Analiza omrežij

5. Zgradba omrežij povezanosti

Vladimir Batagelj

Magistrski program Uporabna statistika
Ljubljana, april 2024

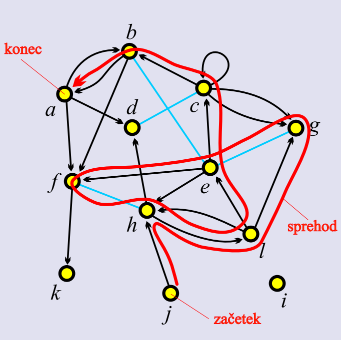
Kazalo

London Tube

prof. Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si
prosojnice (PDF)

30. april 2024 ob 03 : 03/ april 2013

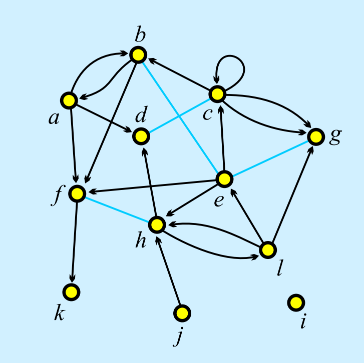
Sprehodi



dolžina $|s|$ sprehoda s je število povezav, ki ga sestavljajo.
$s = (j, h, l, g, e, f, h, l, e, c, b, a)$
$|s| = 11$
Sprehod je *sklenjen* ali *obhod* ntk. njegov začetek in konec sovpadata.
Če ne upoštevamo smeri povezav v 'sprehodu', dobimo *polsprehod* ali *verigo*.
sled – sprehod z različnimi povezavami
pot – sprehod z različnimi vozlišči
cikel – sklenjen sprehod z različnimi notranjimi vozlišči.

Graf je *acikličen*, ntk. ne vsebuje nobenega cikla.

Najkrajše poti



Dolžino najkrajše poti iz u v v označimo z $d(u,v)$.
Če ne obstaja sprehod iz u v v, postavimo $d(u,v) = \infty$.

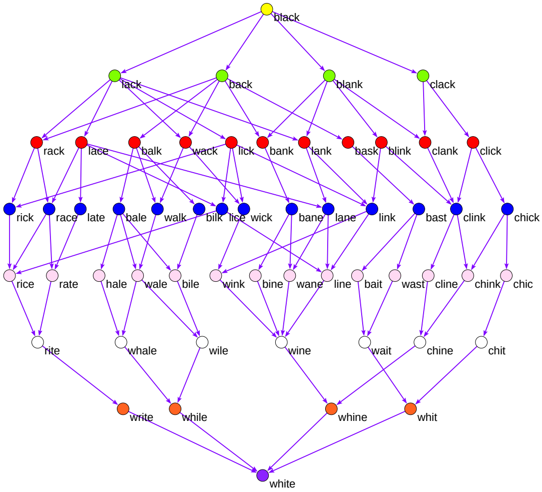
$d(j,a) = |(j,h,d,c,b,a)| = 5$
$d(a,j) = \infty$

$\hat{d}(u,v) = \max(d(u,v), d(v,u))$ je *razdalja*:
$\hat{d}(v,v) = 0$, $\hat{d}(u,v) = \hat{d}(v,u)$,
$\hat{d}(u,v) \leq \hat{d}(u,t) + \hat{d}(t,v)$.

Premer grafa je enak razdalji med, glede na $d(u,v)$, najoddaljenejšima vozliščema: $D = \max_{u,v \in \mathcal{V}} d(u,v)$.

`Network/Create New Network/Subnetwork with Paths/`

Najkrajše poti



DICT28.

Enakovrednosti in razbitja

Relacija R na $\mathcal{V}$ je *enakovrednost* (ekvivalenčna) ntk. je *refleksivna* $\forall v \in \mathcal{V} : vRv$, *simetrična* $\forall u, v \in \mathcal{V} : (uRv \Rightarrow vRu)$ in *tranzitivna* $\forall u, v, z \in \mathcal{V} : uRz \wedge zRv \Rightarrow uRv$.

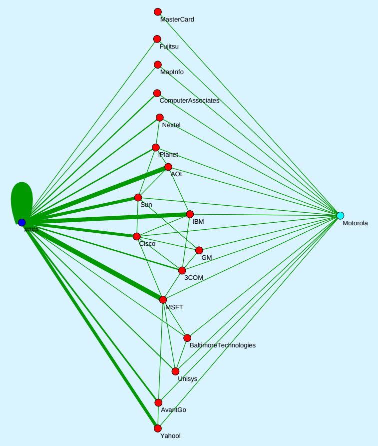
Vsaka enakovrednost R določa neko razbitje v *razrede* $[v] = \{u : vRu\}$.

