OSU CS419/519 Document Analysis

Social Network Analysis

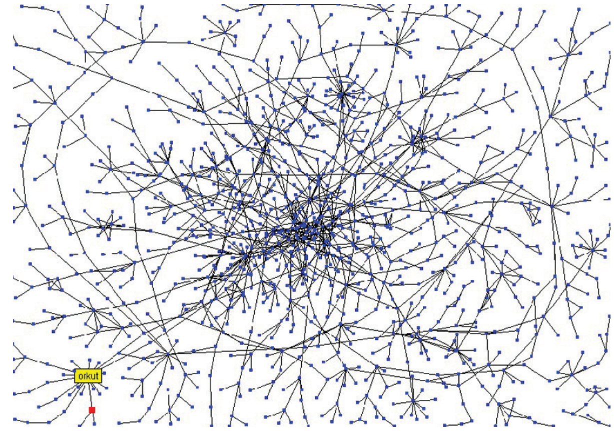
Centrality, Connected Components, Communities and Pagerank

Marian-Andrei Rizoiu, Lexing Xie
Computer Science, ANU

Lecture slides credit: Lada Adamic, Univ. Michigan, Jure Leskovec, Stanford University

---

Center of the network: is counting the edges enough?

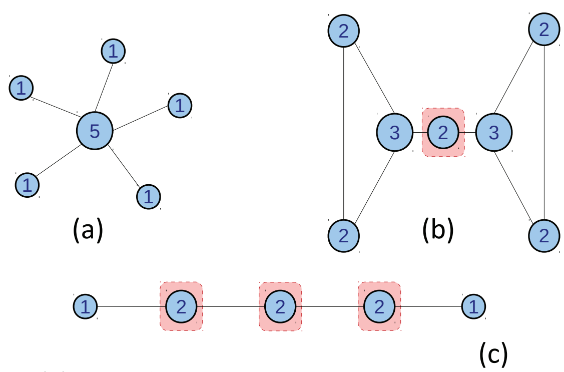


Stanford Social Web (ca. 1999) network of personal homepages at Stanford

---

What does degree not capture?

In what ways does degree fail to capture centrality in the following graphs?

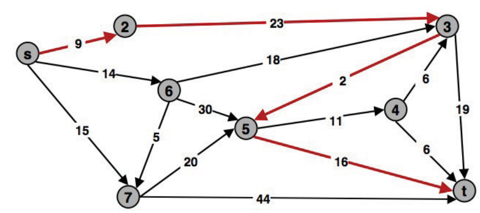


Review: shortest path in a network

Shortest path network: (V, E, s, t, c) .

- **Directed graph (V, E).**
- **Source s ∈ V, sink t ∈ V.**
- **Arc costs c(v, w).**
- **Cost of path = sum of arc costs in path.**

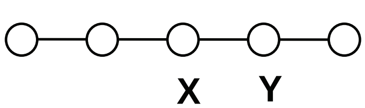
Cost of path s - 2 - 3 - 5 - t
= 9 + 23 + 2 + 16
= 48.



by Wayne Wolf, Princeton University

Betweenness: *centrality capturing brokerage*

- intuition: how many pairs of individuals would have to go through you in order to reach one another in the minimum number of hops?



Betweenness: definition

$$C_B(i) = \sum_{j<k} g_{jk}(i) / g_{jk}$$

Where g_{jk} = the number of shortest paths connecting *jk*
$g_{jk}(i)$ = the number that vertex *i* is on.

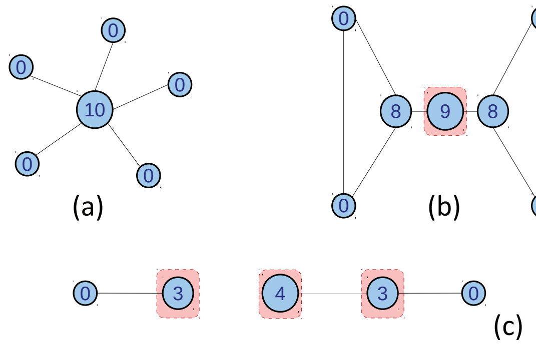
Usually normalized by:

$$C'_B(i) = C_B(i) / \boxed{[(n-1)(n-2)/2]}$$

number of pairs of vertices excluding the vertex itself

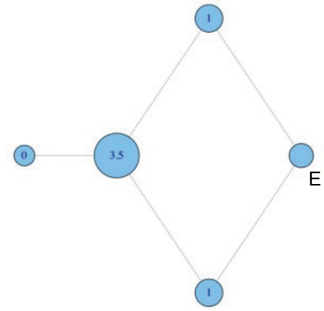
Betweenness: revisiting examples

The values of betweennes on the above toy examples.
Observe the different values for highlighted nodes.



Betweenness: Quiz Question #1

- What is the betweenness of node E?



a) 0.5

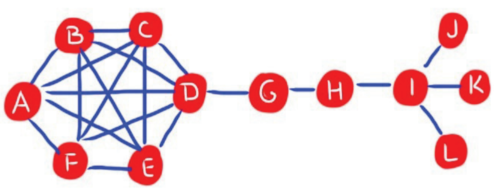
b) 1

c) 1.5

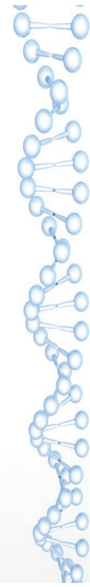
d) 2

Betweenness: Quiz Question #2

- Find a node that has high betweenness, but low degree

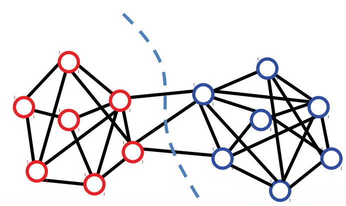


- Find a node that has low betweenness, but high degree



Community detection

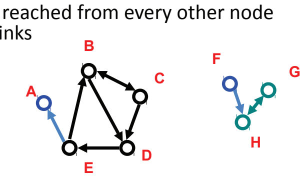
Can we discover community structure in an automated way?



Connected components

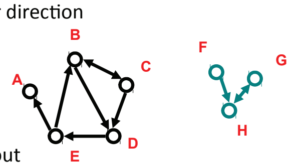
- Strongly connected components
  – Each node within the component can be reached from every other node in the component by following directed links

- Strongly connected components
  - B C D E
  - A
  - G H
  - F



- Weakly connected components: every node can be reached from every other node by following links in either direction

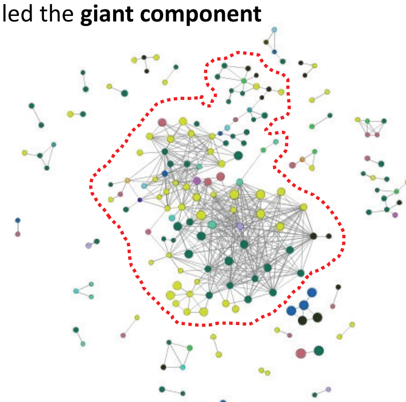
  - Weakly connected components
    - A B C D E
    - G H F



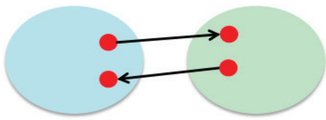
- In undirected networks one talks simply about 'connected components'

Giant component

- if the largest component encompasses a significant fraction of the graph, it is called the **giant component**

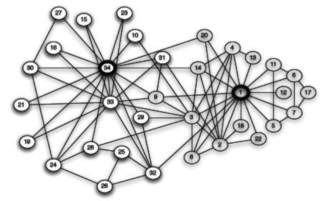


- There is a giant SCC
- There won't be 2 giant SCCs: Why not?
  - Just takes 1 page from one SCC to link to the other SCC
  - If the components have millions of pages the likelihood of this is very large

