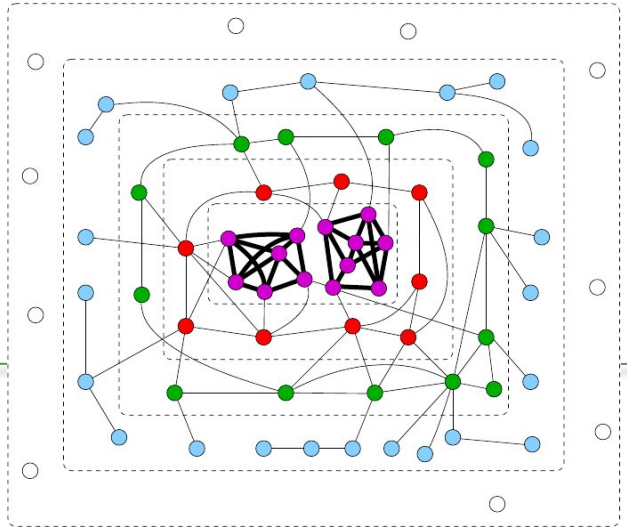
Graph Mining



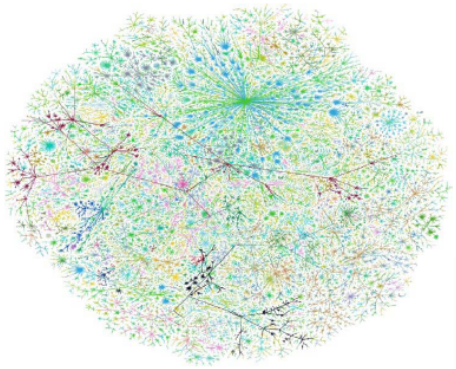
Michalis Vazirgiannis

LIX @ Ecole Polytechnique

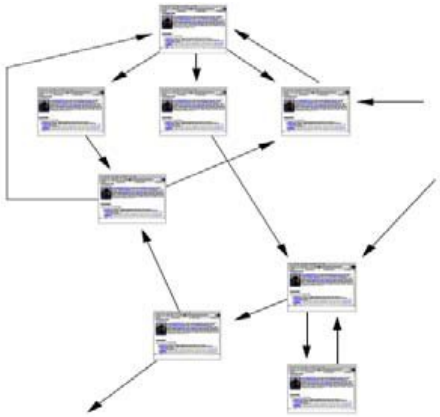
Outline

1. **Introduction & Motivation**
2. Community evaluation measures
3. Graph clustering
4. Graph classification

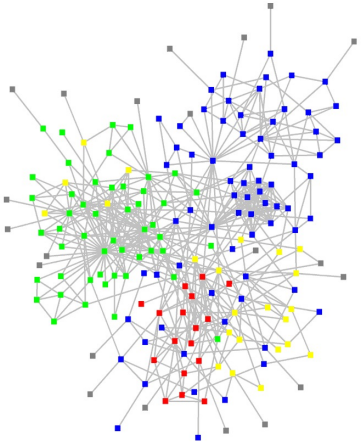
Networks are Everywhere



(a) Internet

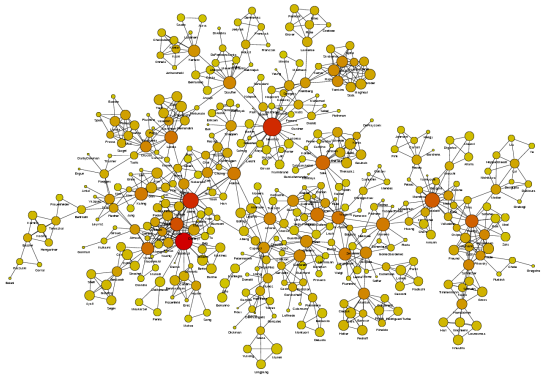


(b) World Wide Web

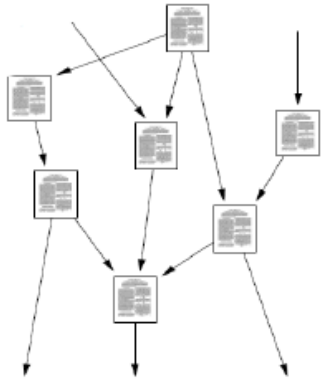


(c) Email network

(d) Social network



(e) Collaboration network



(f) Citation network

Social Networks Growth

- Social networking accounts for 1 of every 6 minutes spent online [http://blog.comscore.com/]
- One out of seven people on Earth is on Facebook
- People on Facebook install 20 million “Apps” every day
- YouTube has more than on billion unique users who visit every month (Oct. 2014)
- Users on YouTube spend a total of 6 billion hours per month (almost an hour for every person on Earth!
- Wikipedia hosts ~34 million articles and has over 91,000 contributors
- 500 million average Tweets per day occur on Twitter (Oct. 2014)

[http://www.jeffbullas.com/2011/09/02/20-stunning-social-media-tatistics/#q3eTJhr64rtD0tLF.99]

Graphs are ubiquitous!

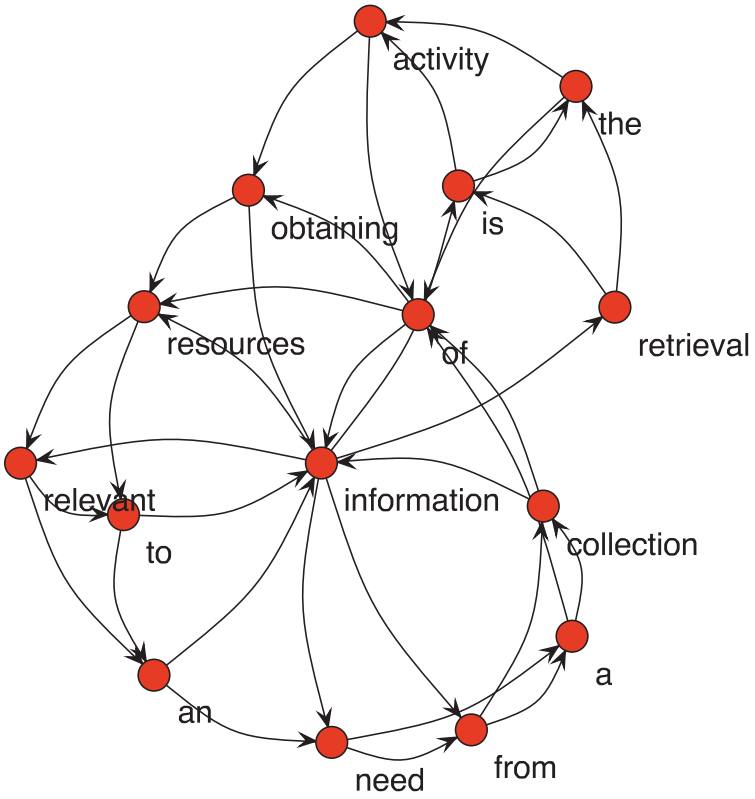
- **Technological networks:**
  - Internet
  - Telephone networks
  - Power grid
  - Road, airline and rail networks
- **Information networks:**
  - World Wide Web
  - Blog networks
  - Citation networks
- **Social networks:**
  - Collaboration networks
  - Organizational networks
  - Communication networks
- **Biological networks:**
  - Networks from Neuroscience
  - Protein-protein interaction networks
  - Gene regulatory networks
  - Food webs
- **Software networks:**
  - Call graphs
  - Software module/component interaction networks

Even representing text - Graph-of-word

information retrieval is the activity of obtaining

information resources relevant to an information need

from a collection of information resources



"Graph of word approach for ad-hoc information retrieval", F. Rousseau, M. Vazirgiannis, Best paper mention award ACM CIKM 2013

How Big are the graphs?

- AS by Skitter (AS-Skitter) - internet topology in 2005 (n = router, m = traceroute)
- LiveJournal (LJ) - social network (n = members, m = friendship)
- U.S. Road Network (USRD) - road network (n = intersections, m = roads)
- Billion Triple Challenge (BTC) - RDF dataset 2009 (n = object, m = relationship)
- WWW of UK (WebUK) - Yahoo Web spam dataset (n = pages, m = hyperlinks)
- Twitter graph (Twitter) - Twitter network (n = users, m = tweets)
- Yahoo! Web Graph (YahooWeb) - WWW pages in 2002 (n = pages, m = hyperlinks

AS-Skitter	**1.7**	**11**	**142 MB**
LJ	**4.8**	**69**	**337.2 MB**
USRD	**24**	**58**	**586.7 MB**
BTC	**165**	**773**	**5.3 GB**
WebUK	**106**	**1877**	**8.6 GB**
Twitter	**42**	**1470**	**24 GB**
YahooWeb	**1413**	**6636**	**120 GB**

Space/Time Complexity

Undirected graph space complexity

Adjaceny matrix: $\Theta(n^2)$

Adjacency list: $\Theta(n+4m)$

Edge List: $O(4m)$

Social scale...
1 billion vertices, 100 billion edges
111 PB adjacency matrix
2.92 TB adjacency list
2.92 TB edge list

Time complexity

Pagerank: $O(n^2)$

all shortest path (Floyd–Warshall algorithm): $O(n^3)$

...

Why Graph Mining

Understand the structure and dynamics of complex interaction systems

- Rich data (in terms of semantics)
- Large scale (big) data

Several application domains

- Community detection
- Web search
- Recommender systems
- Anomaly detection
- Prediction
- ...

Elements of Learning from Graph data

- **Graph models/ graph generators** graph generators (erdos reyni, preferential attachment, kronecker graphs)
- **Node base metrics**: - Ranking algorithms (Pagerank), Ranking evaluation measures (Kendal Tau, NDCG),
- **Graph exploration/preprocessing**: degree distributions, visualization
- **Supervised learning for graphs**: link prediction, graph kernels, graph classification
- **Unsupervised learning:** clustering, community mining, degeneracy.
- **Learning theory in graphs:** model ensembling/selection…

Ranking

- **Ranking in the context of the Web graph**
  - Graph Based ranking
    - Pagerank
    - HITS
  - Pagerank Computation methods

Data are connected!

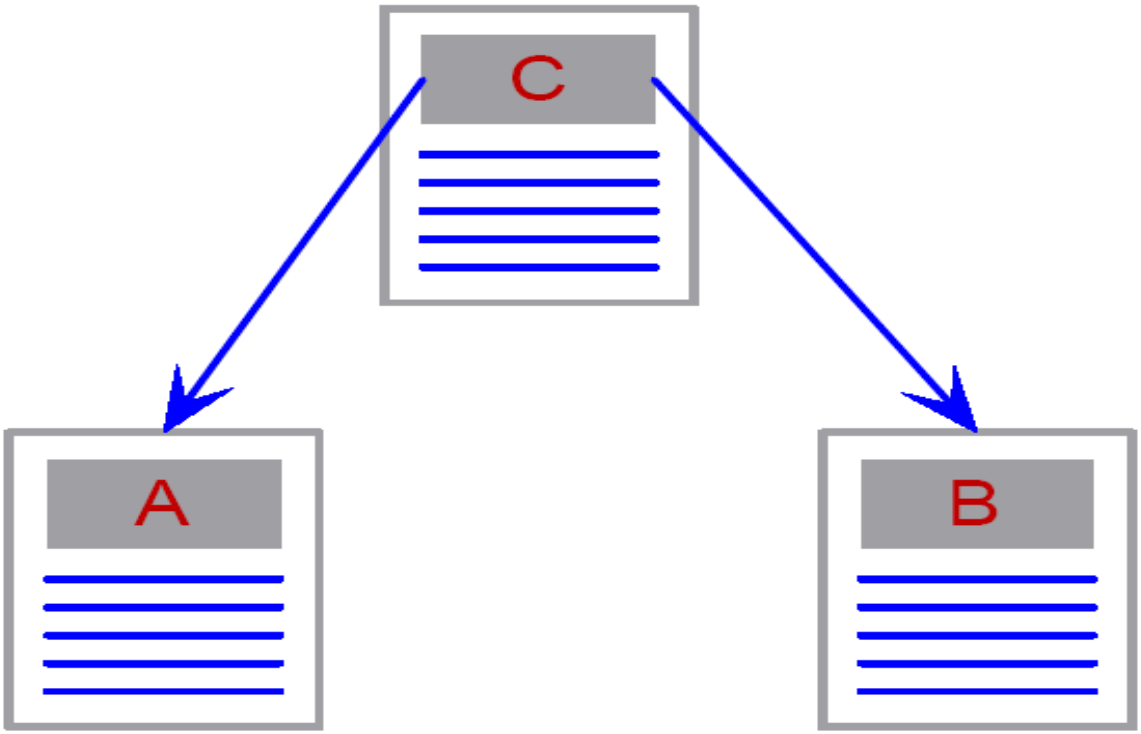
- A.boss = B, B.friends = {C,D,F}, F. follows(N,M,F)
- Social networks, and the web is not just a collection of documents – they form a graph structure
- A link from page *A* to page *B* may indicate:
  - *A* is related to *B*, or
  - *A* is recommending, citing or endorsing *B*
- Links are either
  - referential – *click here and get back home*, or
  - Informational – *click here to get more detail*

Citation Analysis

- The **impact factor** of a journal = *A/B*
  - ***A*** is the number of current year citations to articles appearing in the journal during previous two years.
  - ***B*** is the number of articles published in the journal during previous two years.

