Introduction to Network Analysis

Structure of networks: connectivity

Vladimir Batagelj

IMFM Ljubljana, IAM UP Koper and NRU HSE Moscow

Master's programme

Applied Statistics with Social Network Analysis

International Laboratory for Applied Network Research

NRU HSE, Moscow 2018

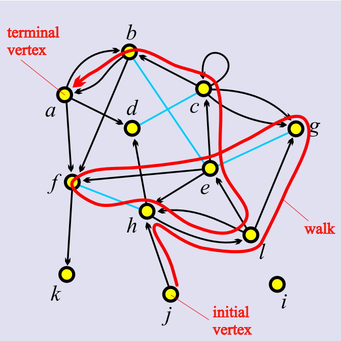
Outline

1. Connectivity
2. Condensation
3. Bow-tie
4. Other connectivities
5. Important nodes
6. Closeness
7. Betweeness
8. Hubs and authorities
9. Clustering

Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si

Current version of slides (December 12, 2018 at 00 : 42): slides PDF

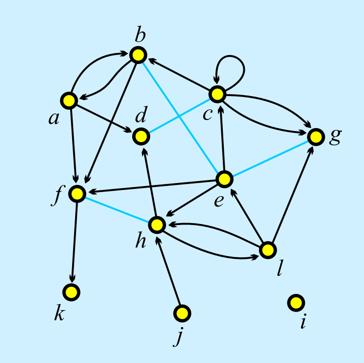
Walks



length $|s|$ of the walk s is the number of links it contains.
$s = (j, h, l, g, e, f, h, l, e, c, b, a)$
$|s| = 11$
A walk is *closed* iff its initial and terminal node coincide.
If we don't consider the direction of the links in the walk we get a *semiwalk* or *chain*.
trail – walk with all links different
path – walk with all nodes different
cycle – closed walk with all internal nodes different

A graph is *acyclic* if it doesn't contain any cycle.

Shortest paths



A shortest path from u to v is also called a *geodesic* from u to v. Its length is denoted by $d(u, v)$.

If there is no walk from u to v then $d(u, v) = \infty$.

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

$d(a, j) = \infty$

$\hat{d}(u, v) = \max(d(u, v), d(v, u))$ is a *distance*:

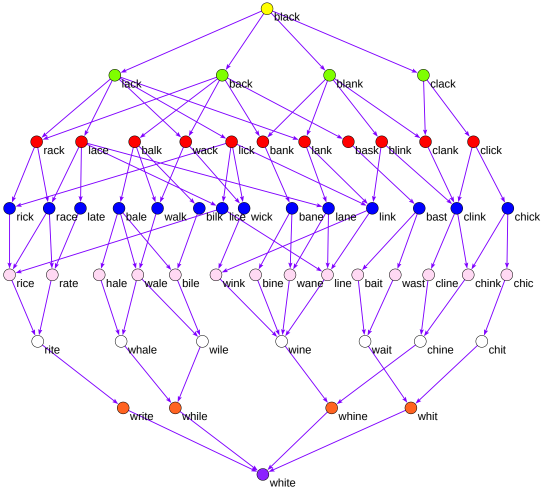
$\hat{d}(v, v) = 0$, $\hat{d}(u, v) = \hat{d}(v, u)$,

$\hat{d}(u, v) \leq \hat{d}(u, t) + \hat{d}(t, v)$.

The *diameter* of a graph equals to the distance between the most distant pair of nodes: $D = \max_{u,v \in \mathcal{V}} d(u, v)$.

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

Shortest paths



DICT28.

Equivalence relations and Partitions

A relation R on $\mathcal{V}$ is an *equivalence* relation iff it is reflexive $\forall v \in \mathcal{V} : vRv$, symmetric $\forall u, v \in \mathcal{V} : uRv \Rightarrow vRu$, and transitive $\forall u, v, z \in \mathcal{V} : uRz \wedge zRv \Rightarrow uRv$.

Each equivalence relation determines a partition into *equivalence classes* $[v] = \{u : vRu\}$.