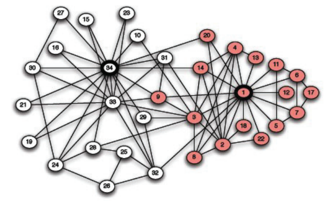


Giant SCC1 Giant SCC2

Splitting Zachary Karate Club



(a) *Karate club network*

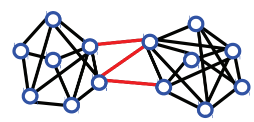


(b) *After a split into two clubs*

betweenness clustering

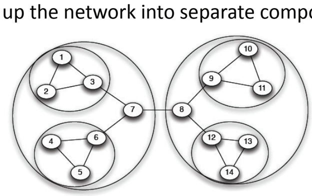
- Algorithm
  - ➢ compute the betweenness of all edges
  - ➢ while (betweenness of any edge > threshold):
    - ✗ remove edge with highest betweenness
    - ✗ recalculate betweenness
- Betweenness needs to be recalculated at each step
  - ➢ removal of an edge can impact the betweenness of another edge
  - ➢ very expensive: all pairs shortest path – $O(N^3)$
  - ➢ may need to repeat up to N times
  - ➢ does not scale to more than a few hundred nodes, even with the fastest algorithms

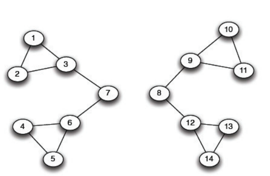
betweenness clustering algorithm



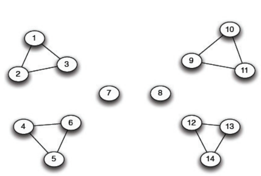
betweenness clustering:

- successively remove edges of highest betweenness (the bridges, or local bridges), breaking up the network into separate components



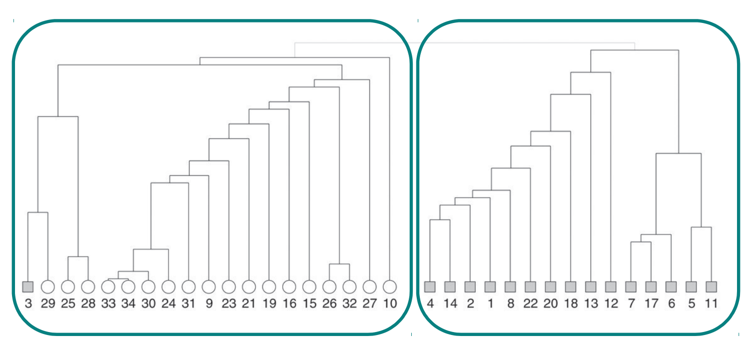


(a) *Step 1*



(b) *Step 2*

betweenness clustering algorithm & the karate club data set



source: Girvan and Newman, PNAS June 11, 2002 99(12):7821-7826

Google's PageRank (Brin/Page 98)

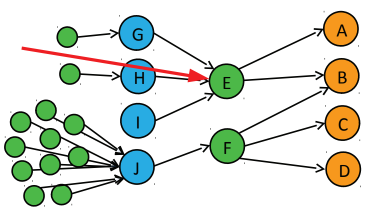
Google's PageRank (Brin/Page 98)

- A technique for estimating page quality
  - Based on web link graph
- Results are combined with IR score
  - Think of it as: TotalScore = IR score * PageRank
  - In practice, search engines use many other factors
  - (for example, Google says it uses more than 200)

PageRank: Intuition

Shouldn't E's vote be worth more than F's?

How many levels should we consider?



- Imagine a contest for The Web's Best Page
  - Initially, each page has one vote
  - Each page votes for all the pages it has a link to
  - To ensure fairness, pages voting for more than one page must split their vote equally between them
  - Voting proceeds in rounds; in each round, each page has the number of votes it received in the previous round
  - In practice, it's a little more complicated - *but not much!*

PageRank

- Each page i is given a rank x_i
- **Goal:** Assign the x_i such that the rank of each page is governed by the ranks of the pages linking to it:

$$x_i = \sum_{j \in B_i} \frac{1}{N_j} x_j$$

Rank of page i

Every page j that links to i

Number of links out from page j

Rank of page j

How do we compute the rank values?