Journal Title	Impact Factor (2002)
J. Mach. Learn. Res.	3.818
IEEE T. Pattern Anal.	2.923
Mach. Learn.	1.944
IEEE Intell. Syst.	1.905
Artif. Intell.	1.703

Co-Citation



- *A* and *B* are co-cited by *C*, implying that
  - they are related or associated.
- The strength of co-citation between *A* and *B* is the number of times they are co-cited.

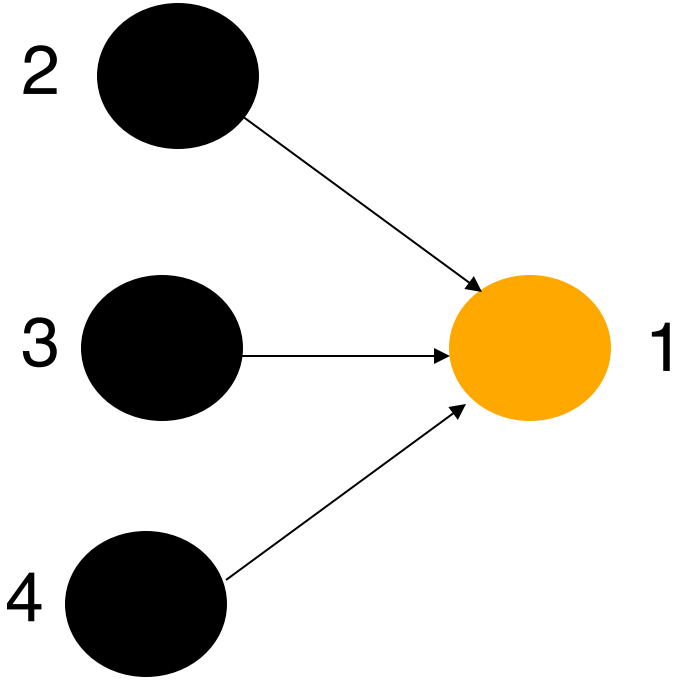
HITS - Kleinberg's Algorithm

- HITS – Hypertext Induced Topic Selection
- For each vertex v Є V in a subgraph of interest:

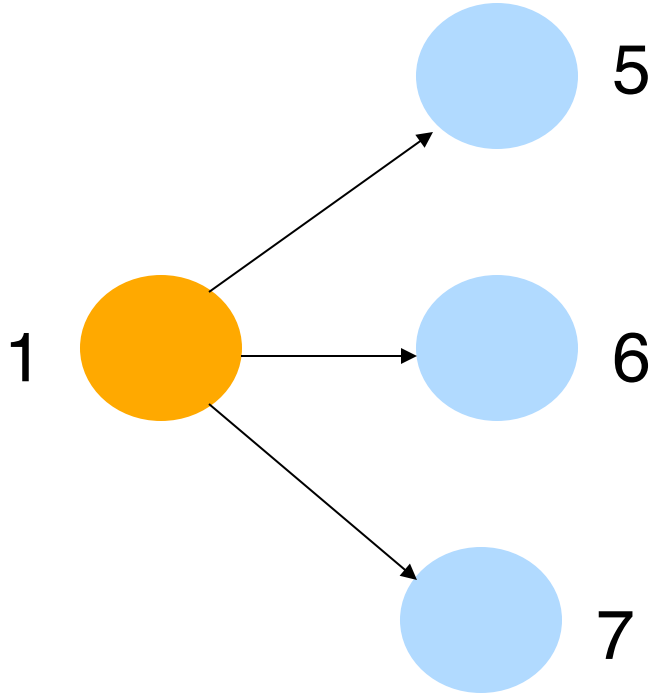
  a(v) - the authority of v
  h(v) - the hubness of v

- A site is very authoritative if it receives many citations. Citation from important sites weight more than citations from less-important sites
- Hubness shows the importance of a site. A good hub is a site that links to many authoritative sites

Authority and Hubness



$a(1) = h(2) + h(3) + h(4)$



$h(1) = a(5) + a(6) + a(7)$

Authority and Hubness Convergence

- Recursive dependency:

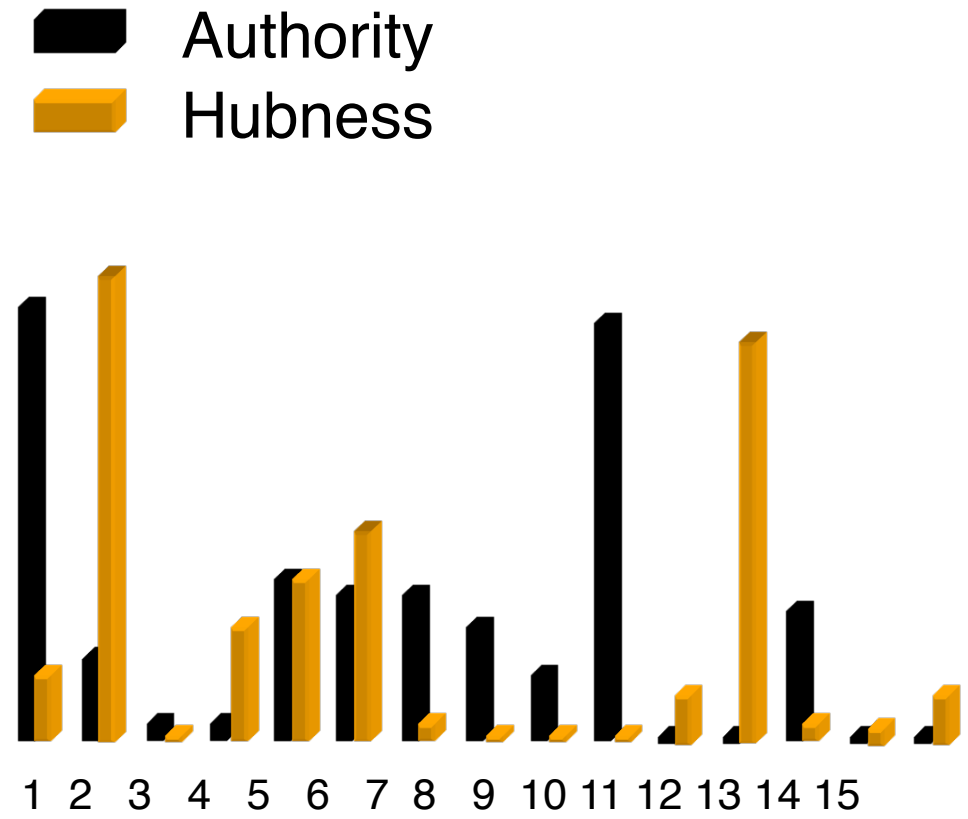
$$a(v) \leftarrow \sum_{w \in pa[v]} h(w)$$

$$h(v) \leftarrow \sum_{w \in ch[v]} a(w)$$

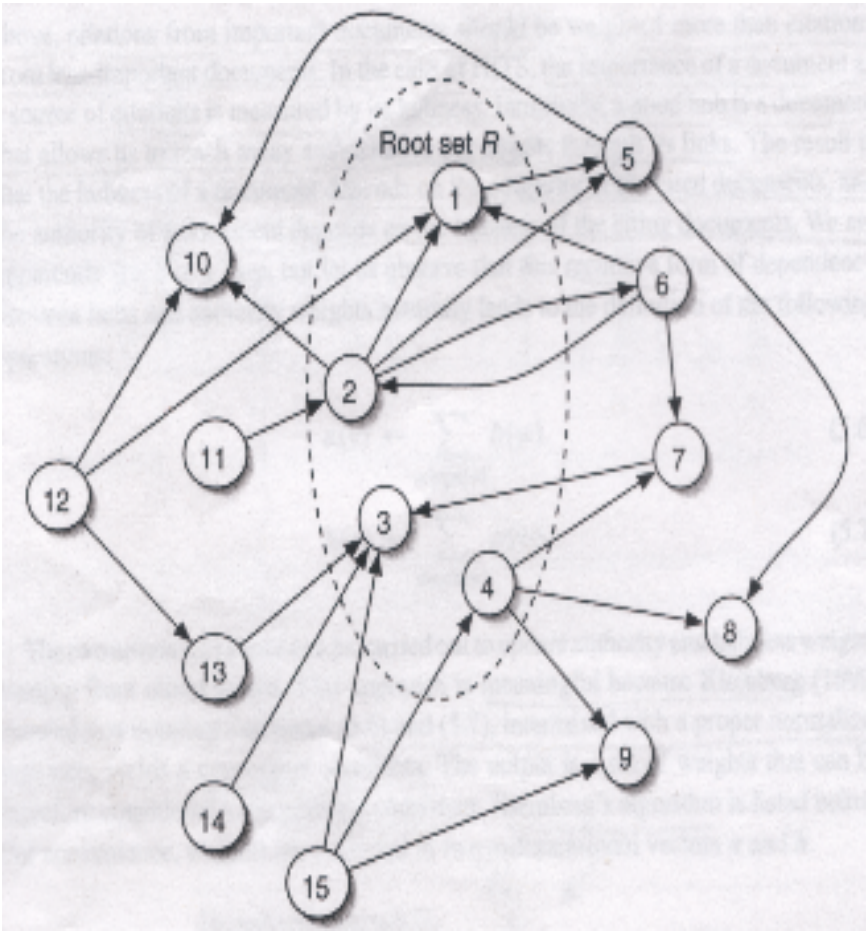
- Using Linear Algebra, we can prove:

  a(v) and h(v) converge

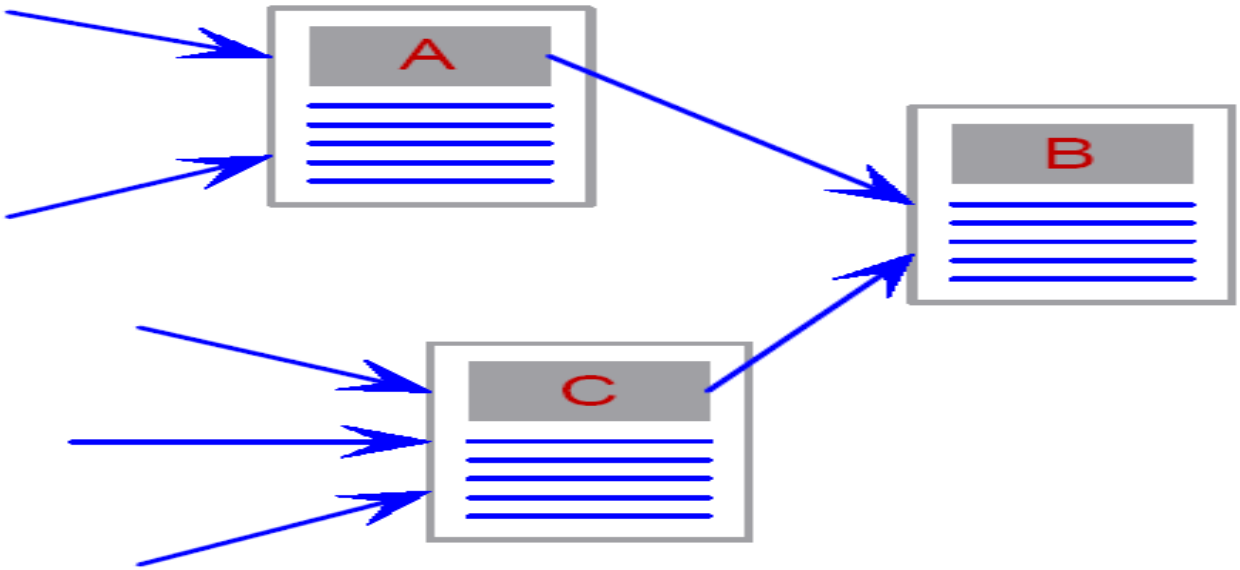
HITS Example Results



Authority and hubness weights



PageRank - Motivation



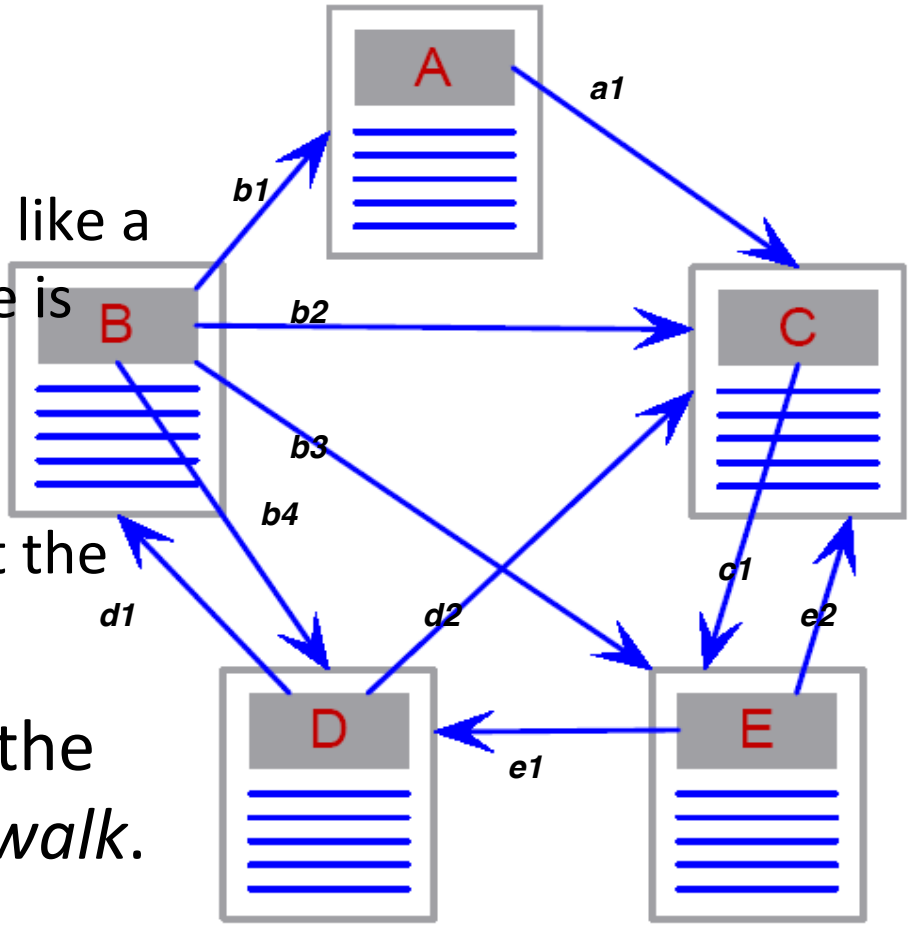
- A link from page *A* to page *B* is a **vote** of the author of *A* for *B,* or a **recommendation** of the page.
- The number incoming links to a page is a measure of importance and authority of the page.
- Also take into account the quality of recommendation, so a page is more important if the sources of its incoming links are important.

