**

          $\lambda(z) \leftarrow \lambda(z) + 1$

7   **else**

        for $z \in \{v, u, w\}$ **do**

          $\lambda(z) \leftarrow \lambda(z) + 1/2$

8 A $\leftarrow$ adjacency matrix of node induced subgraph of V_{high}

9 M $\leftarrow A^3$

10 **for** $v \in V_{high}$ **do**

11   $\lambda(v) \leftarrow \lambda(v) + M(i, i)/2$ where i is index of v

- LEMA: Algoritem AYZ za vsako vozlišče v v grafu G izračuna število ciklov dolžine tri $\lambda(v)$ in ga lahko implementiramo tako, da je njegova časovna zahtevnost reda $O(m^{\frac{2\gamma}{\gamma+1}})$ oz. $O(m^{1.41})$.

Aproksimacija $c(u, v)$

- Slučajno vzorčenje
  - X_i ... omejena slučajna spremenljivka, $0 \leq X_i \leq M$
  - k ... število vzorcev
- Hoeffdingova neenakost

$$\mathbb{P}\left(\left|\frac{1}{k}\sum_{i=1}^{k} X_i - \mathbb{E}\left(\frac{1}{k}\sum_{i=1}^{k} X_i\right)\right| \geq \epsilon\right) \leq 2e^{\frac{-2k\epsilon^2}{M^2}}$$

- Hoeffdingova meja
  - Fiksirajmo ϵ in p. $\Rightarrow$ Dobimo k.
  - Ocenjujemo $c(v)$ za vsako vozlišče v $\Rightarrow$ k je število sosednjih parov, za katere nas zanima, ali so povezani ali ne.
  - Ocenjujemo $c(G)$

    $\Rightarrow$ k je število vozlišč, ki jih vključimo v izračun (moramo poznati $c(v)$ oz. ga oceniti).

    $\Rightarrow$ k je število vozlišč, ki jih vključimo v izračun (moramo poznati $c(v)$ oz. ga oceniti)
  - Podobno za tranzitivnost.

Algorithm 7.1: *c*-APPROXIMATION BY NODE SAMPLING

Input: graph $G = (V, E)$ with $\forall v \in V$: $(v) \geq 2$;
number of samples k;
Output: approximation of $c(G)$;
Data: real r;
$r \leftarrow 0$;
for $(1, \ldots, k)$ **do**
  $v \leftarrow$ uniformRandomNode of all nodes V;
  $r \leftarrow r + \varrho(v)$;
return $\text{apx}(c(G)) = r/k$;

- LEMA: Za vsak konstanten ϵ in konstantno pravilnost p obstaja algoritem, ki oceni koeficient gručavosti $c(v)$ za vsako vozlišče in koeficient tranzitivnosti $T(G)$ za celoten graf v času $O(n)$. Koeficient gručavosti $C(G)$ lahko ocenimo v $O(1)$.

Mrežni motivi

- Kakšne podgrafe lahko najdemo v omrežju?
- Ali se kakšni podgrafi pojavljajo pogosteje, kot bi bilo to statistično pričakovati?
  - Štetje "enakih" podgrafov v grafu
  - Testiranje statistične značilnosti

- Štetje podgrafov:
  - Iskanje podgrafa s k vozlišči
  - Iskanje ciklov dolžine tri (AYZ algoritem)
  - Iskanje podgrafov s tremi vozlišči (modificiran AYZ)
  - Iskanje K_4 podgrafov (Kloks, Kratsch, Müller)
  - Iskanje podgrafov z največ štirimi vozlišči (na neusmerjenih grafih)