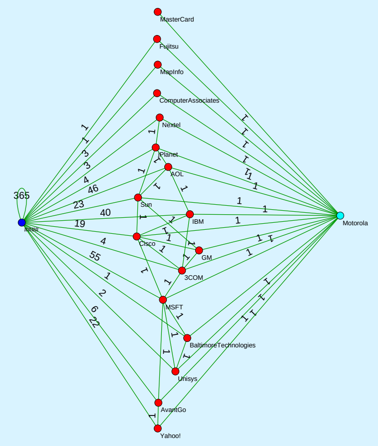
Vsako razbitje $\mathbf{C}$ določa neko enakovrednost $uRv \Leftrightarrow \exists C \in \mathbf{C} : u \in C \wedge v \in C$.

k-sosedi vozlišča v je množica vozlišč, ki so za k oddaljena od v, $N^k(v) = \{u \in v : d(v, u) = k\}$.
Množica vseh množic k-sosedov, $k = 0, 1, \ldots$ vozlišča v je razbitje množice $\mathcal{V}$.
k-soseščina vozlišča v, $N^{(k)}(v) = \{u \in v : d(v, u) \leq k\}$.

`Network/Create Partition/k-Neighbors/`

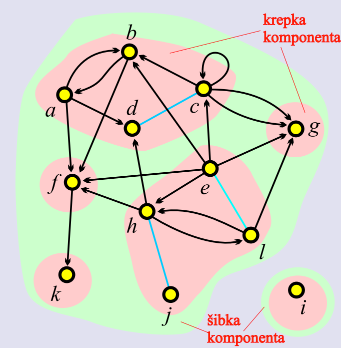
Soseščina Motorole





Debelina povezav je koren iz vrednosti.

Povezanosti



Vozlišče u je *dosegljivo* iz vozlišča v ntk. obstaja sprehod z začetkom v in koncem u.

Vozlišče v je *šibko povezano* z vozliščem u ntk. obstaja veriga s krajiščema v in u.

Vozlišče v je *krepko povezano* z vozliščem u ntk. sta vzajemno dosegljivi.

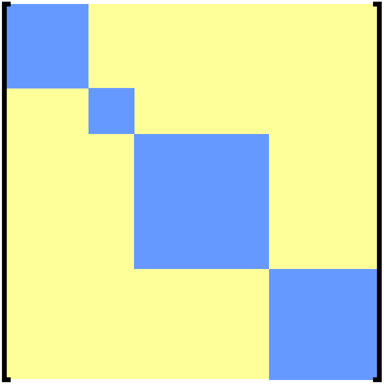
Šibka in krepka povezanost sta enakovrednosti.

graphCon.net

Razredi porajajo šibke/krepke *komponente* ali *kose* grafa.

`Network/Create Partition/Components/`

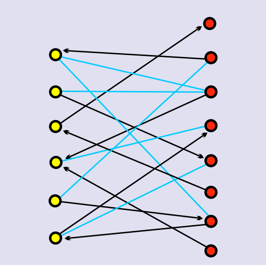
Šibke komponente

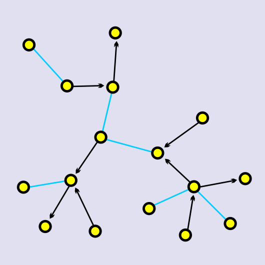


Če preuredimo vozlišča omrežja, tako da vozlišča iz iste skupine šibkega razbitja postavimo skupaj, dobimo matrični prikaz sestavljen iz diagonalnih blokov – šibkih komponent.

Za večino problemov velja, da jih lahko ločeno rešimo za vsako šibko komponento posebej in nato dobljene rešitve združimo v rešitev za celotno omrežje.