What is a Markov Chain?

A Markov chain has two components:

- A network structure much like a web site, where each node is called a state.
- A transition probability of traversing a link given that the chain is in a state.

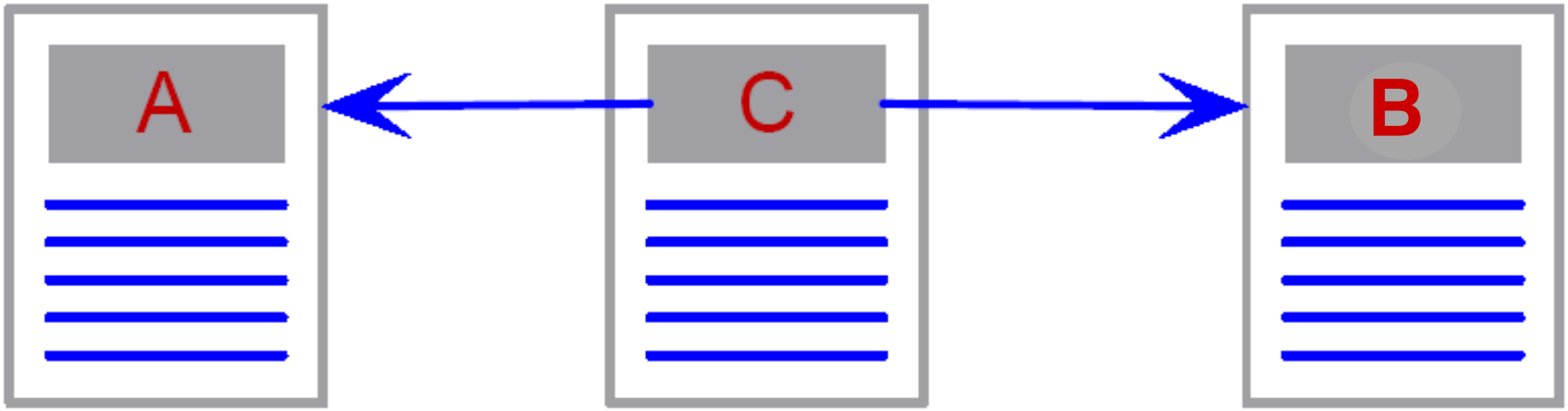
A sequence of steps through the chain is called a *random walk*.



The Random Surfer

- Assume the web is a Markov chain.
- Surfers randomly click on links, where the probability of an outlink from page A is *1/m*, where *m* is the number of outlinks from *A.*
- The surfer occasionally gets *bored* and is *teleported* to another web page, say *B,* where *B* is equally likely to be any page.
- Using the theory of Markov chains it can be shown that if the surfer follows links for long enough, *the PageRank of a web page is the probability that the surfer will visit that page*.

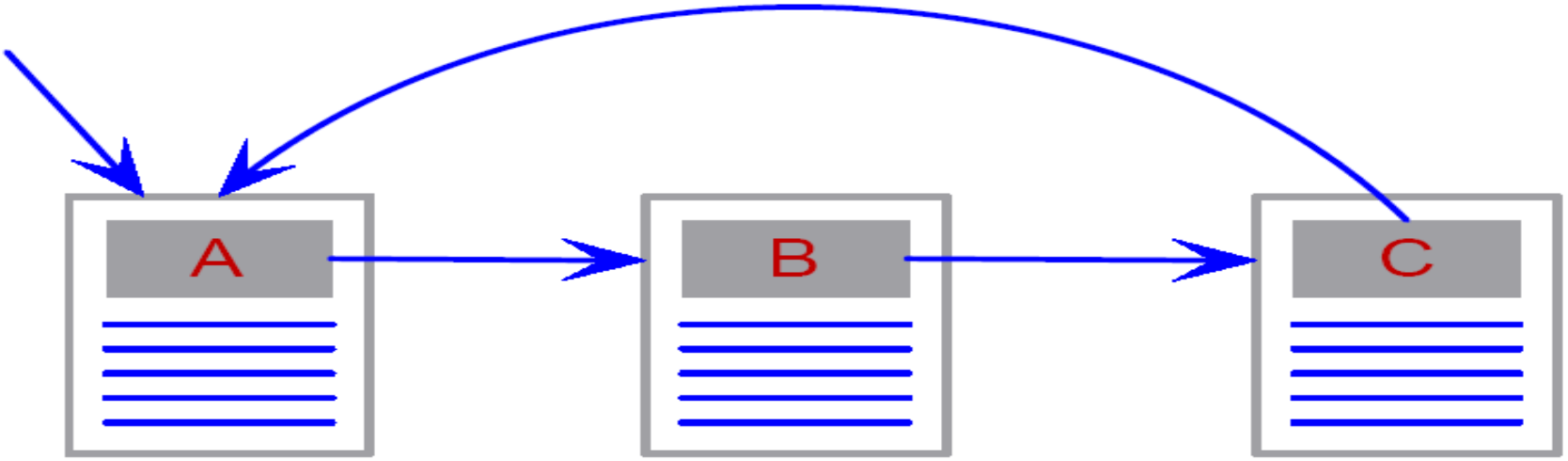
Dangling Pages



- Problem: *A* and *B* have no outlinks.

Solution: Assume *A* and *B* have links to all web pages with equal probability.

Rank Sink



- Problem: Pages in a loop accumulate rank but do not distribute it.
- Solution: Teleportation, i.e. with a certain probability the surfer can jump to any other web page to get out of the loop.

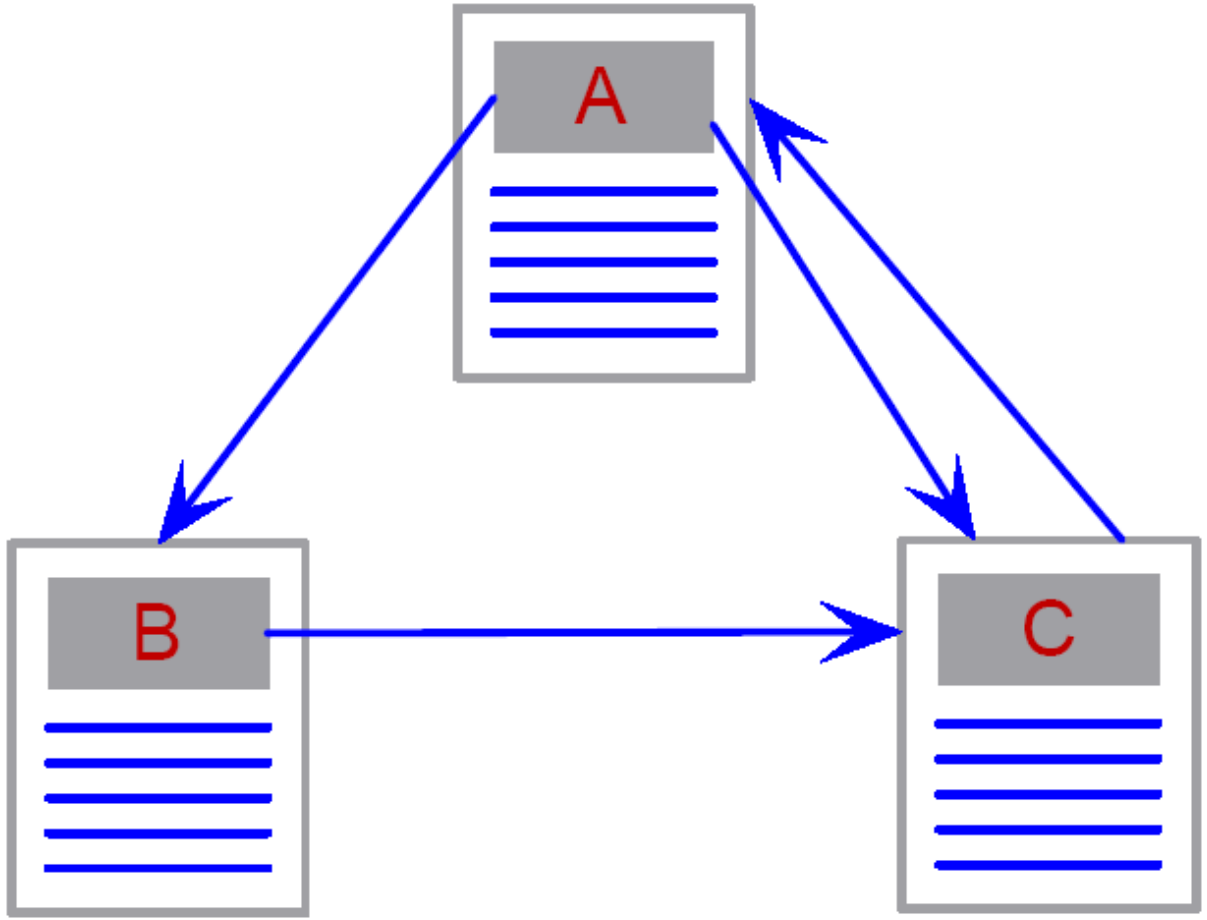
PageRank (*PR*) – Definition

$$PR(P) = \frac{d}{N} + (1-d)\left(\frac{PR(P_1)}{O(P_1)} + \frac{PR(P_2)}{O(P_2)} + \ldots + \frac{PR(P_n)}{O(P_n)}\right)$$

- *P* is a web page
- *Pi* are the web pages that have a link *to P*
- *O*(*Pi*) is the number of outlinks from *Pi*
- *d* is the teleportation probability
- *N* is the size of the web

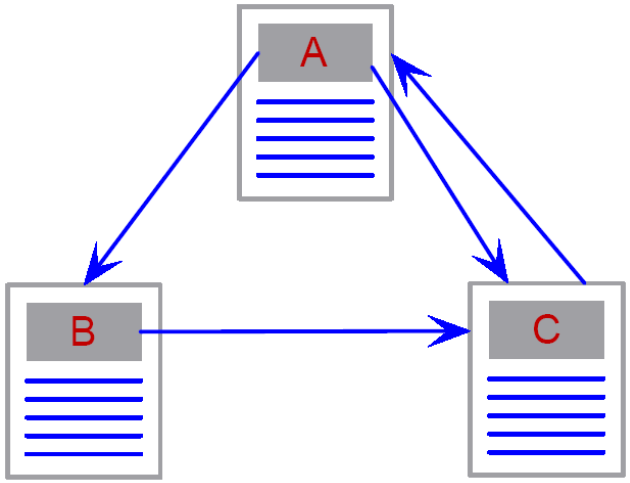
- Difference to HITS
  - HITS takes Hubness & Authority weights
  - The page rank is proportional to its parents' rank, but inversely proportional to its parents' outdegree

Example Web Graph



Iteratively Computing PageRank

- d is normally set to 0.15
- Set initial *PR* values to 1/3
- *Solve the following equations iteratively:*



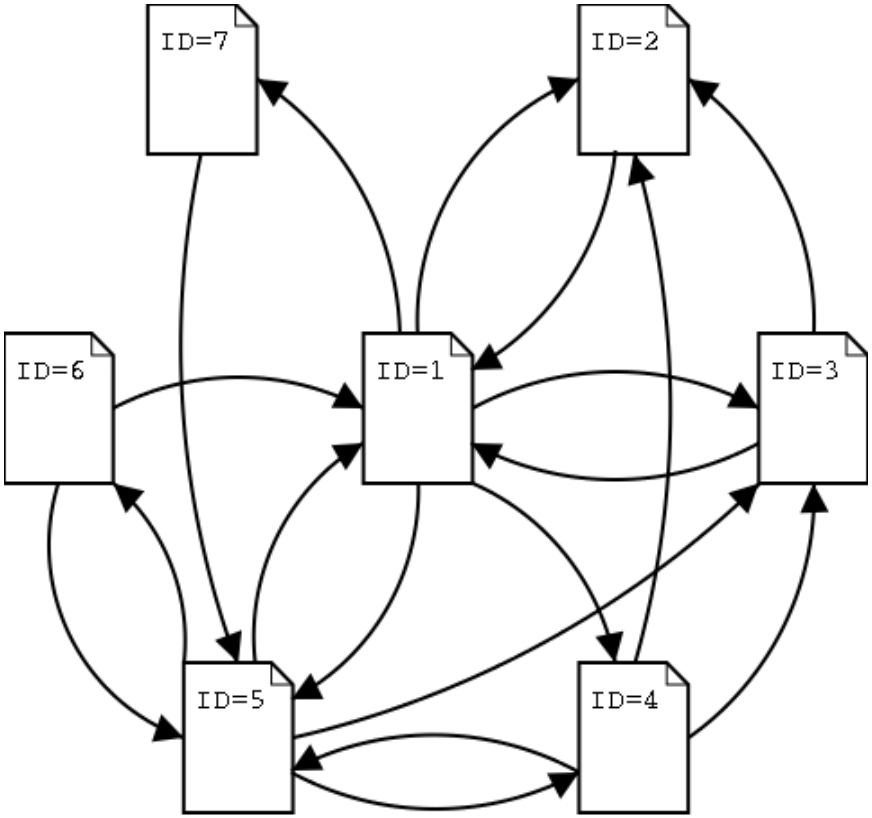
$$PR(A) = 0.15/3 + 0.85PR(C)$$
$$PR(B) = 0.15/3 + 0.85(PR(A)/2)$$
$$PR(C) = 0.15/3 + 0.85(PR(A)/2 + PR(B))$$