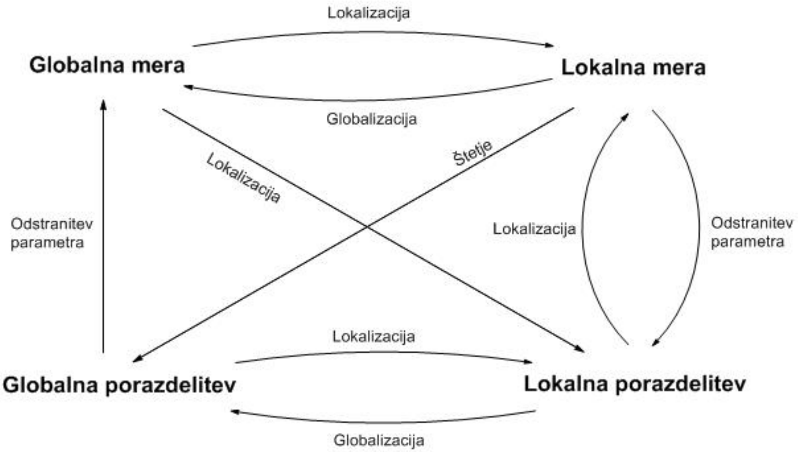
Vrste omrežnih mer

- **Globalna mera** γ označuje (eno samo) vrednost $\gamma_G \in Y$ za vsak graf $G \in \mathcal{G}$.
  - število vozlišč/povezav, diameter, koeficient gručavosti
- **Globalna porazdelitev** Γ označuje preslikavo $\Gamma_G : P \to Y$ za vsak graf $G \in \mathcal{G}$.
  - če $P = \mathbb{N}$, dobimo zaporedje $(\Gamma_G(0), \Gamma_G(1), \ldots)$
  - porazdelitve vhodnih/izhodnih stopenj, hop plot

- **Lokalna mera** λ_G označuje (eno samo) vrednost $\lambda_G(x) \in Y$ za vsak graf $G \in \mathcal{G}$, kjer je x določen element grafa G. Bolj natančno, gre za preslikavo $\lambda_G : X_G \to Y$.
  - vhodna in izhodna stopnja vozlišča, utež/kapaciteta povezave, razdalja

- **Lokalna porazdelitev** Λ označuje preslikavo $\Lambda_G : X_G \times P \to Y$ za vsak graf $G \in \mathcal{G}$.
  - velikost okolice, eksentričnost, diameter

Transformacije omrežnih mer



Slika: Transformacije vrst omrežnih mer

- **Globalizacija**: odstranitev odvisnosti lokalne mere ali porazdelitve od elementov grafa (računanje, izbiranje, konstrukcija)
  - maksimum $\gamma_G := \max\{\gamma_G(x) \mid x \in X_G\}$
  - povprečje $\gamma_G := \frac{1}{|X_G|} \sum_{x \in X_G} \gamma_G(x)$
  - diameter in radij, globalno popačenje

- **Štetje** (lokalna mera $\rightarrow$ globalna porazdelitev): preštejemo število elementov grafa tako, da $\lambda_G(x)$ leži v določeni množici vrednosti
    - $\Gamma_G(t) := |\{x \in X_G \mid \lambda_G(x) = t\}|$
    - porazdelitve stopnje vozlišča

- **Odstranitev parametra** (porazdelitev $\rightarrow$ mera): izračunamo vrednost λ_G iz zaporedja vrednosti, danega s porazdelitvijo Λ_G (spremenljivka je parameter)
  - maksimum $\lambda_G(x) := \max\{\Lambda_G(x,t) \mid t \in P\}$
  - povprečje $\lambda_G := \frac{1}{|P|} \sum_{t \in P} \Lambda_G(x,t)$
  - ekscentričnost

- **Lokalizacija**
  - Globalna mera $\gamma \to$ lokalna porazdelitev Λ:

    izberemo podgraf $H(x,t) \subseteq G$ za vsak $x \in X_G$, $t \in P$ in nastavimo $\Lambda_G(x,t) := \gamma_{H(x,t)}$
  - Globalna mera $\gamma \to$ lokalna mera λ:

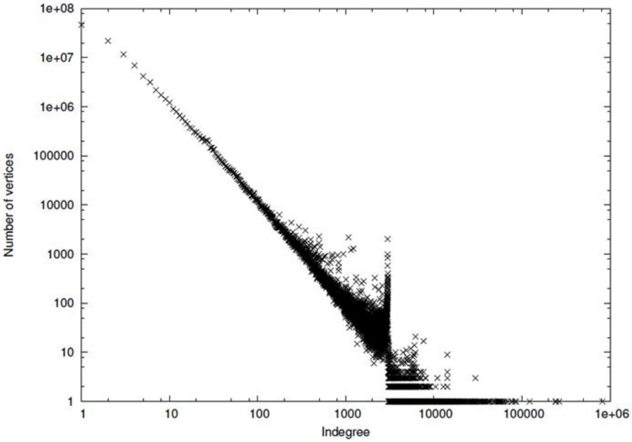
    izberemo podgraf $H(x) \subseteq G$ za vsak $x \in X_G$ in nastavimo $\lambda_G(x) := \gamma_{H(x)}$

- **Lokalizacija**
  - Globalna porazdelitev $\Gamma \rightarrow$ lokalna porazdelitev Λ:

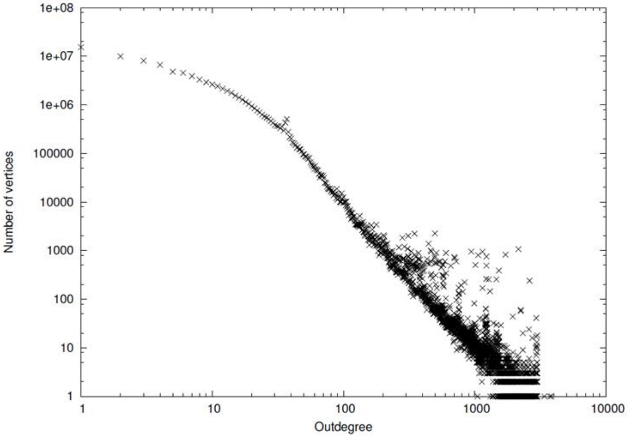
    izberemo podgraf $H(x) \subseteq G$ za vsak $x \in X_G$ in nastavimo $\Lambda_G(x, t) := \Gamma_{H(x)}(t)$
  - Lokalna mera $\lambda \rightarrow$ lokalna porazdelitev Λ:

    izberemo podgraf $H(x) \subseteq G$ za vsak $x \in X_G$ in nastavimo $\Lambda_G(x, t) := \lambda_{H(x,t)}(x)$

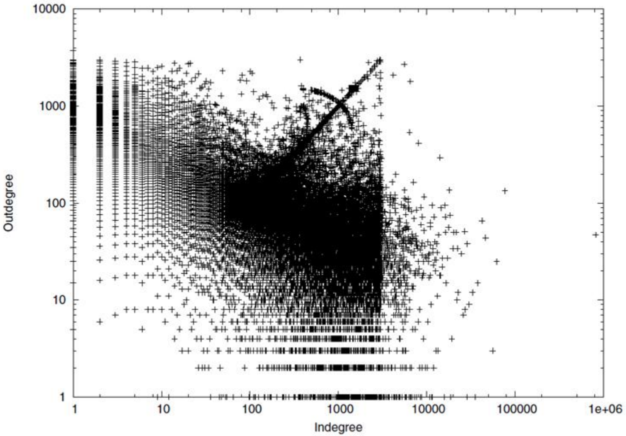
Vizualizacija



Slika: Absolutna porazdelitev vhodne stopnje z logaritemsko skalo (omrežje Crawl of Web Base, 2001)



Slika: Absolutna porazdelitev izhodne stopnje z logaritemsko skalo (omrežje Crawl of Web Base, 2001)



Slika: Razsevni diagram vhodne in izhodne stopnje z logaritemsko skalo (omrežje Crawl of Web Base, 2001)