Posebni grafi – dvodelni, drevo

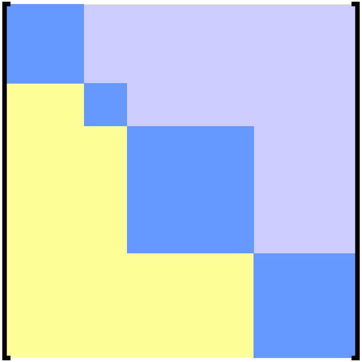


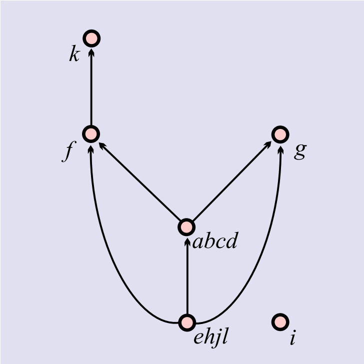


Graf $G = (\mathcal{V}, \mathcal{L})$ je *dvodelen* ntk. lahko množico vozlišč $\mathcal{V}$ razbijemo na podmnožici V_1 in V_2, tako da ima vsaka povezava iz $\mathcal{L}$ eno krajišče v $\mathcal{V}_1$ drugo pa v $\mathcal{V}_2$.

Šibko povezan graf G je *drevo* ntk. ne vsebuje (zank in) polciklov dolžine vsaj 3.

Krepka skrčitev grafa (kondenzacija)

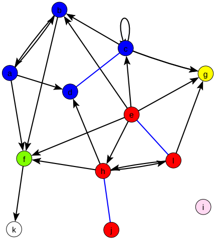


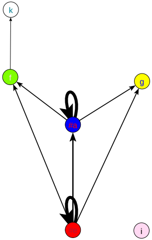


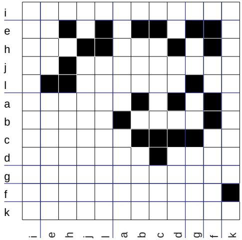
Če v danem grafu skrčimo vsako krepko komponento v ustrezno vozlišče, odstranimo zanke in združimo vzporedne povezave, je tako dobljeni *skrčeni* graf acikličen. Za vsak aciklični graf obstaja *urejenost* / *oštevilčenje* $i : \mathcal{V} \to \mathbb{N}$, tako da velja $(u, v) \in \mathcal{A} \Rightarrow i(u) < i(v)$.

Skrčitev grafa – primer

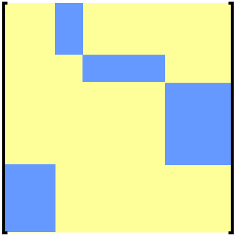
```
Network/Create Partition/Components/Strong [1]
Operations/Network+Partition/Shrink Network  [1][0]
Network/Create New Network/Transform/Remove/Loops [yes]
Network/Acyclic Network/Depth Partition/Acyclic
Partition/Make Permutation
Permutation/Inverse Permutation
select partition [Strong Components]
Operations/Partition+Permutation/Functional Composition/Partition*Permutation
Partition/Make Permutation
select [original network]
File/Network/Export Matrix to EPS/Using Permutation
```

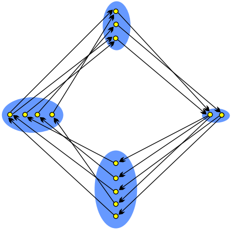






Zgradba krepkih komponent





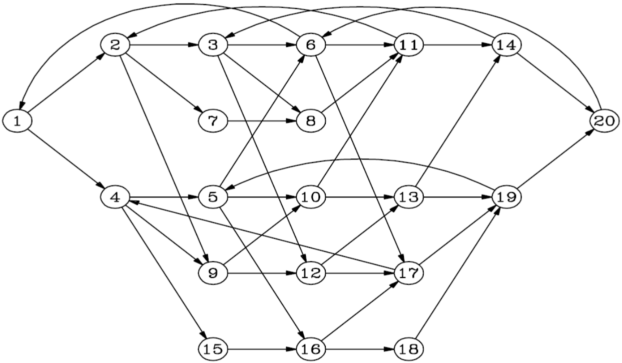
Notranja zgradba krepke komponente – naj bo d največji skupni delitelj dolžin obhodov v krepki komponenti.

Komponenta je *enostavna*, če je $d = 1$; sicer je *periodična* s periodo d.

Množico vozlišč $\mathcal{V}$ krepko povezanega usmerjenega grafa $\mathcal{G} = (\mathcal{V}, R)$ je mogoče razbiti na d skupin $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_d$, tako da za vsako povezavo $(u, v) \in R$ velja $u \in \mathcal{V}_i \Rightarrow v \in \mathcal{V}_{(i \bmod d)+1}$.