Example Computation of PR

	PR(A)	PR(B)	PR(C)	ERROR
1	0,333333333	0,333333333	0,333333333	
2	0,333333333	0,191666667	0,475	0,04014
3	0,45375	0,191666667	0,354583333	0,029
4	0,351395833	0,24284375	0,405760417	0,01571
5	0,394896354	0,199343229	0,405760417	0,00378
6	0,394896354	0,217830951	0,387272695	0,00068
7	0,379181791	0,217830951	0,402987258	0,00049
8	0,39253917	0,211152261	0,396308569	0,00027
9	0,386862284	0,216829147	0,396308569	6,4E-05
10	0,386862284	0,214416471	0,398721246	1,2E-05
11	0,388913059	0,214416471	0,396670471	8,4E-06
12	0,3871699	0,21528805	0,39754205	4,6E-06
13	0,387910742	0,214547208	0,39754205	1,1E-06
14	0,387910742	0,214862066	0,397227192	2E-07
15	0,387643113	0,214862066	0,397494821	1,4E-07
16	0,387870598	0,214748323	0,397381079	7,8E-08
17	0,387773917	0,214845004	**0,397381079**	1,9E-08

- Error converges fast, ~10 repetitions
- Page C is the top ranked one

Matrix Notation



Page ID	OutLinks
1	2,3,4,5,7
2	1
3	1,2
4	2.3.5
5	1,3,4,6
6	1.5
7	5

Adjacency Matrix

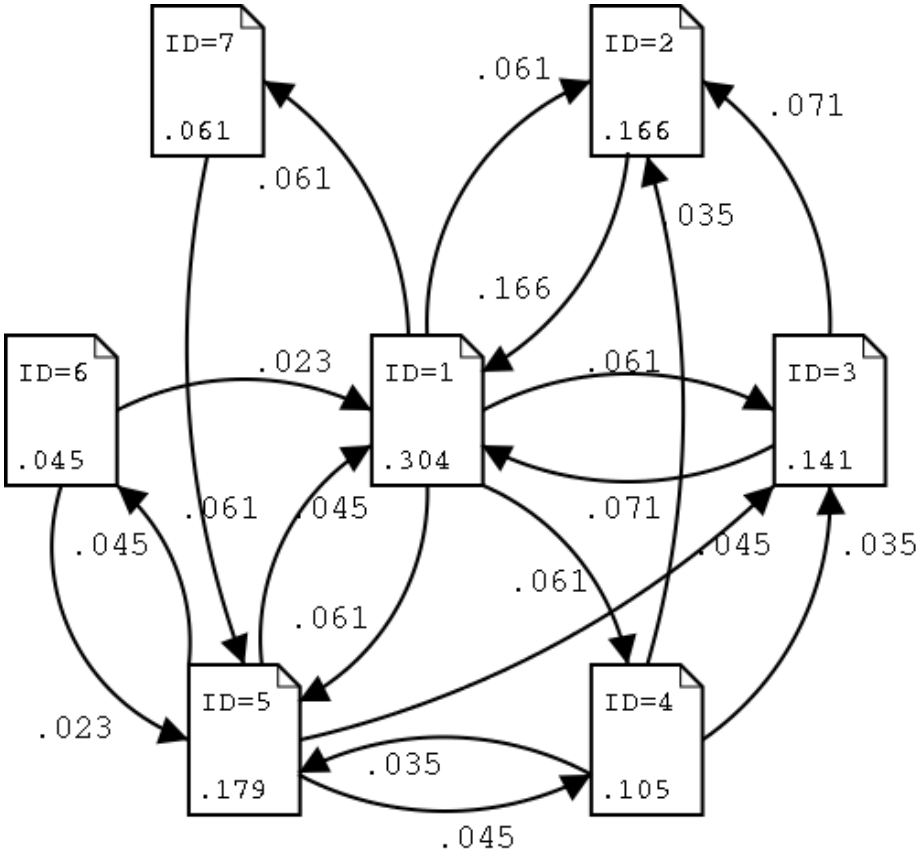
$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

* http://www.kusatro.kyoto-u.com

Matrix Notation

- r be the pagerank vector
- Initialization: $r[i] = 1/n$
- Power method:
  - $r^{t+1} = A\ r^{t+1}$
- Computation converges to the primary eigenvector of the adjacency matrix A
- r normalized => pagerank of page p = probability to visit p in infinite random walks
- To ensure convergence set $A[i,j] = 1/n$ if i and j are not connected

Matrix Notation



PR	ID	OutLink	InLink
0.304	1	2,3,4,5,7	2,3,5,6
0.179	5	1,3,4,6	1,4,6,7
0.166	2	1	1,3,4
0.141	3	1,2	1,4,5
0.105	4	2,3,5	1,5
0.061	7	5	1
0.045	6	1,5	5

- Confirm the result

  # of inlinks from high ranked page

  hard to explain about 5&2, 6&7

- Interesting Topic

  How do you create your homepage highly ranked?

PageRank Algorithm

PAGERANK(M, n, ϵ)

1 $\mathbf{1} \leftarrow [1, \ldots, 1] \in \mathbb{R}^n$
2 $z \leftarrow \frac{1}{n}\mathbf{1}$
3 $x_0 \leftarrow z$
4 $t \leftarrow 0$
5 **repeat**
6 $t \leftarrow t + 1$
7 $x_t \leftarrow M^{\mathrm{T}} x_{t-1}$
8 $d_t \leftarrow \|x_{t-1}\|_1 - \|x_t\|_1$
9 $x_t \leftarrow x_1 + d_t z$
10 $\delta \leftarrow \|x_{t-1} - x_t\|_1$
11 **until** $\delta < \epsilon$
12 **return** x_t

* Page et al, 1998

The Largest Matrix Computation in the World

- Computing PageRank can be done via matrix multiplication, where the matrix has several billion rows and columns.
- The matrix is sparse as average number of outlinks is between 7 and 8.
- Setting d = 0.15 or below requires at most 100 iterations to convergence.
- Researchers still trying to speed-up the computation.

Ranking function web search

- Web search engines take into account 100's of features to rank documents assuming a query
- Two important features are
  - The *PageRank* value of the page containing the *query* terms
  - The *relevance* of the term to the specific page
- Given a term *t* the score of a document *d* is computed as:

$$score_t(di) = w_1(\text{relevance}(t,d_i)) + w_2 pr(d_i)$$

- Where relevance: tf-idf, BM25 etc.
- In a specific case we used:

$$score_t(d) = (\text{tf/idf}(t,d) \cdot \text{title}(t,d))^{1.5} \cdot pr(d)$$

References

Pagerank

- Amy Nicole Langville, Carl Dean Meyer: Survey: Deeper Inside PageRank. Internet Mathematics 1(3): 335-380 (2003)
- "PageRank Computation and the Structure of the Web: Experiments and Algorithms", Arvind Arasu, Jasmine Novak, Andrew Tomkins & John Tomlin
- Klaus Berberich, Michalis Vazirgiannis, Gerhard Weikum, "Time-Aware Authority Ranking", Internet Mathematics 2(3): 301-332 (2005)

Communities in Real Networks

- Real networks are not **random graphs** (e.g., the Erdos-Renyi random graph model)
- Present fascinating patterns and properties:
  - The **degree distribution** is skewed, following a power-law
  - The **average distance** between the nodes of the network is short (the small-world phenomenon)
  - The edges between the nodes may not represent reciprocal relations, forming **directed networks with non-symmetric links**
  - **Edge density** is inhomogeneous (groups of nodes with high concentration of edges within them and low concentration between different groups

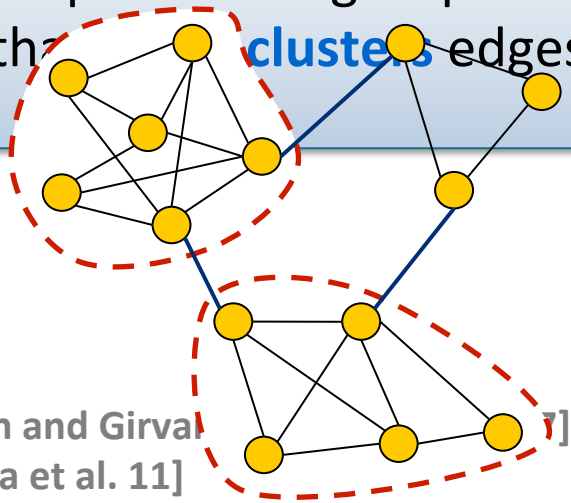
Outline

1. Introduction & Motivation
2. **Community evaluation measures**
3. Graph clustering
4. Graph classification

Basics

- The notion of **community structure** captures the tendency of nodes to be organized into modules (communities, clusters, groups)
  - Members within a community are **more similar** among each other
- Typically, the communities in graphs (networks) correspond to **densely connected** entities (nodes)

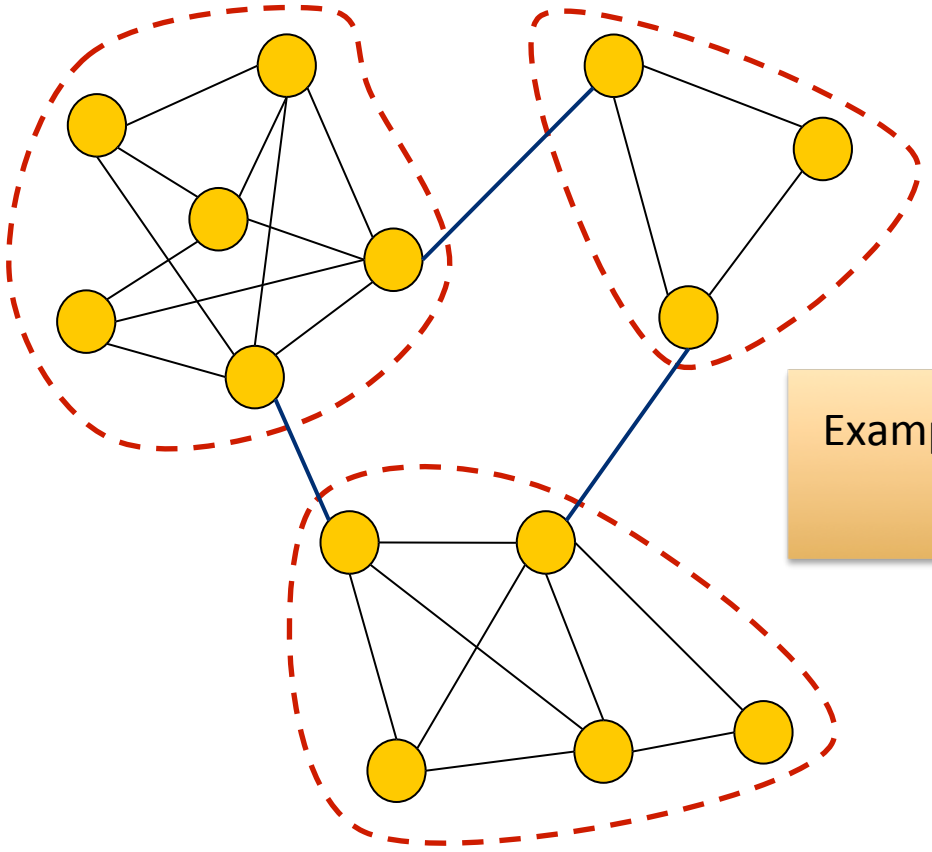
A community corresponds to a group of nodes with more **intra-cluster** edges than **inter-cluster** edges



Example graph with three communities

[Newman '03], [Newman and Girvan], [Fortunato '10], [Danon et al. '05], [Coscia et al. 11]

Schematic representation of communities



Example graph with three communities

Community detection in graphs

- How can we extract the inherent communities of graphs?
- Typically, a two-step approach
  1. Specify a **quality measure** (evaluation measure, objective function) that quantifies the desired properties of communities
  2. Apply **algorithmic techniques** to assign the nodes of graph into communities, optimizing the objective function
- Several measures for quantifying the quality of communities have been proposed
- They mostly consider that communities are set of nodes with many edges between them and few connections with nodes of different communities
  - Many possible ways to formalize it