Iterative PageRank (simplified)

Initialize all ranks to be equal, e.g.:

$$x_i^{(0)} = \frac{1}{n}$$

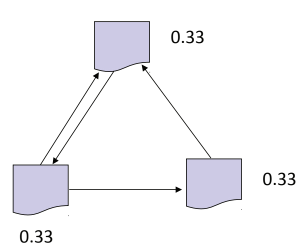
Iterate until convergence

$$x_i^{(k+1)} = \sum_{j \in B_i} \frac{1}{N_j} x_j^{(k)}$$

Example: Step 0

Initialize all ranks to be equal

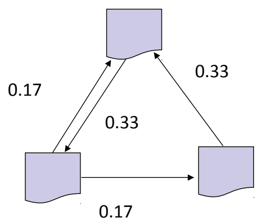
$$x_i^{(0)} = \frac{1}{n}$$



Example: Step 1

Propagate weights across out-edges

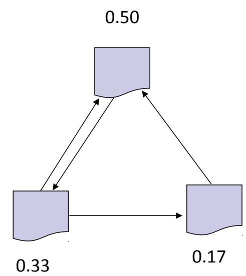
$$x_i^{(k+1)} = \sum_{j \in B_i} \frac{1}{N_j} x_j^{(k)}$$



Example: Step 2

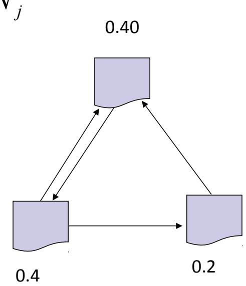
Compute weights based on in-edges

$$x_i^{(1)} = \sum_{j \in B_i} \frac{1}{N_j} x_j^{(0)}$$



Example: Convergence

$$x_i^{(k+1)} = \sum_{j \in B_i} \frac{1}{N_j} x_j^{(k)}$$



Naïve PageRank Algorithm Restated

- Let
  - *N(p)* = number outgoing links from page p
  - B_p = set of pages that back-link to page p

$$PageRank(p) = \sum_{b \in B_p} \frac{1}{N(b)} PageRank(b)$$

  - Each page b distributes its importance to all of the pages it points to (so we scale by 1/N(b))
  - Page p's importance is increased by the importance of its back set

In Linear Algebra formulation

– Create an m x m matrix M to capture links:

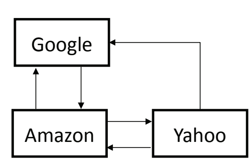
$$M(i,j) = \begin{cases} 1/n_j, & \text{if page } i \text{ is pointed to by page } j \text{ and page } j \text{ has } n_j \text{ outgoing links} \\ 0, & \text{otherwise} \end{cases}$$

– Initialize all PageRanks to 1, multiply by M repeatedly until all values converge:

$$\begin{bmatrix} PageRank(p_1') \\ PageRank(p_2') \\ \dots \\ PageRank(p_m') \end{bmatrix} = M \begin{bmatrix} PageRank(p_1) \\ PageRank(p_2) \\ \dots \\ PageRank(p_m) \end{bmatrix}$$

– Computes principal eigenvector via power iteration

A Brief Example



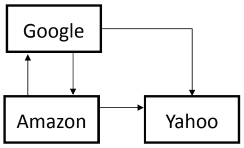
$$\begin{bmatrix} g' \\ y' \\ a' \end{bmatrix} = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0 & 0 & 0.5 \\ 1 & 0.5 & 0 \end{bmatrix} * \begin{bmatrix} g \\ y \\ a \end{bmatrix}$$

Running for multiple iterations:

$$\begin{bmatrix} g \\ y \\ a \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0.5 \\ 1.5 \end{bmatrix}, \begin{bmatrix} 1 \\ 0.75 \\ 1.25 \end{bmatrix}, \dots \begin{bmatrix} 1 \\ 0.67 \\ 1.33 \end{bmatrix}$$

Total rank sums to number of pages

Oops #1 – PageRank Sinks



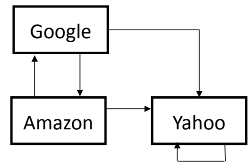
$$\begin{bmatrix} g' \\ y' \\ a' \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0 \end{bmatrix} * \begin{bmatrix} g \\ y \\ a \end{bmatrix}$$