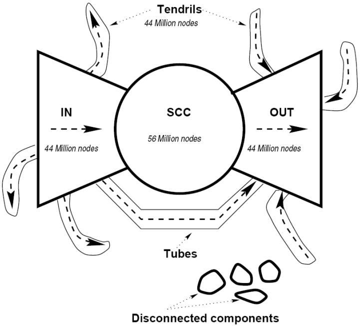
`Network/Create Partition/`
`Components/Strong-Periodic`

. . . Zgradba krepkih komponent



Bonhourejev periodični graf. Pajek – matrike

Spletni metuljček (Bow-tie)



Kumar &: The Web as a graph

Naj bo $\mathcal{S}$ *največja krepka komponenta* v omrežju $\mathcal{N}$; $\mathcal{W}$ šibka komponenta, ki vsebuje $\mathcal{S}$; $\mathcal{I}$ množica vozlišč, iz katerih je $\mathcal{S}$ dosegljiva; $\mathcal{O}$ množica vozlišč dosegljivih iz $\mathcal{S}$; $\mathcal{T}$ (cevi) vozlišča (niso v $\mathcal{S}$) na poteh iz $\mathcal{I}$ v $\mathcal{O}$; $\mathcal{R} = \mathcal{W} \setminus (\mathcal{I} \cup \mathcal{S} \cup \mathcal{O} \cup \mathcal{T})$ (lovke); in $\mathcal{D} = \mathcal{V} \setminus \mathcal{W}$. Razbitje

$$\{\mathcal{I}, \mathcal{S}, \mathcal{O}, \mathcal{T}, \mathcal{R}, \mathcal{D}\}$$

imenujemo *metuljčno* razbitje $\mathcal{V}$.

Opomba: za splošna usmerjena omrežja slika ne prikazuje vseh možnosti – v množici $\mathcal{R}$ so lahko tudi verige.

```
Network/Create Partition/Bow-Tie
```

Dvopovezanost

Vozlišči u in v sta *dvopovezani* ntk. sta povezani (v obe smeri) s po dvema neodvisnima (brez skupnih notranjih vozlišč) potema. Dvopovezanost določa razbitje množice **povezav**.

Vozlišče je *stično* vozlišče ali *stičišče* ntk. njegova odstranitev poveča število šibkih komponent grafa.

Povezava je *most* ntk. njena odstranitev poveča število šibkih komponent grafa.

`Network/Create New Network/with Bi-Connected Components`

k-povezanost

Vozliščna povezanost κ grafa G je enaka najmanjšemu številu vozlišč, ki jih je potrebno odvzeti iz grafa, tako da je graf porojen s prestalimi vozlišči nepovezan ali trivialen (enak K_1).

Povezavna povezanost λ grafa G je enaka najmanjšemu številu povezav, ki jih je potrebno odvzeti iz grafa, tako da je graf porojen s prestalimi povezavami nepovezan ali trivialen.

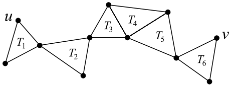
Velja Whitneyeva neenakost: $\kappa(G) \leq \lambda(G) \leq \delta(G)$.

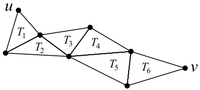
Graf G je *(po vozliščih) $k-$povezan*, če je $\kappa(G) \geq k$ in je *po povezavah $k-$povezan*, če je $\lambda(G) \geq k$.

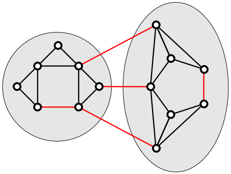
Velja Whitneyeva različica Mengerjevega izreka: Graf G je po vozliščih/povezavah $k-$povezan ntk. vsak par vozlišč povezuje vsaj k po vozliščih/povezavah ločenih sprehodov.

Trikotniška povezanost – neusmerjeni grafi

V neusmerjenem grafu imenujemo *trikotnik* podgraf izomorfen K_3. Zaporedje trikotnikov $(T_1, T_2, \ldots, T_s)$ grafa G *(vozliščno) trikotniško povezuje* vozlišči $u, v \in \mathcal{V}$ ntk. $u \in T_1$ in $v \in T_s$ ali $u \in T_s$ in $v \in T_1$ ter $\mathcal{V}(T_{i-1}) \cap \mathcal{V}(T_i) \neq \emptyset$, $i = 2, \ldots s$; in *povezavno trikotniško povezuje* vozlišči $u, v \in \mathcal{V}$ ntk zadošča še strožji različici zadnjega pogoja $\mathcal{E}(T_{i-1}) \cap \mathcal{E}(T_i) \neq \emptyset$, $i = 2, \ldots s$.