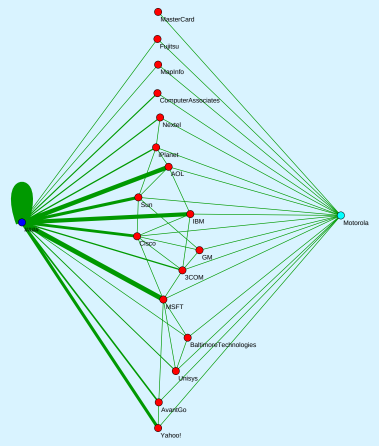
Each partition **C** determines an equivalence relation $uRv \Leftrightarrow \exists C \in \mathbf{C} : u \in C \wedge v \in C$.

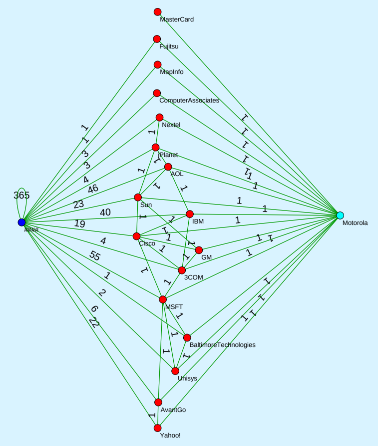
k-neighbors of v is the set of nodes on 'distance' k from v, $N^k(v) = \{u \in v : d(v, u) = k\}$.
The set of all k-neighbors, $k = 0, 1, \ldots$ of v is a partition of $\mathcal{V}$.

k-neighborhood of v, $N^{(k)}(v) = \{u \in v : d(v, u) \leq k\}$.

`Network/Create Partition/k-Neighbors`

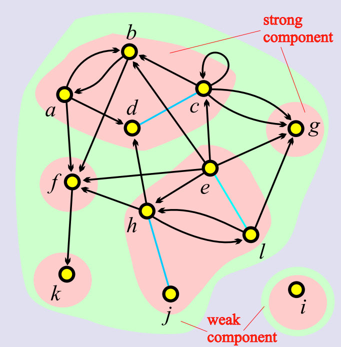
Motorola’s neighborhood





The thickness of edges is a square root of its value.

Connectivity



Node u is *reachable* from node v iff there exists a walk with initial node v and terminal node u.

Node v is *weakly connected* with node u iff there exists a semiwalk with v and u as its end-nodes.

Node v is *strongly connected* with node u iff they are mutually reachable.

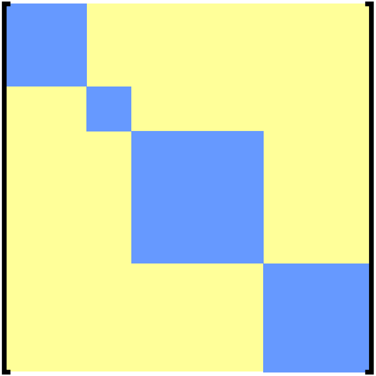
Weak and strong connectivity are equivalence relations. Equivalence classes induce weak/strong *components*.

`Network/Create Partition/Components/`

Connectivity in igraph

```
> wdir <- "C:/.../2017/Moscow/Rnet/test"
> setwd(wdir)
> library(igraph)
> source("C:\\...\\Rnet\\test\\igraph+.R")
> R <- read.graph("./nets/class.net",format="pajek")
> vertex_attr(R)$shape <- NULL
> plot(R)
> w <- components(R,mode="weak")
> w
> s <- components(R,mode="strong")
> s
> V(R)$strong <- s$membership
> col <- c("red","green","orange","blue","green","magenta",
  "grey","black")
> plot(R,vertex.color=col[s$membership])
> main <- extract_clusters(R,"strong",c(4))
> plot(main)
```

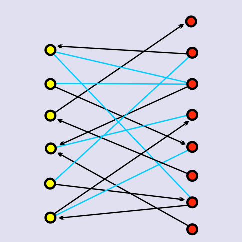
Weak components

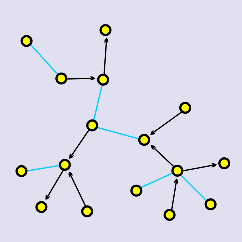


Reordering the nodes of network such that the nodes from the same class of weak partition are put together we get a matrix representation consisting of diagonal blocks – weak components.

Most problems can be solved separately on each component and afterward these solutions combined into final solution.