'dead end' - PageRank is lost after each round

Running for multiple iterations:

$$\begin{bmatrix} g \\ y \\ a \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.5 \\ 1 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 0.25 \\ 0.5 \\ 0.25 \end{bmatrix}, \ldots, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Oops #2 – PageRank hogs



$$\begin{bmatrix} g' \\ y' \\ a' \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0 & 0 \end{bmatrix} * \begin{bmatrix} g \\ y \\ a \end{bmatrix}$$

PageRank cannot flow out and accumulates

Running for multiple iterations:

$$\begin{bmatrix} g \\ y \\ a \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.5 \\ 2 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 0.25 \\ 2.5 \\ 0.25 \end{bmatrix}, \ldots, \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix}$$

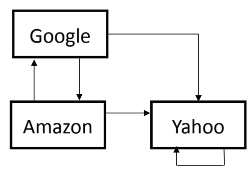
Improved PageRank

- Remove out-degree 0 nodes (or consider them to refer back to referrer)
- Add decay factor d to deal with sinks

$$PageRank(p) = (1-d) + d \sum_{b \in B_p} \frac{1}{N(b)} PageRank(b)$$

- Typical value: d=0.85

Stopping the Hog



$$\begin{bmatrix} g' \\ y' \\ a' \end{bmatrix} = 0.85 \begin{bmatrix} 0 & 0 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0 & 0 \end{bmatrix} * \begin{bmatrix} g \\ y \\ a \end{bmatrix} + \begin{bmatrix} 0.15 \\ 0.15 \\ 0.15 \end{bmatrix}$$

Running for multiple iterations:

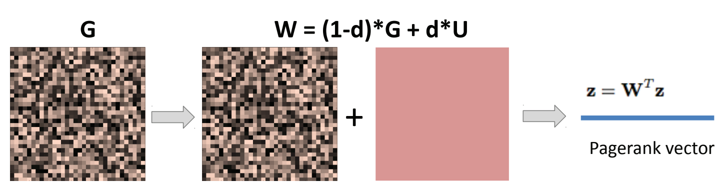
$$\begin{bmatrix} g \\ y \\ a \end{bmatrix} = \begin{bmatrix} 0.57 \\ 1.85 \\ 0.57 \end{bmatrix}, \begin{bmatrix} 0.39 \\ 2.21 \\ 0.39 \end{bmatrix}, \begin{bmatrix} 0.32 \\ 2.36 \\ 0.32 \end{bmatrix}, \ldots, \begin{bmatrix} 0.26 \\ 2.48 \\ 0.26 \end{bmatrix}$$

… though does this seem right?

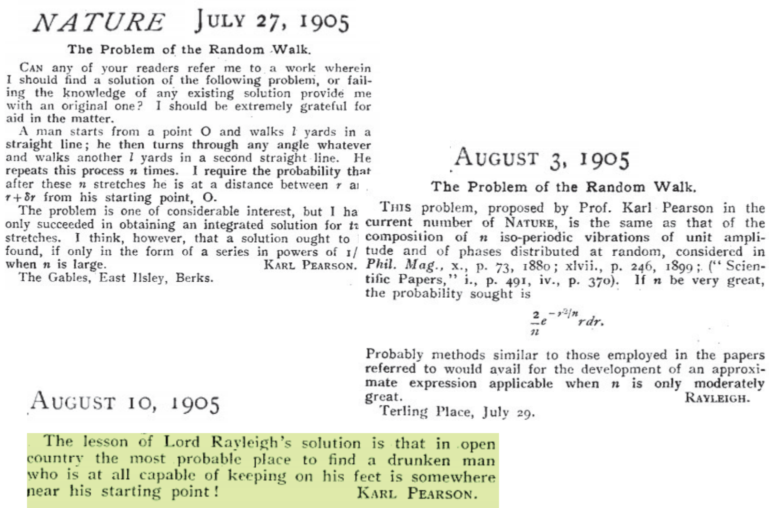
Random Surfer Model

- PageRank has an intuitive basis in random walks on graphs
- Imagine a random surfer, who starts on a random page and, in each step,
  - with probability d, clicks on a random link on the page
  - with probability 1-d, jumps to a random page (bored?)
- The PageRank of a page can be interpreted as the fraction of steps the surfer spends on the corresponding page
  - Transition matrix can be interpreted as a Markov Chain