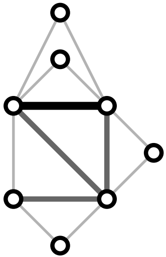






Vozliščna trikotniška povezanost je enakovrednost na vozliščih; povezavna pa na povezavah. Članek.

Trikotniško omrežje

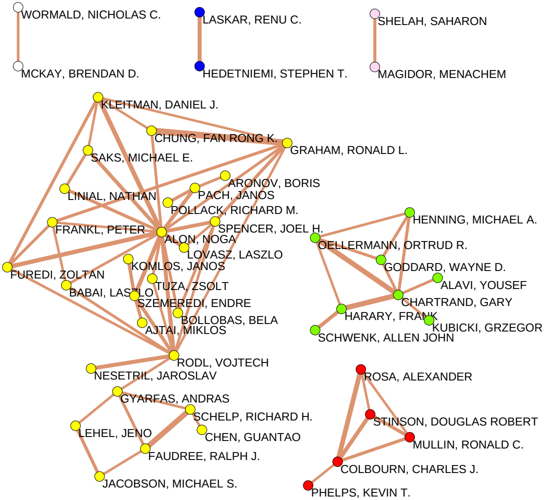


Naj bo $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ enostaven neusmerjen graf. Prirejeno *trikotniško omrežje* $\mathcal{N}_T(\mathcal{G}) = (\mathcal{V}, \mathcal{E}_T, w)$ določeno z $\mathcal{G}$ je podgraf $\mathcal{G}_T = (\mathcal{V}, \mathcal{E}_T)$ grafa $\mathcal{G}$, kjer je E_T množica tistih povezav iz $\mathcal{E}$, ki leže na vsaj enem trikotniku. Utež $w(e)$ povezave $e \in \mathcal{E}_T$ je enaka številu različnih trikotnikov, ki jim povezava e pripada.

Trikotniška omrežja omogočajo učinkovito razkrivanje gostih delov omrežja. Če povezava e pripada *k-kliki* (podgrafu izomorfnemu K_k) v $\mathcal{G}$, je $w(e) \geq k - 2$.

`Network/Create New Network/with Ring Counts/3-Rings/Undirected`

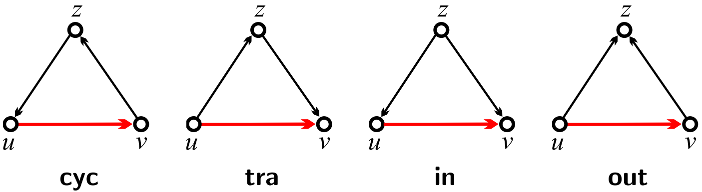
Povezavni izrez na ravni 16 trikotniškega omrežja Erdős-ovega grafa sodelovanj



brez Erdősa,
$n = 6926$,
$m = 11343$

Trikotniška povezanost – usmerjeni grafi

Če je graf $\mathcal{G}$ mešan, zamenjamo neusmerjene povezave s pari nasprotno usmerjenih. Naj bo $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ enostaven usmerjen graf brez zank. Za izbrano usmerjeno povezavo $(u, v) \in \mathcal{A}$ obstajajo štiri vrste *usmerjenih trikotnikov*: **cyc**lic, **tra**nsitive, **in**put in **out**put.



. . . Trikotniška povezanost – usmerjeni grafi

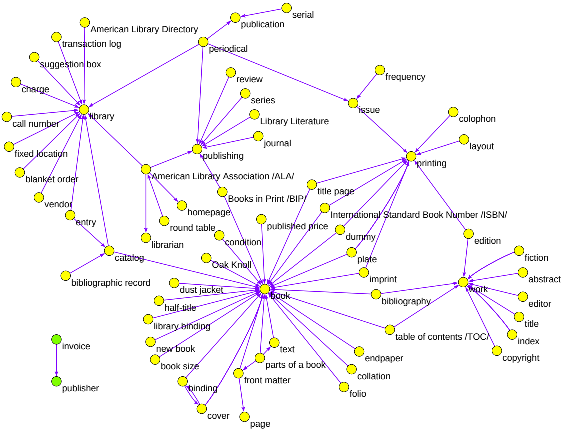
Če pozabimo na vlogo izbrane povezave, imamo le dve vrsti trikotnikov, ki jim povezava lahko pripada: ciklične (**cyc**) in tranzitivne (**tra**, **in**, **out**). V programu `Pajek` ukaz

`Network/Create New Network/with Ring Counts/3-Rings/Directed`

omogoča določiti ustrezna omrežja ($\mathcal{N}_{cyc}$ – ciklične uteži, $\mathcal{N}_{tra}$ – tranzitivnostne uteži, $\mathcal{N}_{sc}$ – tranzitivne bližnjice).