Community evaluation measures

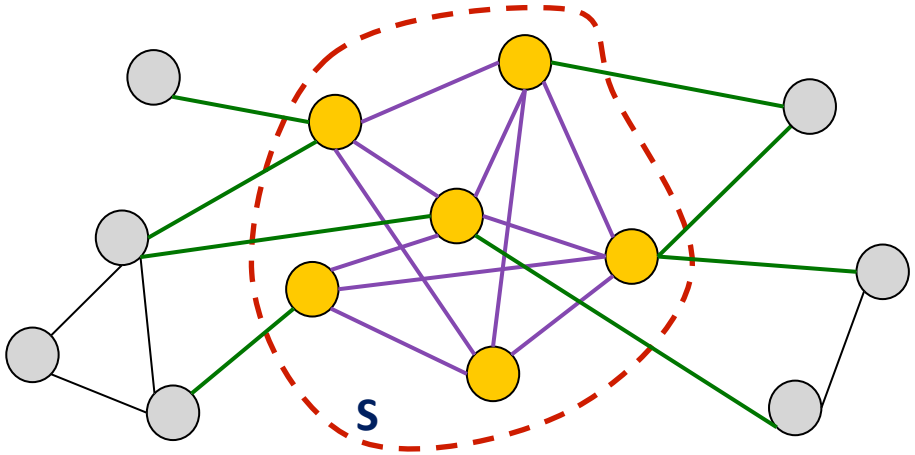
- Focus on
  - Intra-cluster edge density (# of edges within community),
  - Inter-cluster edge density (# of edges across communities)
  - Both two criteria

- We group the community evaluation measures according to
  - Evaluation based on **internal** connectivity
  - Evaluation based on **external** connectivity
  - Evaluation based on **internal and external** connectivity
  - Evaluation based on **network model**

[Leskovec et al. '10], [Yang and Leskovec '12], [Fortunato '10]

Notation

- **G = (V, E)** is an undirected graph, **|V| = n**, **|E| = m**
- **S** is the set of nodes in the cluster
- n_S = **|S|** is the number of nodes in **S**
- m_S is the number of edges in **S**, $m_s = \left|\{(u,v): u \in S, v \in S\}\right|$
- c_S is the number of edges on the boundary of **S**, $c_s = \left|\{(u,v): u \in S, v \notin S\}\right|$
- d_u is the degree of node ***u***
- ***f* (S)** represent the clustering quality of set ***S***



Nodes in **S** (n_S)

Edges in **S** (m_S)

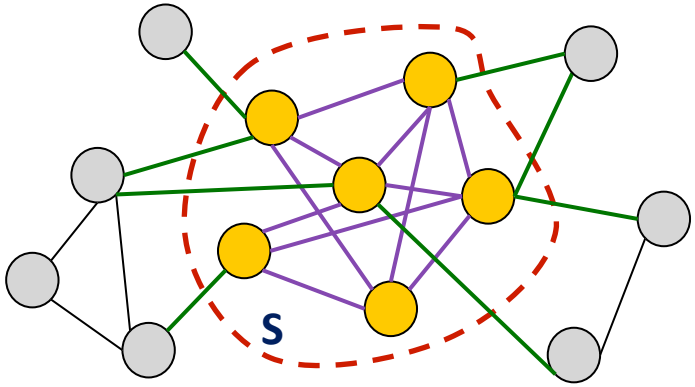
Edges in boundary of **S** (c_S)

Evaluation based on internal connectivity (1)

- **Internal density** [Radicchi et al. '04]

$$f(S) = \frac{m_s}{n_s(n_s - 1)/2}$$

Captures the internal edge density of community **S**



- **Edges inside** [Radicchi et al. '04]

$$f(S) = m_s$$

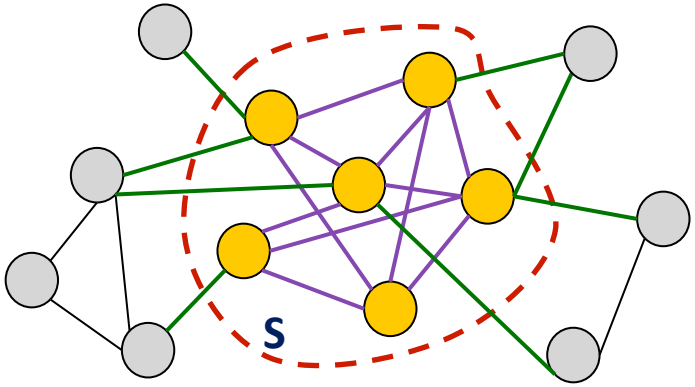
Number of edges between the nodes of **S**

Evaluation based on external connectivity

- **Expansion** [Radicchi et al. '04]

$$f(S) = \frac{c_s}{n_s}$$

Measures the number of edges per node that point outside **S**



- **Cut ratio** [Fortunato '10]

$$f(S) = \frac{c_s}{n_s(n - n_s)}$$

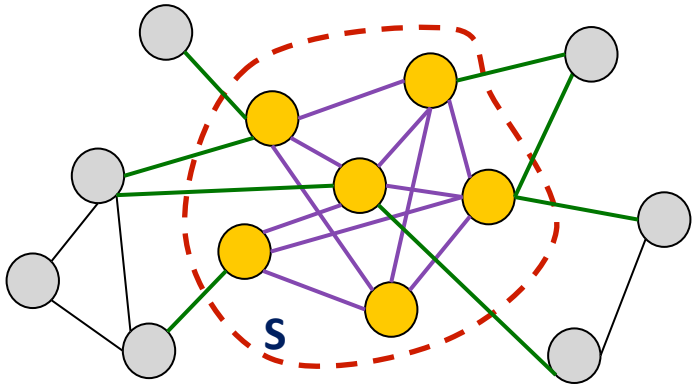
Fraction of existing edges – out of all possible edges – that leaving **S**

Evaluation based on internal and external connectivity (1)

Conductance [Chung '97]

$$f(S) = \frac{c_s}{2m_s + c_s}$$

Measures the fraction of total edge volume that points outside **S**



Normalized cut [Shi and Malic '00]

$$f(S) = \frac{c_s}{2m_s + c_s} + \frac{c_s}{2(m - m_s) + c_s}$$

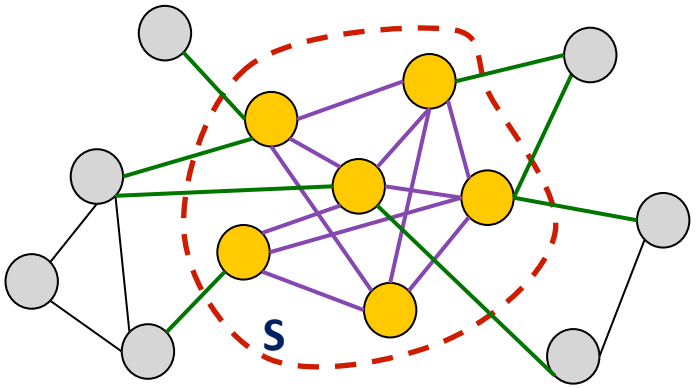
Measures the fraction of total edge volume that points outside **S** normalized by the size of **S**

Evaluation based on internal connectivity (3)

- **Triangle participation ratio (TPR)** [Yang and Leskovec '12]

$$f(S) = \frac{\left|\left\{u : u \in S, \left\{(v,w) : v,w \in S, (u,v) \in E, (u,w) \in E, (v,w) \in E\right\} \neq \emptyset\right\}\right|}{n_s}$$

Fraction of nodes in **S** that belong to a triangle



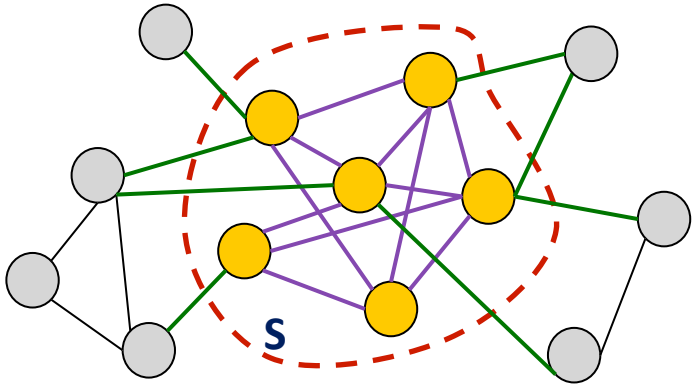
Evaluation based on network model

Modularity [Newman and Girvan '04], [Newman '06]

$$f(S) = \frac{1}{4}\left(m_s - E\left(m_s\right)\right)$$

Measures the difference between the number of edges in **S** and the expected number of edges **E(m_s)** in case of a configuration model

- Typically, a random graph model with the same degree sequence



Outline

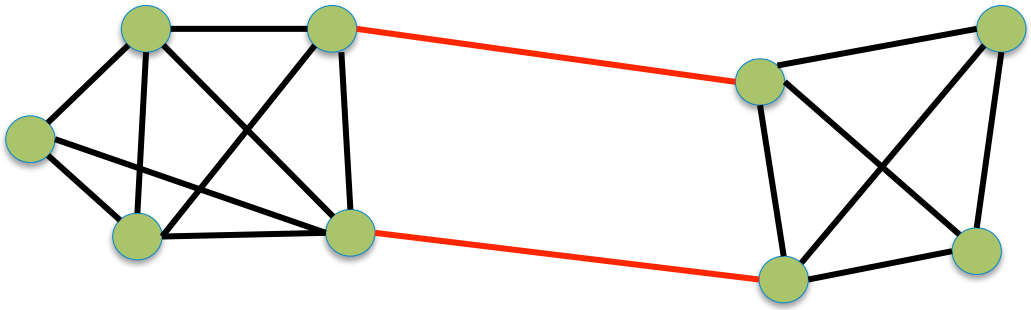
1. Introduction & Motivation
2. Community evaluation measures
3. **Graph clustering**
4. Graph classification

Notations

- Given Graph G=(V,E) undirected:
  - Vertex Set V={v_1,…..v_n}, Edge e_{ij} between v_i and v_j
    - we assume weight $w_{ij}>0$ for e_{ij}
  - |V| : number of vertices
  - d_i degree of v_i : $d_i = \sum_{v_j \in V} w_{ij}$
  - $v(V) = \sum_{v_i \in V} d_i$
  - for $A \subset V$ $\bar{A} = V - A$
  - Given $A, B \subset V$ & $A \cap B = \emptyset, w(A,B) = \sum_{v_i \in A, vj \in B} w_{ij}$
  - D : Diagonal matrix where $D(i,i) = d_i$
  - W : Adjacency matrix $W(i,j) = w_{ij}$

Graph-Cut

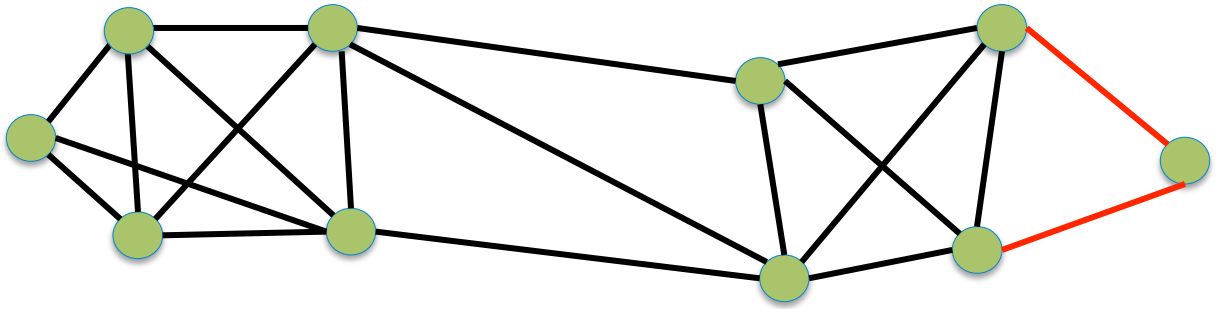
- For k clusters:
  - $cut(A_1, .., Ak) = 1/2 \sum_{i=1}^{k} w(Ai, \overline{A_i})$
    - undirected graph:1/2 we count twice each edge



- Min-cut:Minimize the edges' weight a cluster shares with the rest of the graph

Min-Cut

- Easy for k=2 : Mincut(A_1,A_2)
  - Stoer and Wagner: "A Simple Min-Cut Algorithm"
- In practice one vertex is separated from the rest
  - The algorithm is drawn to outliers



Normalized Graph Cuts

- We can normalize by the size of the cluster (size of sub-graph) :
  - number of Vertices (Hagen and Kahng, 1992):

    $$Ratiocut(A_1, \dots Ak) = \sum_{i=1}^{k} \frac{cut(Ai, \overline{A_i})}{|Ai|}$$

  - sum of weights (Shi and Malik, 2000) :

    $$Ncut(A_1, \dots Ak) = \sum_{i=1}^{k} \frac{cut(Ai, \overline{A_i})}{v(A_i)}$$

- Optimizing these functions is NP-hard
- Spectral Clustering provides solution to a relaxed version of the above

From Graph Cuts to Spectral Clustering

- For simplicity assume k=2:
  - Define $f: V \to \mathbb{R}$ for Graph G :

$$f_i = \begin{cases} 1 & v_i \in A \\ -1 & v_i \in \overline{A} \end{cases}$$

- Optimizing the original cut is equivalent to an optimization of:

$$\sum_{i,j=1}^{n} w_{ij}\left(f_i - f_j\right)^2$$

$$= \sum_{v_i \in A, vj \in \overline{A}} w_{ij}(1+1)^2 + \sum_{v_i \in \overline{A}, vj \in A} w_{ij}(-1-1)^2$$

$$= \mathbf{8} * \boldsymbol{cut}\left(\boldsymbol{A}, \overline{\boldsymbol{A}}\right)$$

Graph Laplacian

- How is the previous useful in Spectral clustering?