Special graphs – bipartite, tree

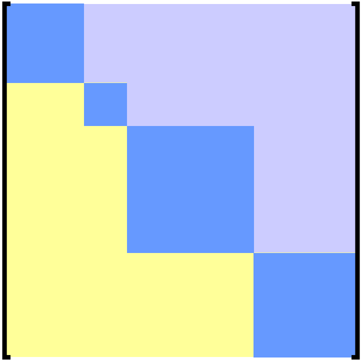


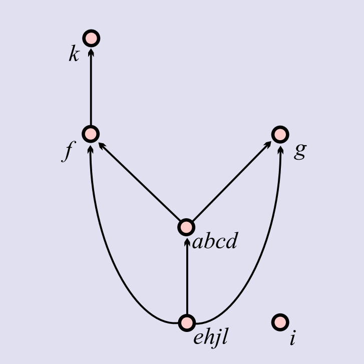


A graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ is *bipartite* iff its set of nodes $\mathcal{V}$ can be partitioned into two sets $\mathcal{V}_1$ and $\mathcal{V}_2$ such that every link from $\mathcal{L}$ has one end-node in $\mathcal{V}_1$ and the other in $\mathcal{V}_2$.

A weakly connected graph $\mathcal{G}$ is a *tree* iff it doesn't contain loops and semicycles of length at least 3.

Condensation

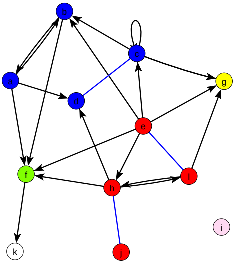


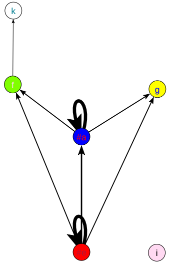


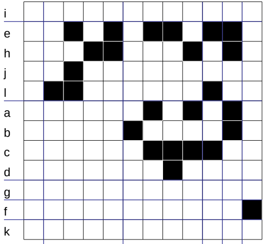
If we shrink every strong component of a given graph into a node, delete all loops and identify parallel arcs the obtained *reduced* graph is acyclic. For every acyclic graph an *ordering* / *level* function $i : \mathcal{V} \to \mathbb{N}$ exists s.t. $(u, v) \in \mathcal{A} \Rightarrow i(u) < i(v)$.