PageRank, random walk on graphs



- with probability *d*, clicks on a random link on the page
- with probability *1-d*, jumps to a random page (bored?)
- Transition matrix *W* can be interpreted as a Markov Chain



NATURE JULY 27, 1905

The Problem of the Random Walk.

CAN any of your readers refer me to a work wherein I should find a solution of the following problem, or failing the knowledge of any existing solution provide me with an original one? I should be extremely grateful for aid in the matter.

A man starts from a point O and walks l yards in a straight line; he then turns through any angle whatever and walks another l yards in a second straight line. He repeats this process n times. I require the probability that after these n stretches he is at a distance between r a $r+\delta r$ from his starting point, O.

The problem is one of considerable interest, but I ha only succeeded in obtaining an integrated solution for t stretches. I think, however, that a solution ought to found, if only in the form of a series in powers of 1/ when n is large. KARL PEARSON.

The Gables, East Ilsley, Berks.

AUGUST 3, 1905

The Problem of the Random Walk.

THIS problem, proposed by Prof. Karl Pearson in the current number of NATURE, is the same as that of the composition of n iso-periodic vibrations of unit amplitude and of phases distributed at random, considered in *Phil. Mag.*, x., p. 73, 1880; xlvii., p. 246, 1899; ("Scientific Papers," i., p. 491, iv., p. 370). If n be very great, the probability sought is

$$\frac{2}{n} e^{-r^2/n} r\,dr.$$

Probably methods similar to those employed in the papers referred to would avail for the development of an approximate expression applicable when n is only moderately great. RAYLEIGH.

Terling Place, July 29.

AUGUST 10, 1905

The lesson of Lord Rayleigh's solution is that in open country the most probable place to find a drunken man who is at all capable of keeping on his feet is somewhere near his starting point! KARL PEARSON.

Search Engine Optimization (SEO)

- Has become a big business
- White-hat techniques
  - Google webmaster tools
  - Add meta tags to documents, etc.
- Black-hat techniques
  - Link farms
  - Keyword stuffing, hidden text, meta-tag stuffing, ...
  - Spamdexing
    - Initial solution: ...
    - Some people started to abuse this to improve their own rankings
  - Doorway pages / cloaking
    - Special pages just for search engines
    - BMW Germany and Ricoh Germany banned in February 2006
  - Link buying

Recap: PageRank

- Estimates absolute 'quality' or 'importance' of a given page based on inbound links
  - Query-independent
  - Can be computed via fixpoint iteration
  - Can be interpreted as the fraction of time a 'random surfer' would spend on the page
  - Several refinements, e.g., to deal with sinks
- Considered relatively stable
  - But vulnerable to black-hat SEO
- An important factor, but not the only one
  - Overall ranking is based on many factors (Google: >200)

What could be the other 200 factors?

	Positive	Negative
On-page	Keyword in title? URL? Keyword in domain name? Page freshness Rate of change ...	Links to 'bad neighborhood' Keyword stuffing Over-optimization Hidden content (text has same color as background) Automatic redirect/refresh ...
Off-page	High PageRank Anchor text of inbound links Links from authority sites Links from well-known sites Domain expiration date ...	Fast increase in number of inbound links (link buying?) Link farming Different pages user/spider Content duplication ...

- Note: This is entirely speculative!

Summary

- Notions of centrality
  - Betweenness *for undirected graphs*
  - Community detection using betweenness
- Macro structure of networks
  - Strongly and weakly connected components
  - Applications in web/discussion forums etc.
- PageRank centrality and algorithm
  - Another centrality measure *for directed graphs*!
  - Freq. of encounters on random walk = importance
  - Used for ranking webpages