$$\sum_{i,j=1}^{n} w_{ij}(f_i - fj)^2$$

$$= \sum_{i,j=1}^{n} w_{ij} {f_i}^2 - 2\sum_{i,j=1}^{n} w_{ij} f_i f_j + \sum_{i,j=1}^{n} w_{ij} {f_j}^2$$

$$= \sum_{i,j=1}^{n} d_i {f_i}^2 - 2\sum_{i,j=1}^{n} w_{ij} f_i f_j + \sum_{i,j=1}^{n} d_j {f_j}^2$$

$$= 2\left(\sum_{i,j=1}^{n} d_{ii} {f_i}^2 - \sum_{i,j=1}^{n} w_{ij} f_i f_j\right)$$

$$= 2\left(\boldsymbol{f^T D f - f^T W f}\right) = 2\boldsymbol{f^T (D - W) f} = 2\boldsymbol{f^T L f}$$

- **f:**a single vector with the cluster assignments of the vertices
- **L=D-W :** the Laplacian of a graph

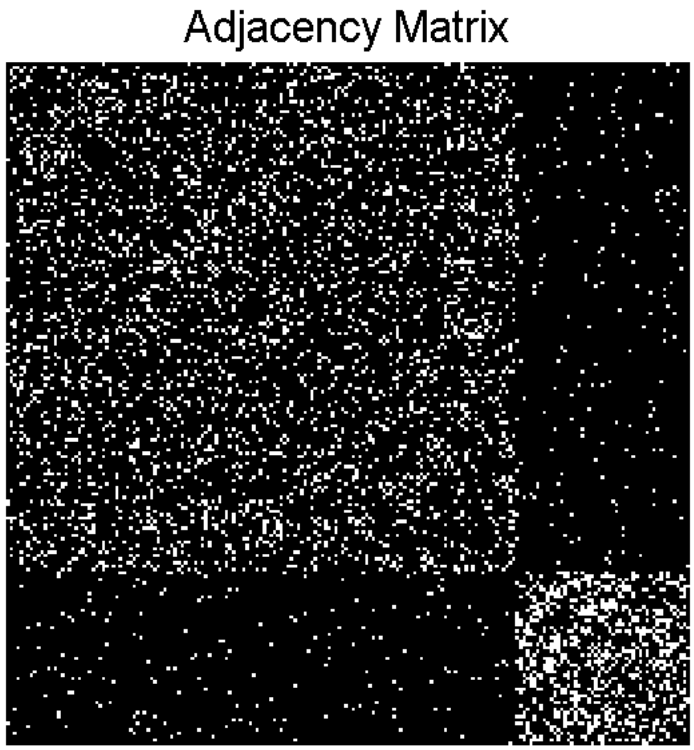
Properties of L

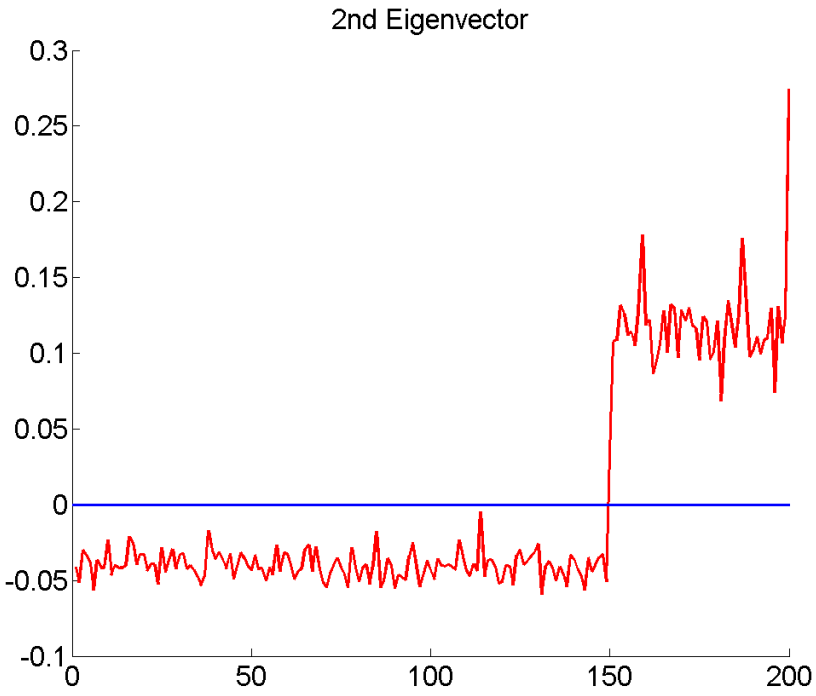
- L is
  - Symmetric
  - Positive
  - Semi-definite
- The smallest eigenvalue of L is 0
  - The corresponding eigenvector is $\mathbb{1}$
- L has n non-negative, real valued eigenvalues
  - $0 = \lambda 1 \leq \lambda 2 \leq \ldots \leq \lambda n$

Two Way Cut from the Laplacian

- We could solve $min_f f^T L f$ where $f \in \{-1,1\}^n$
- NP-Hard for discrete cluster assignments
  - Relax the constraint $to\ f \in R^n$:
    $min_f f^T L f$ subject to $f^T f = n$
- The solution to this problem is given by:
  - (**Rayleigh-Ritz Theorem**) the eigenvector corresponding to smallest eigenvalue: 0 and the corresponding eigenvector (full of 1s) offers no information
- We use the second eigenvector as an approximation
  - $f_i > 0$ the vertex belongs to one cluster , $f_i < 0$ to the other

Example


Adjacency Matrix


2nd Eigenvector
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
0
50
100
150
200

Ratio Cut

- $Ratiocut(A_1, \dots Ak) = \sum_{i=1}^{k} \frac{cut(Ai, \overline{A_i})}{|Ai|}$
  - Define $f: V \rightarrow \mathbb{R}$ for Graph G :

$$f_i = \begin{cases} \sqrt{\frac{|\overline{A}|}{|A|}} & vi \in A \\ -\sqrt{\frac{|A|}{|\overline{A}|}} & v_i \in \overline{A} \end{cases}$$

  - $\sum_{i,j=1}^{n} w_{ij}(f_i - fj)^2 = 2cut(\mathrm{A}, \overline{A}) \left(\sqrt{\frac{|\overline{A}|}{|A|}} + \sqrt{\frac{|A|}{|\overline{A}|}} + 2 \right)$

    $= 2|V|Ratiocut(A, \overline{A})$

Ratio Cut

- We have $min_f f^T L f$ subject to

$$f^T 1 = 0, \; fTf = n$$

$$f^T 1 = \sum_i^n f_i = \sum_{v_i \in A} \sqrt{\frac{|\bar{A}|}{|A|}} + \sum_{v_i \in \bar{A}} - \sqrt{\frac{|A|}{|\bar{A}|}} = |A| \sqrt{\frac{|\bar{A}|}{|A|}} - |\bar{A}| \sqrt{\frac{|A|}{|\bar{A}|}} = 0$$

$$f^T f = \sum_i^n f_i^2 = |\bar{A}| + |A| = n$$

- The second smallest eigenvalue of $Lf = \lambda f$ approximates the solution

Normalized Cut

- $Ncut(A_1, \dots Ak) = \sum_{i=1}^{k} \frac{cut(Ai, \overline{A_i})}{v(A_i)}$
- Define $f: V \rightarrow \mathbb{R}$ for Graph G :

$$f_i = \begin{cases} \sqrt{\frac{v(\overline{A})}{v(A)}} & vi \in A \\ -\sqrt{\frac{v(A)}{v(\overline{A})}} & vi \in \overline{A} \end{cases}$$

- $\sum_{i,j=1}^{n} w_{ij}(f_i - fj)^2 = 2cut(\mathrm{A}, \overline{A}) \left(\sqrt{\frac{v(\overline{A})}{v(A)}} + \sqrt{\frac{v(A)}{v(\overline{A})}} + 2 \right)$

$$= 2v(\mathrm{V})\mathrm{Ncut}(\mathrm{A}, \overline{A})$$

Normalized Cut

- Similarly: $\min_f f^T L f$ subject to

$$f^T D 1 = 0, f^T D f = v(V)$$

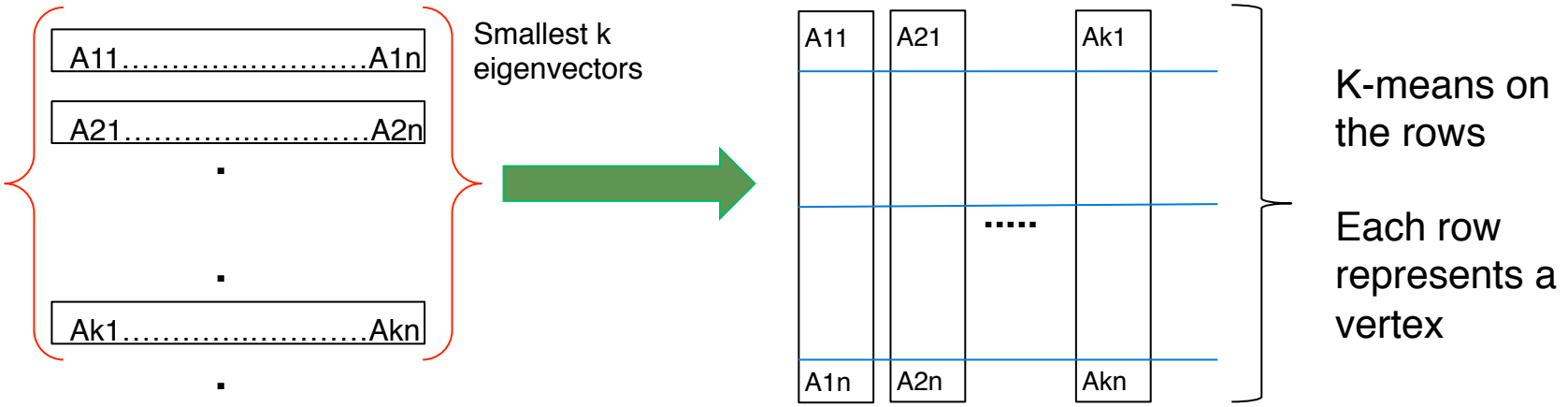
- Assume $h = D^{1/2} f$
  - $min_h h^T D^{-1/2} L D^{-1/2} h$ subject to

$$h^T D^{1/2} 1 = 0, \qquad h^T h = v(V)$$

  - The answer is in the eigenvector of the second smallest eigenvalue of $L_{sym} = D^{-1/2} L D^{-1/2}$
    Shi and Malik (2000)
- L_{sym} is the normalized Laplacian
  - has n non-negative, real valued eigenvalues
  - $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

Multi-Way Graph Partition

- The cluster assignment is given by the smallest k eigenvectors of L
- The real values need to be converted to cluster assignments
  - We use k-means to cluster the rows
  - We can substitute *L* with L_{sym}



References – Graph clustering

- UIrike von Luxburg, A Tutorial on Spectral Clustering, Statistics and Computing, 2007
- Davis, C., W. M. Kahan (March 1970). The rotation of eigenvectors by a perturbation. III. SIAM J. Numerical Analysis 7
- Shi, Jianbo, and Jitendra Malik. "Normalized cuts and image segmentation, "*Pattern Analysis and Machine Intelligence, IEEE Transactions on* (2000).
- Mechthild Stoer and Frank Wagner. 1997. A simple min-cut algorithm. *J. ACM*
- Ng, Jordan & Weiss, K-means algorithm on the embedded eigen-space, NIPS 2001
- Hagen, L. Kahng, , "New spectral methods for ratio cut partitioning and clustering," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , 1992

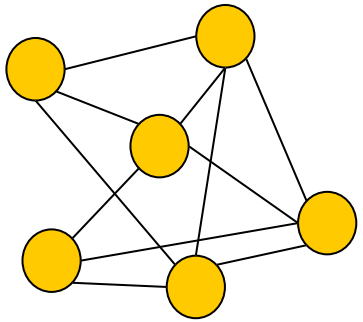
Graph Clustering Algorithms

- Spectral Clustering
- **Modularity Based Methods**

Main idea

- **Modularity** function [Newman and Girvan '04], [Newman '06]
- Initially introduced as a measure for assessing the strength of communities
  - **Q = (fraction of edges within communities) –**

    (expected number of edges within communities)
- What is the **expected** number of edges?
- Consider a configuration model
  - **Random graph** model with the same degree distribution
  - Let P_{ij} = probability of an edge between nodes **i** and **j** with degrees k_i and k_j respectively
  - Then $P_{ij} = k_i k_j / 2m$, where $m = |E| = \frac{1}{2} \sum_i k_i$



Formal definition of modularity

Modularity Q

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(C_i, C_j)$$

where

- **A** is the adjacency matrix
- k_i, k_j the degrees of nodes **i** and **j** respectively
- **m** is the number of edges
- C_i is the community of node **i**
- **δ(.)** is the Kronecker function: 1 if both nodes **i** and **j** belong on the same community ($C_i = C_j$), 0 otherwise

[Newman and Girvan '04], [Newman '06]

Properties of modularity

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(c_i, c_j)$$

- **Larger** modularity **Q** indicates **better** communities (more than random intra-cluster density)
  - The community structure would be better if the number of internal edges exceed the expected number
- Modularity value is always **smaller than 1**
- It can also take **negative values**
  - E.g., if each node is a community itself
  - No partitions with positive modularity → No community structure
  - Partitions with large negative modularity → Existence of subgraphs with small internal number of edges and large number of inter-community edges

[Newman and Girvan '04], [Newman '06], [Fortunato '10]