Condensation – Example

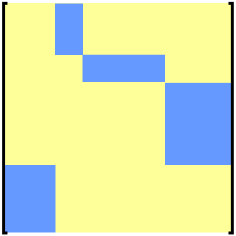
```
Network/Create Partition/Components/Strong [1]
Operations/Network+Partition/Shrink Network  [1][0]
Network/Create New Network/Transform/Remove/Loops [yes]
Network/Acyclic Network/Depth Partition/Acyclic
Partition/Make Permutation
Permutation/Inverse Permutation
select partition [Strong Components]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

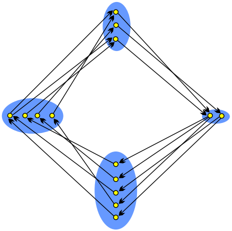






Internal structure of strong components





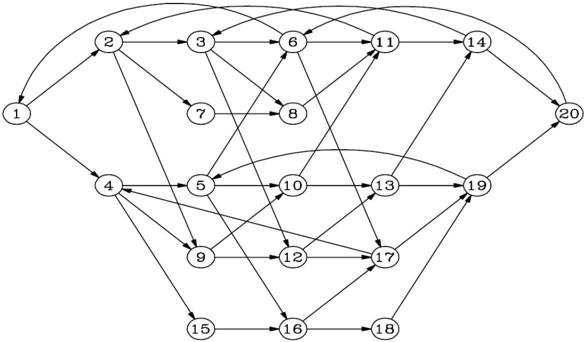
Let d be the largest common divisor of lengths of closed walks in a strong component.

The component is said to be *simple*, iff $d = 1$; otherwise it is *periodical* with a period d.

The set of nodes $\mathcal{V}$ of strongly connected directed graph $\mathcal{G} = (\mathcal{V}, R)$ can be partitioned into d clusters $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_d$, s.t. for every arc $(u, v) \in R$ holds $u \in \mathcal{V}_i \Rightarrow v \in \mathcal{V}_{(i \bmod d)+1}$.

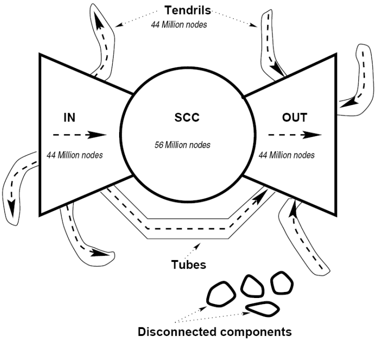
```
Network/Create Partition/
    Components/Strong-Periodic
```

...Internal structure of strong components



Bonhoure's periodical graph. Pajek data

Bow-tie structure of the Web graph



Kumar &: The Web as a graph

Let $\mathcal{S}$ be the *largest strong component* in network $\mathcal{N}$; $\mathcal{W}$ the weak component containing $\mathcal{S}$; $\mathcal{I}$ the set of nodes from which $\mathcal{S}$ can be reached; $\mathcal{O}$ the set of nodes reachable from $\mathcal{S}$; $\mathcal{T}$ (tubes) set of nodes (not in $\mathcal{S}$) on paths from $\mathcal{I}$ to $\mathcal{O}$; $\mathcal{R} = \mathcal{W} \setminus (\mathcal{I} \cup \mathcal{S} \cup \mathcal{O} \cup \mathcal{T})$ (tendrils); and $\mathcal{D} = \mathcal{V} \setminus \mathcal{W}$. The partition

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

is called the *bow-tie* partition of $\mathcal{V}$.

Note: chains can exist in the set $\mathcal{R}$.

`Network/Create Partition/Bow-Tie`

Biconnectivity

Nodes u and v are *biconnected* iff they are connected (in both directions) by two independent (no common internal node) paths. Biconnectivity determines a partition of the set of links.

A node is an *articulation* node iff its deletion increases the number of weak components in a graph.

A link is a *bridge* iff its deletion increases the number of weak components in a graph.

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

The notion of biconnectivity can be generalized do k-connectivity. No very efficient algorithm for $k > 3$ exists.

k-connectivity

Node connectivity κ of graph $\mathcal{G}$ is equal to the smallest number of nodes that, if deleted, induce a disconnected graph or the trivial graph K_1.

Link connectivity λ of graph $\mathcal{G}$ is equal to the smallest number of links that, if deleted, induce a disconnected graph or the trivial graph K_1.

Whitney's inequality: $\kappa(\mathcal{G}) \leq \lambda(\mathcal{G}) \leq \delta(\mathcal{G})$.

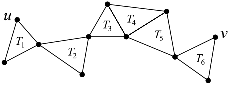
Graph $\mathcal{G}$ is *(node) $k-$connected*, if $\kappa(\mathcal{G}) \geq k$ and is *link $k-$connected*, if $\lambda(\mathcal{G}) \geq k$.

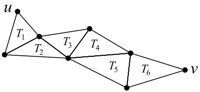
Whitney / Menger theorem: Graph $\mathcal{G}$ is node/link $k-$connected iff every pair of nodes can be connected with k node/link internally disjoint (semi)walks.

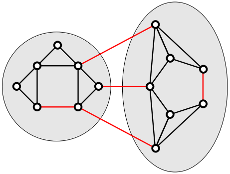
Triangular and short cycle connectivities

In an undirected graph we call a *triangle* a subgraph isomorphic to K_3.

A sequence $(T_1, T_2, \ldots, T_s)$ of triangles of $\mathcal{G}$ *(node) triangularly connects* nodes $u, v \in \mathcal{V}$ iff $u \in T_1$ and $v \in T_s$ or $u \in T_s$ and $v \in T_1$ and $\mathcal{V}(T_{i-1}) \cap \mathcal{V}(T_i) \neq \emptyset$, $i = 2, \ldots s$. It *edge triangularly connects* nodes $u, v \in \mathcal{V}$ iff a stronger version of the second condition holds $\mathcal{E}(T_{i-1}) \cap \mathcal{E}(T_i) \neq \emptyset$, $i = 2, \ldots s$.



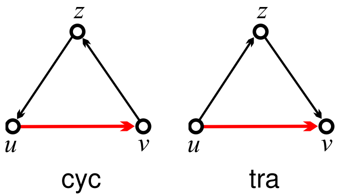




Node triangular connectivity is an equivalence on $\mathcal{V}$; and edge triangular connectivity is an equivalence on $\mathcal{E}$. See the paper.

Triangular connectivity in directed graphs

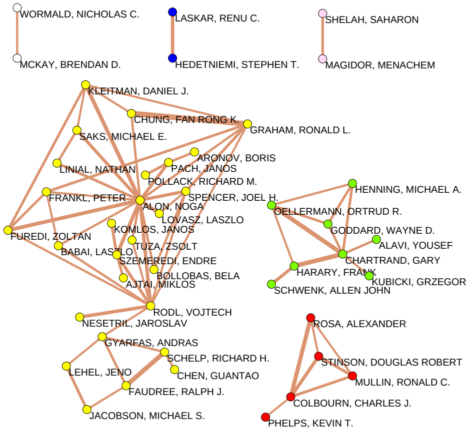
If a graph $\mathcal{G}$ is mixed we replace edges with pairs of opposite arcs. In the following let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ be a simple directed graph without loops. For a selected arc $(u, v) \in \mathcal{A}$ there are only two different types of directed triangles: **cyc**lic and **tra**nsitive.



For each type we get the corresponding triangular network $\mathcal{N}_{cyc}$ and $\mathcal{N}_{tra}$ by determining the corresponding weight w_{cyc} or w_{tra} to its arcs, counting the number of cyclic/transitive triangles that contain the arc. We remove arcs with weight zero. The notion of triangular connectivity can be extended to the notion of *short (semi) cycle connectivity*.

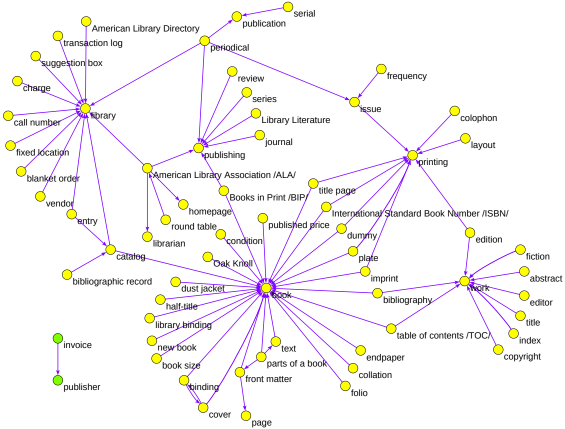
`Network/Create New Network/with Ring Counts/3-Rings/Directed`

Edge-cut at level 16 of triangular network of Erdős collaboration graph



without Erdős,
$n = 6926$,
$m = 11343$

Arc-cut at level 11 of transitive triangular network of ODLIS dictionary



Important nodes in network