Pojem trikotniške povezanosti lahko posplošimo na *povezanost s kratkimi (pol)cikli – obroči* in ustrezna omrežja; lahko pa tudi na povezanost z majhnimi podgrafi (npr. "klikami", glejte Palla).

Povezavni izrez na ravni 11 tranzitivnega trikotniškega omrežja slovarja ODLIS



Pomembna vozlišča v omrežju

Za določitev pomembnih/zanimivih elementov (vozlišč/povezav) v omrežju ponavadi poskusimo izraziti naša pričakovanja z mero (indeks, utež), ki zadošča zahtevi

> *večja je za dani element vrednost mere, pomembnejši/ zanimivejši je ta element*

V analizi omrežij raziskovalci prepogosto nekritično kar uporabijo mere iz literature/programov.

Za formalni pristop glej Roberts.

Pomembna vozlišča v omrežju

Pri izgradnji *mer pomembnosti* moramo najprej upoštevati ali je omrežje usmerjeno ali neusmerjeno. Meram pomembnosti na neusmerjenih omrežjih pravimo mere *središčnosti*; na usmerjenih omrežjih pa mere *veljave*. Slednje se naprej delijo na mere *ugleda* ali *podpore* (upoštevamo vstopajoče povezave) in mere *vpliva* (upoštavamo izstopajoče povezave).

Če zamenjamo dano usmerjeno omrežje z njemu nasprotnim (obrnemo smeri povezav) preidejo mere ugleda v mere vpliva, in obratno.

Dejanski pomen mere pomembnosti je odvisen od relacije (omrežja). Tako npr. je 'najuglednejša' oseba glede na relacijo '__ ne mara sodelovati z __' dejansko najmanj priljubljena oseba.

Odstranitev pomembnega vozlišča iz omrežja povzroči občutno spremembo v zgradbi/delovanju omrežja.

Normalizacija

Naj bo $p : \mathcal{V} \to \mathbb{R}$ neka mera pomembnosti vozlišč omrežja $\mathcal{N} = (\mathcal{V}, \mathcal{L})$. Če želimo vrednosti mere p primerjati med različnimi omrežji, moramo poskrbeti za primerljivost. Pogosto jo poskušamo zagotoviti tako, da mero p *normaliziramo*.

Naj bo $\mathcal{N} \in \mathbf{N}(\mathcal{V})$, kjer je $\mathbf{N}(\mathcal{V})$ izbrana množica omrežij nad isto množico $\mathcal{V}$,

$$p_{max} = \max_{\mathcal{N} \in \mathbf{N}(\mathcal{V})} \max_{v \in \mathcal{V}} p_{\mathcal{N}}(v) \qquad \text{in} \qquad p_{min} = \min_{\mathcal{N} \in \mathbf{N}(\mathcal{V})} \min_{v \in \mathcal{V}} p_{\mathcal{N}}(v)$$

Tedaj je normalizirana mera enaka

$$p'(v) = \frac{p(v) - p_{min}}{p_{max} - p_{min}} \in [0, 1]$$

Stopnje

Najpreprostejšo mero pomembnosti predstavljajo stopnje vozlišč. Ker sta za enostavna omrežja $\deg_{min} = 0$ in $\deg_{max} = n - 1$, je ustrezna normalizirana mera

središčnost $$\deg'(v) = \frac{\deg(v)}{n-1}$$

in podobno

ugled $$\text{indeg}'(v) = \frac{\text{indeg}(v)}{n}$$

vpliv $$\text{outdeg}'(v) = \frac{\text{outdeg}(v)}{n}$$

Namesto stopenj glede na osnovno omrežje lahko vzamemo tudi stopnje glede na relacijo dosegljivosti (tranzitivna ovojnica).

```
Network/Create Partition/Degree
Network/Create Vector/Centrality/Degree
Network/Create Vector/Centrality/Proximity Prestige
```

Dostopnost

Če upoštevamo razdalje $d(u,v)$ med vozlišči v omrežju $\mathcal{N} = (\mathcal{V}, \mathcal{L})$ lahko vpeljemo

polmer $r(v) = \max_{u \in \mathcal{V}} d(v,u)$
Količino $D = \max_{u,v \in \mathcal{V}} d(v,u)$ imenujemo *premer* omrežja.
skupna dostopnost $S(v) = \sum_{u \in \mathcal{V}} d(v,u)$

Za usmerjeno omrežje sta vpeljani meri meri vpliva. Meri ugleda dobimo, če v obrazcih $d(u,v)$ zamenjamo z $d(v,u)$.