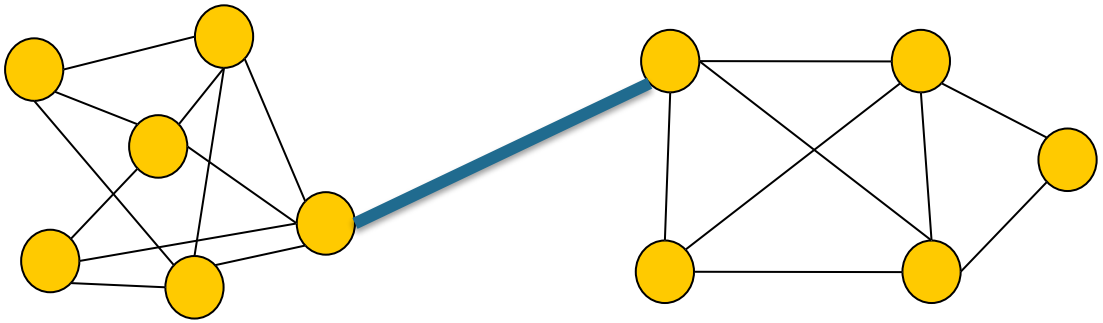
Applications of modularity

- Modularity can be applied:
  - As **quality function** in clustering algorithms
  - As **evaluation measure** for comparison of different partitions or algorithms
  - As a community detection tool itself
    - **Modularity optimization**
  - As criterion for reducing the size of a graph
    - Size reduction preserving modularity [Arenas et al. '07]

[Newman and Girvan '04], [Newman '06], [Fortunato '10]

Modularity-based community detection

- Modularity was first applied as a **stopping criterion** in the Newman-Girvan algorithm
- Newman-Girvan algorithm [Newman and Girvan '04]
  - A **divisive** algorithm (detect and remove edges that connect vertices of different communities)
  - **Idea:** try to identify the edges of the graph that are most between other vertices → responsible for connecting many node pairs
  - Select and remove edges based to the value of **betweenness centrality**
  - **Betweenness centrality:** number of **shortest paths** between every pair of nodes, that pass through an edge



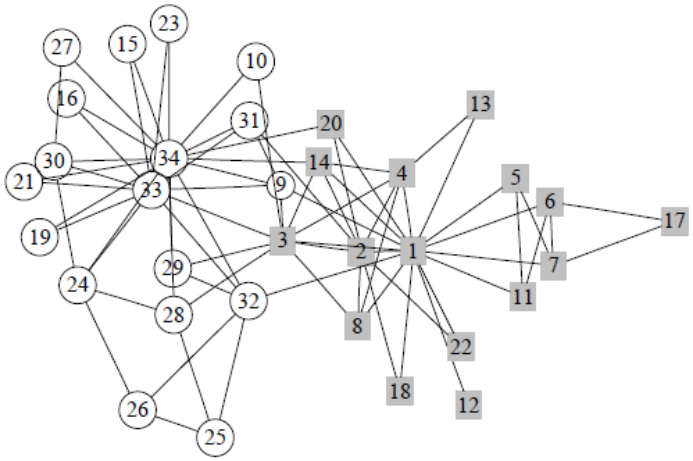
Edge betweenness is higher for edges that connect different communities

Newman-Girvan algorithm (1)

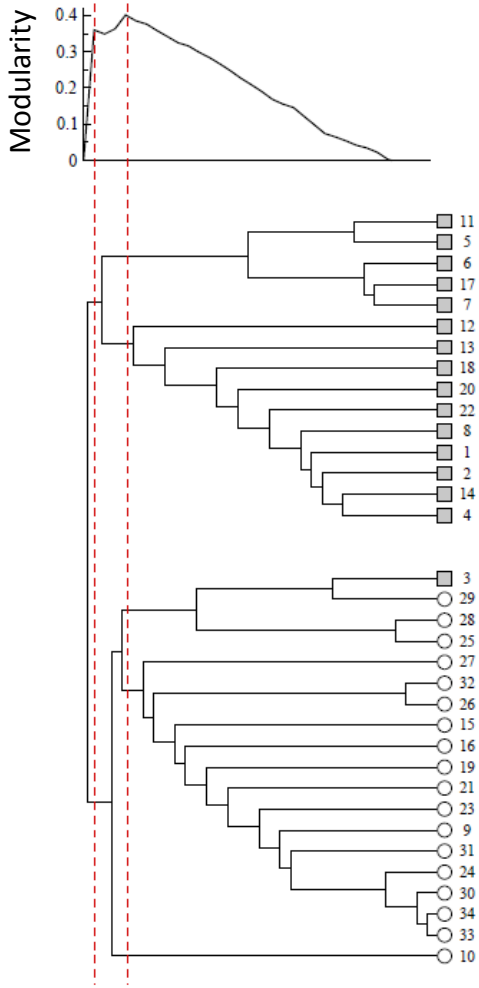
- **Basic steps:**
  1. Compute betweenness centrality for all edges in the graph
  2. Find and remove the edge with the highest score
  3. Recalculate betweenness centrality score for the remaining edges
  4. Go to step 2
- How do we know if the produced communities are **good ones** and stop the algorithm?
  - The output of the algorithm is in the form of a **dendrogram**
  - Use **modularity** as a criterion to cut the dendrogram and terminate the algorithm (Q ~= 0.3-0.7 indicates good partitions)
- Complexity: **$O(m^2n)$** (or **$O(n^3)$** on a sparse graph)

[Newman and Girvan '04], [Girvan and Newman '02]

Newman-Girvan algorithm (2)



Zachary's karate club



Community structure

[Newman and Girvan '04]

Modularity optimization

- High values of modularity indicate good quality of partitions
- **Goal:** find the partition that corresponds to the maximum value of modularity
- **Modularity maximization** problem
  - Computational difficult problem **[Brandes et al. '06]**
  - Appoximation techniques and heuristics
- Four main categories of techniques
  1. Greedy techniques
  2. **Spectral optimization**
  3. Simulated annealing
  4. Extremal optimization

[Fortunato '10]

Spectral optimization (1)

- **Idea:** Spectral techniques for modularity optimization
- **Goal:** Assign the nodes into two communities, **X** and **Y**
- Let $s_i, \forall i \in V$ be an indicator variable where $s_i = +1$ if **i** is assigned to **X** and $s_i = -1$ if **i** is assigned to **Y**

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(C_i, C_j)$$

$$= \frac{1}{4m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)(s_i s_j + 1)$$

$$= \frac{1}{4m}\sum_{ij} B_{ij} s_i s_j = \frac{1}{4m} s^T \mathrm{B} s$$

- **B** is the **modularity matrix**

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$$

[Newman '06], [Newman '06b]

Spectral optimization (2)

- Modularity matrix **B**

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$$

- Vector **s** can be written as a linear combination of the eigenvectors $\mathbf{u_i}$ of the modularity matrix **B**

$$s = \sum_i a_i u_i \qquad \text{where} \qquad a_i = u_i^T s$$

- Modularity can now expressed as

$$Q = \frac{1}{4m} \sum_i a_i u_i^T \; B \; \sum_j a_j u_j^T = \frac{1}{4m} \sum_{i=1}^{n} \left(u_i^T s \right)^2 \beta_i$$

Where $\boldsymbol{\beta_i}$ is the eigenvalue of **B** corresponding to eigenvector $\mathbf{u_i}$

[Newman '06], [Newman '06b]

Spectral optimization (3)

- Spectral modularity optimization algorithm
  1. Consider the eigenvector $\mathbf{u_1}$ of **B** corresponding to the largest eigenvalue
  2. Assign the nodes of the graph in one of the two communities **X** ($\mathbf{s_i = +1}$) and **Y** ($\mathbf{s_i = -1}$) based on the **signs** of the corresponding components of the eigenvector

$$s_i = \begin{cases} 1 & if\ u_1(i) \geq 0 \\ -1 & if\ u_1(i) < 0 \end{cases}$$

  - More than two partitions?
  1. **Iteratively**, divide the produced partitions into two parts
  2. If at any step the split does not contribute to the modularity, leave the corresponding subgraph as is
  3. End when the entire graph has been splinted into no further divisible subgraphs
  - Complexity: **$O(n^2 \log n)$** for sparse graphs

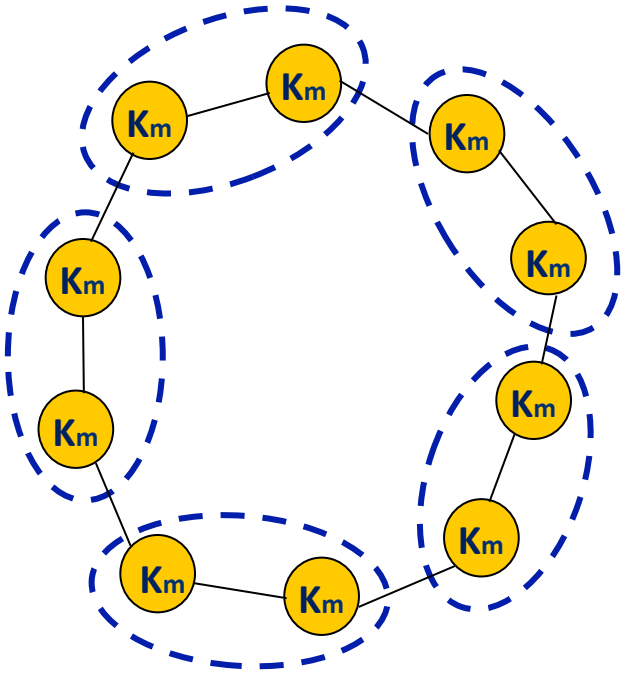
[Newman '06], [Newman '06b]

Extensions of modularity

- Modularity has been extended in several directions
  - Weighted graphs **[Newman '04]**
  - Bipartite graphs **[Guimera et al '07]**
  - Directed graphs (next in this tutorial) **[Arenas et al. '07], [Leicht and Newman '08]**
  - Overlapping community detection (next in this tutorial) **[Nicosia et al. '09]**
  - Modifications in the configuration model – local definition of modularity **[Muff et al. '05]**

Resolution limit of modularity

- **Resolution Limit** of modularity **[Fortunato and Barthelemy '07]**
- The method of modularity optimization may not detect communities with relatively small size, which depends on the total number of edges in the graph



- K_m are cliques with **m** edges (**$m \leq \sqrt{|E|}$**)
- K_m represent well-defined clusters
- However, the maximum modularity corresponds to clusters formed by two or more cliques
- It is difficult to know if the community returned by modularity optimization corresponds to a **single community** or a **union of smaller communities**

References (modularity)

- M.E.J. Newman and M. Girvan. Finding and evaluating community structure in networks. Physical Review E 69(02), 2004.
- M.E.J. Newman. Modularity and community structure in networks. PNAS, 103(23), 2006.
- S.E. Schaeffer. Graph clustering. Computer Science Review 1(1), 2007.
- S. Fortunato. Community detection in graphs. Physics Reports 486 (3-5), 2010.
- M. Coscia, F. Giannotti, and D. Pedreschi. A classification for community discovery methods in complex networks. Statistical Analysis and Data Mining 4 (5), 2011.
- A. Arenas, J. Duch, A. Fernandez, and S. Gomez. Size reduction of complex networks preserving modularity. New J. Phys., 9(176), 2007.
- M. Girvan and M.E.J. Newman. Community structure in social and biological networks. PNAS 99(12), 2002.
- U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner. On Modularity Clustering. IEEE TKDE 20(2), 2008.
- M.E.J. Newman. Fast algorithm for detecting community structure in networks. Phys. Rev. E 69, 2004.
- A. Clauset, M.E.J. Newman, and C. Moore. Finding community structure in very large networks. Phys. Rev. E 70, 2004.

References (modularity)

- M.E.J. Newman. Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E 74, 2006.
- R. Guimera, M. Sales-Pardo, L.A.N. Amaral. Modularity from Fluctuations in Random Graphs and Complex Networks. Phys. Rev. E 70, 2004.
- J. Duch and A. Arenas. Community detection in complex networks using Extremal Optimization. Phys. Rev. E 72, 2005.
- A. Arenas, J. Duch, A. Fernandez, and S. Gomez. Size reduction of complex networks preserving modularity. New Journal of Physics 9(6), 2007.
- E.A. Leicht and M.E.J. Newman. Community structure in directed networks. Phys. Rev. Lett. 100, 2008.
- V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri. Extending the definition of modularity to directed graphs with overlapping communities. J. Stat. Mech. 03, 2009.
- S. Muff, F. Rao, A. Caflisch. Local modularity measure for network clusterizations. Phys. Rev. E, 72, 2005.
- S. Fortunato and M. Barthelemy. Resolution limit in community detection. PNAS 104(1), 2007.

Outline

- Modularity Based Methods
- **Louvain algorithm**

Louvain algorithm

- method to extract the community structure of networks
- heuristic method based on modularity optimization.
- Relatively low cost in terms of computation time.
- quality of the communities good, as measured by modularity.