To identify important / interesting elements (nodes, links) in a network we often try to express our intuition about important / interesting element using an appropriate measure (index, weight) following the scheme

> *larger is the measure value of an element,*
> *more important / interesting is this element*

Too often, in analysis of networks, researchers uncritically pick some measure from the literature. For formal approach see Roberts.

It seems that the most important distinction between different node *indices* is based on the view/decision whether the network is considered directed or undirected.

. . . Important nodes in network

This gives us two main types of indices:

- networks containing directed links (we replace edges by pairs of opposite arcs): measures of *importance*; with two subgroups: measures of *influence*, based on out-going arcs; and measures of *support*, based on incoming arcs;
- measures of *centrality*, based on all links.

For undirected networks all three types of measures coincide.
If we change the direction of all arcs (replace the relation with its inverse relation) the measure of influence becomes a measure of support, and vice versa.

. . . Important nodes in network

The real meaning of measure of importance depends on the relation described by a network. For example the most 'important' person for the relation '__ doesn't like to work with __' is in fact the least popular person.

Removal of an important node/link from a network produces a substantial change in structure/functioning of the network.

Normalization

Let $p : \mathcal{V} \to \mathbb{R}$ be an index in network $\mathcal{N} = (\mathcal{V}, \mathcal{L}, p)$. If we want to compare indices p over different networks we have to make them comparable. Usually we try to achieve this by *normalization* of p. Let $\mathcal{N} \in \mathbf{N}(\mathcal{V})$, where $\mathbf{N}(\mathcal{V})$ is a selected family of networks over the same set of nodes V,

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

then we define the normalized index as

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

Degrees

The simplest index are the degrees of nodes. Since for simple networks $\deg_{min} = 0$ and $\deg_{max} = n - 1$, the corresponding normalized indices are

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

and similary

support $$\mathrm{indeg}'(v) = \frac{\mathrm{indeg}(v)}{n}$$

influence $$\mathrm{outdeg}'(v) = \frac{\mathrm{outdeg}(v)}{n}$$

Instead of degrees in original network we can consider also the degrees with respect to the reachability relation (transitive closure).

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

Closeness

Most indices are based on the distance $d(u, v)$ between nodes in a network $\mathcal{N} = (\mathcal{V}, \mathcal{L})$. Two such indices are

radius $\quad r(v) = \max_{u \in \mathcal{V}} d(v, u)$

total closeness $\quad S(v) = \sum_{u \in \mathcal{V}} d(v, u)$

These two indices are measures of influence – to get measures of support we have to replace in definitions $d(u, v)$ with $d(v, u)$.

If the network is not strongly connected r_{max} and S_{max} are equal to ∞. Sabidussi (1966) introduced a related measure $1/S(v)$; or in its normalized form

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

$D = \max_{u,v \in \mathcal{V}} d(v, u)$ is called the *diameter* of network.

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

Betweeness

Important are also the nodes that can control the information flow in the network. If we assume that this flow uses only the shortest paths (geodesics) we get a measure of *betweeness* (Anthonisse 1971, Freeman 1977)

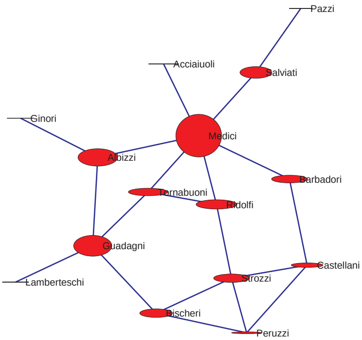
$$b(v) = \frac{1}{(n-1)(n-2)} \sum_{\substack{u,t \in \mathcal{V}: g_{u,t} > 0 \\ u \neq v, t \neq v, u \neq t}} \frac{g_{u,t}(v)}{g_{u,t}}$$

where $g_{u,t}$ is the number of geodesics from u to t; and $g_{u,t}(v)$ is the number of those among them that pass through node v.

For computation of geodesic matrix see Brandes.

`Network/Create Vector/Centrality/Betweenness`

Padgett's Florentine families



		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

Hubs and authorities

To each node v of a network $\mathcal{N} = (\mathcal{V}, \mathcal{L})$ we assign two values: quality of its content (*authority*) x_v and quality of its references (*hub*) y_v.

A good authority is selected by good hubs; and good hub points to good authorities (see Kleinberg).

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

Let $\mathbf{W}$ be a matrix of network $\mathcal{N}$ and $\mathbf{x}$ and $\mathbf{y}$ authority and hub vectors. Then we can write these two relations as $\mathbf{x} = \mathbf{W}^T\mathbf{y}$ and $\mathbf{y} = \mathbf{W}\mathbf{x}$.

We start with $\mathbf{y} = [1, 1, \ldots, 1]$ and then compute new vectors $\mathbf{x}$ and $\mathbf{y}$. After each step we normalize both vectors. We repeat this until they stabilize.

We can show that this procedure converges. The limit vector $\mathbf{x}^*$ is the principal eigen vector of matrix $\mathbf{W}^T\mathbf{W}$; and $\mathbf{y}^*$ of matrix $\mathbf{W}\mathbf{W}^T$.

... Hubs and authorities

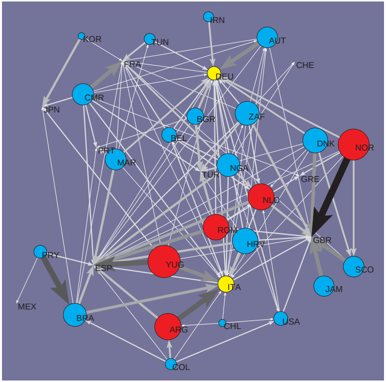
Similar procedures are used in search engines on the web to evaluate the importance of web pages.
PageRank, PageRank / Google, HITS / AltaVista, SALSA, theory.