Če omrežje ni krepko povezano, sta r_{max} in S_{max} enaki ∞. Sabidussi (1966) je zato kot mero dostopnosti vpeljal $1/S(v)$ oziroma v normalizirani obliki

dostopnost $$cl(v) = \frac{n-1}{\sum_{u \in \mathcal{V}} d(v,u)}$$

`Network/Create Vector/Centrality/Closeness`
`Network/Create New Network/Subnetwork with Paths/Info on Diameter`

Vmesnost

Pomembna so tudi vozlišča, ki lahko nadzirajo pretok podatkov po omrežju. Če privzamemo, da so za prenos pomembne le najkrajše poti, dobimo kot mero *vmesnosti* (Anthonisse 1971, Freeman 1977)

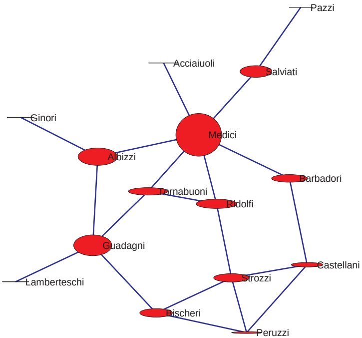
$$b(v) = \frac{1}{(n-1)(n-2)} \sum_{\substack{u,t \in \mathcal{V}: g_{u,t} > 0 \\ u \neq v, t \neq v, u \neq t}} \frac{g_{u,t}(v)}{g_{u,t}}$$

kjer je $g_{u,t}$ število najkrajših poti iz u v t; in $g_{u,t}(v)$ število takih med njimi, ki gredo skozi vozlišče v.

Hiter postopek za izračun vmesnosti je razvil Brandes.

`Network/Create Vector/Centrality/Betweenness`

Padgett-ove floretinske rodbine



		close	between
1.	Acciaiuoli	0.368421	0.000000
2.	Albizzi	0.482759	0.212454
3.	Barbadori	0.437500	0.093407
4.	Bischeri	0.400000	0.104396
5.	Castellani	0.388889	0.054945
6.	Ginori	0.333333	0.000000
7.	Guadagni	0.466667	0.254579
8.	Lamberteschi	0.325581	0.000000
9.	Medici	0.560000	0.521978
10.	Pazzi	0.285714	0.000000
11.	Peruzzi	0.368421	0.021978
12.	Ridolfi	0.500000	0.113553
13.	Salviati	0.388889	0.142857
14.	Strozzi	0.437500	0.102564
15.	Tornabuoni	0.482759	0.091575

Kazala in viri

Vozliščem povezanega usmerjenega omrežja $\mathcal{N} = (\mathcal{V}, \mathcal{L})$ priredimo dve vrednosti: kakovost vira (vsebine) x_v in kakovost kazala y_v (Kleinberg, 1998). Na dober vir kažejo dobra kazala in dobro kazalo kaže na dobre vire

$$x_v = \sum_{u:(u,v)\in\mathcal{L}} y_u \quad \text{in} \quad y_v = \sum_{u:(v,u)\in\mathcal{L}} x_u$$

Naj bo $\mathbf{W}$ matrika omrežja $\mathcal{N}$ in $\mathbf{x}$ ter $\mathbf{y}$ vektorja obeh lastnosti. Tedaj lahko zvezi zapišemo $\mathbf{x} = \mathbf{W}^T\mathbf{y}$ oziroma $\mathbf{y} = \mathbf{W}\mathbf{x}$.

Začnimo z $\mathbf{y} = [1, 1, \ldots, 1]$ in nato zaporedoma izračunamo po obeh zvezah nove približke za $\mathbf{x}$ in $\mathbf{y}$. Oba vektorja po vsakem koraku normaliziramo. To ponavljamo dokler se vektorja ne ustalita. Pokazati je mogoče, da opisani postopek konvergira. Limitni vektor $\mathbf{x}^*$ je glavni lastni vektor matrike $\mathbf{W}^T\mathbf{W}$; $\mathbf{y}^*$ pa matrike $\mathbf{W}\mathbf{W}^T$.

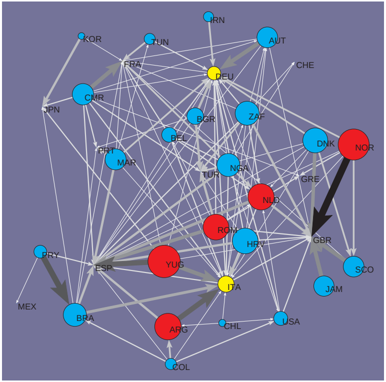
...Kazala in viri

Podobni postopki se uporabljajo v spletnih iskalnikih za ocenjevanje pomembnosti posameznih strani.
PageRank, PageRank / Google, HITS / AltaVista, SALSA, teorija.

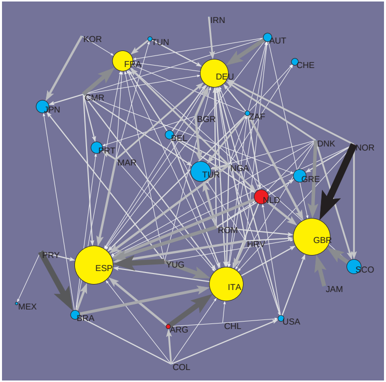
`Network/Create Vector/Centrality/Hubs-Authorities`

Na naslednji prosojnici: Na svetovnem nogometnem prvenstvu v Parizu leta 1998 je sodelovalo 22 nogometnih reprezentanc. V omrežju so vse države, iz katerih so nogometaši igrali v ligah teh 22 držav, in vse države, v katerih ligah so igrali nogometaši iz teh 22 držav. Relacija je *igralec iz države x igra v državi y*; utež je število takih igralcev. Podatke je zbral Lothar Krempel. `football.net`

... Kazala in viri: nogometaši



Izvozniki (kazala/hubs)



Uvozniki (viri/authorities)

Nakopičenost

Nakopičenost v vozlišču v je določena kot razmerje med številom vseh povezav v podgrafu $G^1(v)$ porojenim s soseščino danega vozlišča in številom povezav v polnem grafu na teh vozliščih

$$C(v) = \frac{2|\mathcal{L}(G^1(v))|}{\deg(v)(\deg(v) - 1)}$$

za $\deg(v) > 1$; in $C(v) = 0$ sicer.

Vpliv velikosti soseščine lahko zagotovimo z naslednjim popravkom

$$C_1(v) = \frac{\deg(v)}{\Delta} C(v)$$

kjer je Δ največja stopnja v grafu G. Ta doseže največjo možno vrednost le na vozliščih, ki pripadajo osamljeni kliki reda Δ.

`Network/Create Vector/Clustering Coefficients/CC2`

članek

Izračun indeksov in uteži v Pajek-u

Xingqin Qi et al. so v članku Terrorist Networks, Network Energy and Node Removal vpeljali mero *Laplaceova središčnost*

$$L(v) = \deg(v)(\deg(v) + 1) + 2 \sum_{u \in N(v)} \deg(u)$$

```
select the network
Network/Create Vector/Centrality/Degree/All
Operations/Network+Vector/Neighbours/Sum/All [False]
Vector/Transform/Multiply by [2]
select the degree vector as First
select the degree vector as Second
Vectors/Multiply (First*Second)
Vectors/Add (First+Second)
select the 2*sum on neighbors as Second
Vectors/Add (First+Second)
dispose auxiliary vectors
File/Vector/Change Label [Laplace All centrality]
```

Za določitev pomembnih podomrežij v cestnem omrežju lahko uporabimo utež $w(u, v) = \deg(u) \deg(v)$.

```
Network/Create vector/Centrality/Degree/All
Operations/Network+Vector/Transform/Vector(s)->Line Values/Multiply
```

Usredinjenost omrežja

Mero pomembnosti $p : \mathcal{V} \to \mathbb{R}$ lahko povzamemo na celotnem omrežju kot njegovo usredinjenost $C(p)$:

$$p^* = \max_{v \in \mathcal{V}} p(v)$$

$$D(p) = \sum_{v \in \mathcal{V}} (p^* - p(v))$$

$$D^* = \max_{\mathcal{N} \in \mathbf{N}(\mathcal{V})} D(p_\mathcal{N})$$

Tedaj je *usredinjenost* glede na p

$$C(p) = \frac{D(p)}{D^*}$$

Za večino mer je najbolj usredinjena zvezda S_n in najmanj polni graf K_n.