```
Network/Create New Network/Subnetwork with Paths/Info on
Network/Create Vector/Centrality/Closeness
Network/Create Vector/Centrality/Betweeness
Network/Create Vector/Centrality/Hubs-Authorities
Network/Create Vector/Centrality/Clustering Coefficients
```

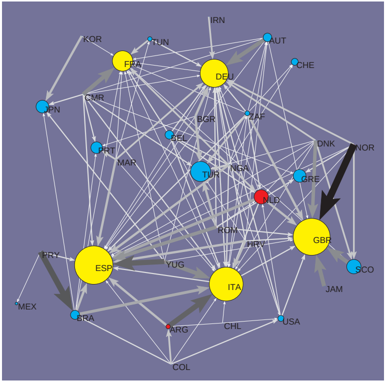
Examples: Krebs, Krempl. World Cup 1998 in Paris, 22 national teams. A player from first country is playing in the second country.

There exist other measures based on eigen-values and eigen-vectors such as Katz, Bonachich and Brandes. See also Borgatti.

. . . Hubs and authorities: football



Exporters (hubs)



Importers (authorities)

Clustering

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be simple undirected graph. *Clustering* in node v is usually measured as a quotient between the number of links in subgraph $\mathcal{G}^1(v) = \mathcal{G}(N^1(v))$ induced by the neighbors of node v and the number of links in the complete graph on these nodes:

$$C(v) = \begin{cases} \dfrac{2|\mathcal{L}(\mathcal{G}^1(v))|}{\deg(v)(\deg(v)-1)} & \deg(v) > 1 \\ 0 & \text{otherwise} \end{cases}$$

We can consider also the size of node neighborhood by the following correction

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

where Δ is the maximum degree in graph $\mathcal{G}$. This measure attains its largest value in nodes that belong to an isolated clique of size Δ.

`Network/Create Vector/Clustering`

User defined indices

Xingqin Qi et al. defined in their paper Terrorist Networks, Network Energy and Node Removal a new measure of centrality based on Laplacian energy – *Laplacian centrality*

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

```
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]
```

macro

Network centralization measures

Extremal approach: Let $p : \mathcal{V} \to \mathbb{R}$ be an index in network $\mathcal{N} = (\mathcal{V}, \mathcal{L})$. We introduce the quantities

$$\begin{aligned} p^\star &= \max_{v \in \mathcal{V}} p(v) \\ D &= \sum_{v \in \mathcal{V}} (p^\star - p(v)) \\ D^\star &= \max\{D(\mathcal{N}) : \mathcal{N} \text{ is a network on } \mathcal{V}\} \end{aligned}$$

Then we can define *centralization* with respect to p

$$C_p(\mathcal{N}) = \frac{D(\mathcal{N})}{D^\star}$$

Usually the most centralized graph is the star S_n and the least centralized is the complete graph K_n.

. . . Network centralization measures

Variational approach: The other approach is based on the variance. First we compute the average node centrality

$$\overline{p} = \frac{1}{n} \sum_{v \in \mathcal{V}} p(v)$$

and then define

$$V_p(\mathcal{N}) = \frac{1}{n} \sum_{v \in \mathcal{V}} (p(v) - \overline{p})^2$$

Important nodes in igraph

```
> R <- read.graph("./nets/class.net",format="pajek")
> vertex_attr(R)$shape <- NULL
> b <- betweenness(R,normalized=TRUE)
> plot(R,vertex.size=b*100)
> c <- closeness(R,normalized=TRUE)
> plot(R,vertex.size=c*100)
> e <- eigen_centrality(R)
> plot(R,vertex.size=e$vector*30)
> hub=hub.score(R)$vector
> plot(R,vertex.size=hub*20)
> aut=authority.score(R)$vector
> plot(R,vertex.size=aut*20)
> b <- bonpow(R,rescale=TRUE)
> plot(R,vertex.size=b*200)
> # clustering coefficient
> t <- transitivity(R,type="local")
> plot(R,vertex.size=t*25)
```