The algorithm

- Assume a weighted undirected graph $G(E,V)$ and a splitting G in $\{Ci\}$ communities. Then the modularity of the community splitting is:

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta\left(C_i, C_j\right)$$

- A_{ij} is the weight among nodes i and j, k_i is the sum of weight of the edges attached to vertex i, c_i the community to which i belongs and the δ function $\delta(u,v)$ is 1 if u-v and 0 otherwise. Finally $m = \frac{1}{2}\sum_{ij} A_{ij}$

Phase 1:

- weighted graph. Create for each node i a community .
- For each node i consider all its neighbors j. For each of them evaluate the gain in modularity if we place i in the community of j.
- The node i is placed in the community that maximizes the gain
- The process is repeated until there is no further gain for any reassignment.

$$\Delta Q = \left[\frac{\sum_{in} + k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]$$

Phase1: re-assignment of nodes – modularity gain

- Reassigning a node i to a cluster C incurs a gain in modularity:

$$\Delta Q = \left[\frac{\sum_{in} + k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m} \right)^2 \right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right]$$

$\sum_{in}$ sum of the weights of links in cluster C,

$\sum_{tot}$ sum of weights of the links incident to nodes in C ,

κ_i sum weights of links incident node i,

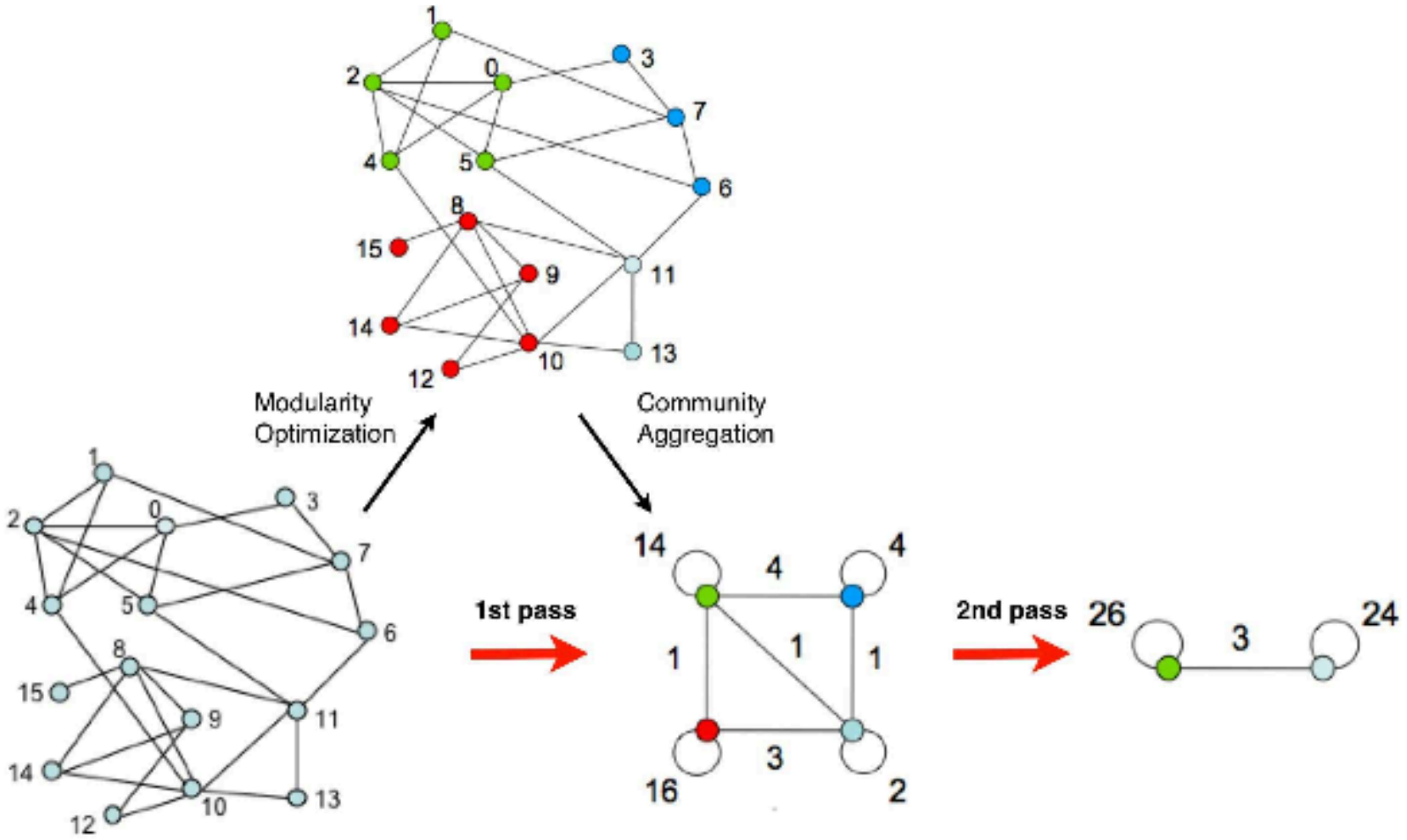
$\kappa_{i,in}$ sum weights of links from i to nodes in C

m sum of the weights of all the links in the network.

Phase 2: collapsing communities

- For each community c_i all nodes i in collapse in to one super-node c_i
- Inter community links weight: sum of the weights of the links among the communities nodes.
- Within community links collapse to a self link
- Go to phase 1

Iterative passes



Example

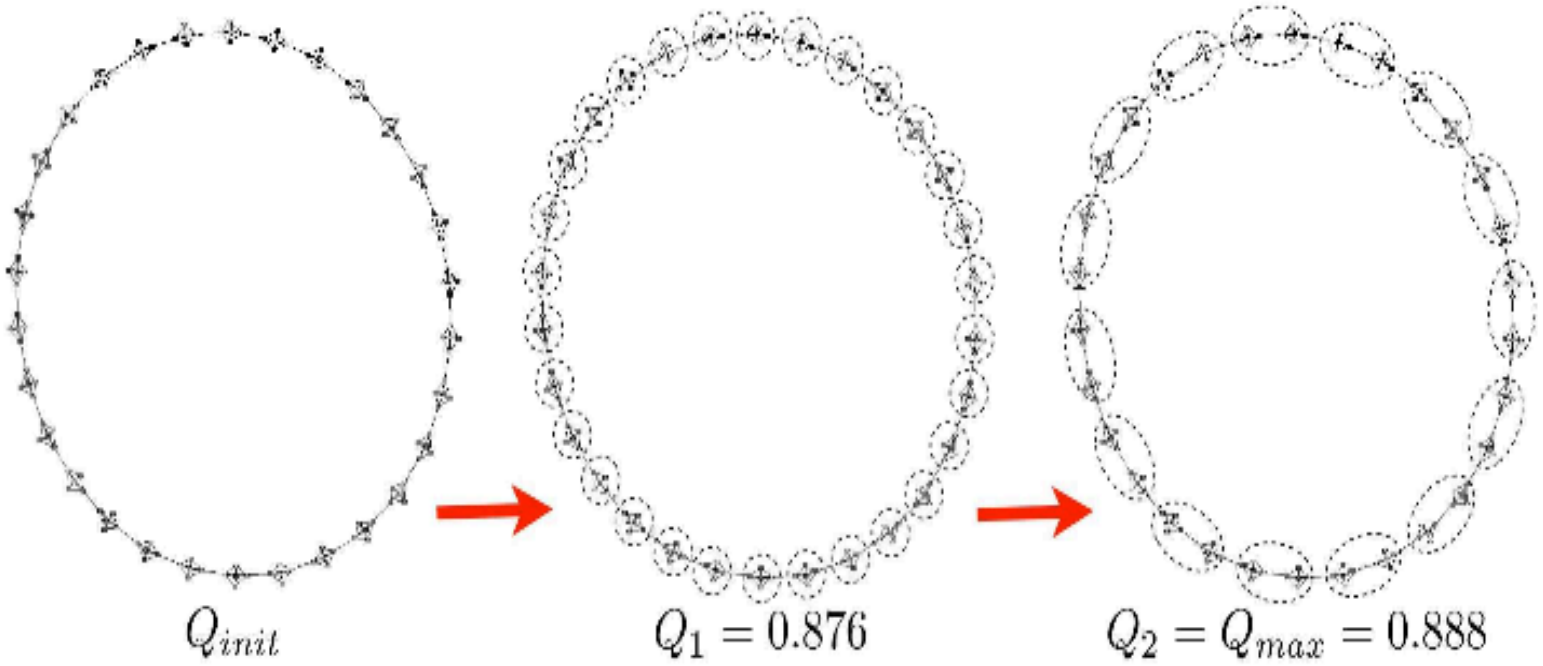


Figure 2. We have applied our method to the ring of 30 cliques discussed in [23]. The cliques are composed of 5 nodes and are inter-connected through single links. The first pass of the algorithm finds the natural partition of the network. The second pass finds the global maximum of modularity where cliques are combined into groups of two.

Performance considerations

Performance on large scale graphs (2008)

- sub-network of the .uk domain (39 M nodes, 783 M links
- Stanford WebBase crawler (118 million nodes and 1Bn links.
- Time: 12 minutes and 152 minutes respectively

Almost linear complexity for sparse data

- Number of communities decreases drastically after just a few passes
- Running time is concentrated on the first iterations.

Resolution limit problem modularity circumvented

- intrinsic multi-level nature

References (modularity)

- M.E.J. Newman and M. Girvan. Finding and evaluating community structure in networks. Physical Review E 69(02), 2004.
- M.E.J. Newman. Modularity and community structure in networks. PNAS, 103(23), 2006.
- S.E. Schaeffer. Graph clustering. Computer Science Review 1(1), 2007.
- S. Fortunato. Community detection in graphs. Physics Reports 486 (3-5), 2010.
- M. Coscia, F. Giannotti, and D. Pedreschi. A classification for community discovery methods in complex networks. Statistical Analysis and Data Mining 4 (5), 2011.
- A. Arenas, J. Duch, A. Fernandez, and S. Gomez. Size reduction of complex networks preserving modularity. New J. Phys., 9(176), 2007.
- M. Girvan and M.E.J. Newman. Community structure in social and biological networks. PNAS 99(12), 2002.
- U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner. On Modularity Clustering. IEEE TKDE 20(2), 2008.
- M.E.J. Newman. Fast algorithm for detecting community structure in networks. Phys. Rev. E 69, 2004.
- A. Clauset, M.E.J. Newman, and C. Moore. Finding community structure in very large networks. Phys. Rev. E 70, 2004.

References (modularity)

- M.E.J. Newman. Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E 74, 2006.
- R. Guimera, M. Sales-Pardo, L.A.N. Amaral. Modularity from Fluctuations in Random Graphs and Complex Networks. Phys. Rev. E 70, 2004.
- J. Duch and A. Arenas. Community detection in complex networks using Extremal Optimization. Phys. Rev. E 72, 2005.
- A. Arenas, J. Duch, A. Fernandez, and S. Gomez. Size reduction of complex networks preserving modularity. New Journal of Physics 9(6), 2007.
- E.A. Leicht and M.E.J. Newman. Community structure in directed networks. Phys. Rev. Lett. 100, 2008.
- V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri. Extending the definition of modularity to directed graphs with overlapping communities. J. Stat. Mech. 03, 2009.
- S. Muff, F. Rao, A. Caflisch. Local modularity measure for network clusterizations. Phys. Rev. E, 72, 2005.
- S. Fortunato and M. Barthelemy. Resolution limit in community detection. PNAS 104(1), 2007.
- Finding community structure in very large networks, Aaron Clauset,M. E. J. Newman, and Cristopher Moore, http://arxiv.org/pdf/cond-mat/0408187v2.pdf
- Near linear time algorithm to detect community structures in large-scale networks, Phys. Rev. E 76, 036106 – Published 11 September 2007

References (community evaluation measures)

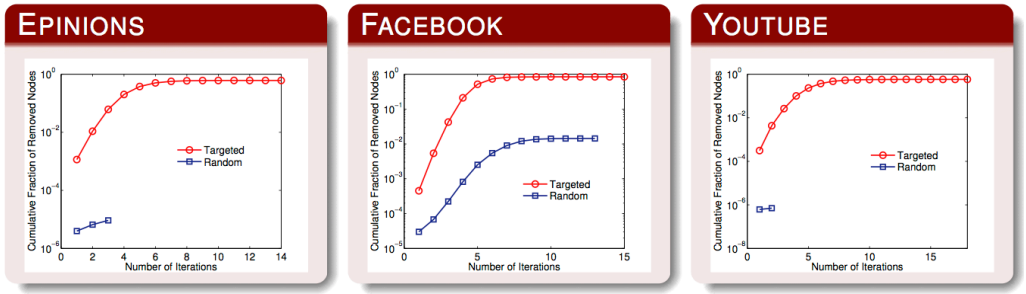
- M.E.J. Newman. The structure and function of complex networks. SIAM REVIEW 45, 2003.
- M.E.J. Newman and M. Girvan. Finding and evaluating community structure in networks. Physical Review E 69(02), 2004.
- S.E. Schaeffer. Graph clustering. Computer Science Review 1(1), 2007.
- S. Fortunato. Community detection in graphs. Physics Reports 486 (3-5), 2010.
- L. Danon, J. Duch, A. Arenas, and A. Diaz-guilera. Comparing community structure identification. Journal of Statistical Mechanics: Theory and Experiment 9008 , 2005.
- M. Coscia, F. Giannotti, and D. Pedreschi. A classification for community discovery methods in complex networks. Statistical Analysis and Data Mining 4 (5), 2011.
- J. Leskovec, K.J. Lang, and M.W. Mahoney. Empirical comparison of algorithms for network community detection. In: WWW, 2010.
- F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. PNAS, 101(9), 2004.
- J. Yang and J. Leskovec. Defining and Evaluating Network Communities based on Ground-Truth. In: ICDM, 2012.
- Fan Chung. Spectral Graph Theory. CBMS Lecture Notes 92, AMS Publications, 1997.

Graph Mining – Our Research topics

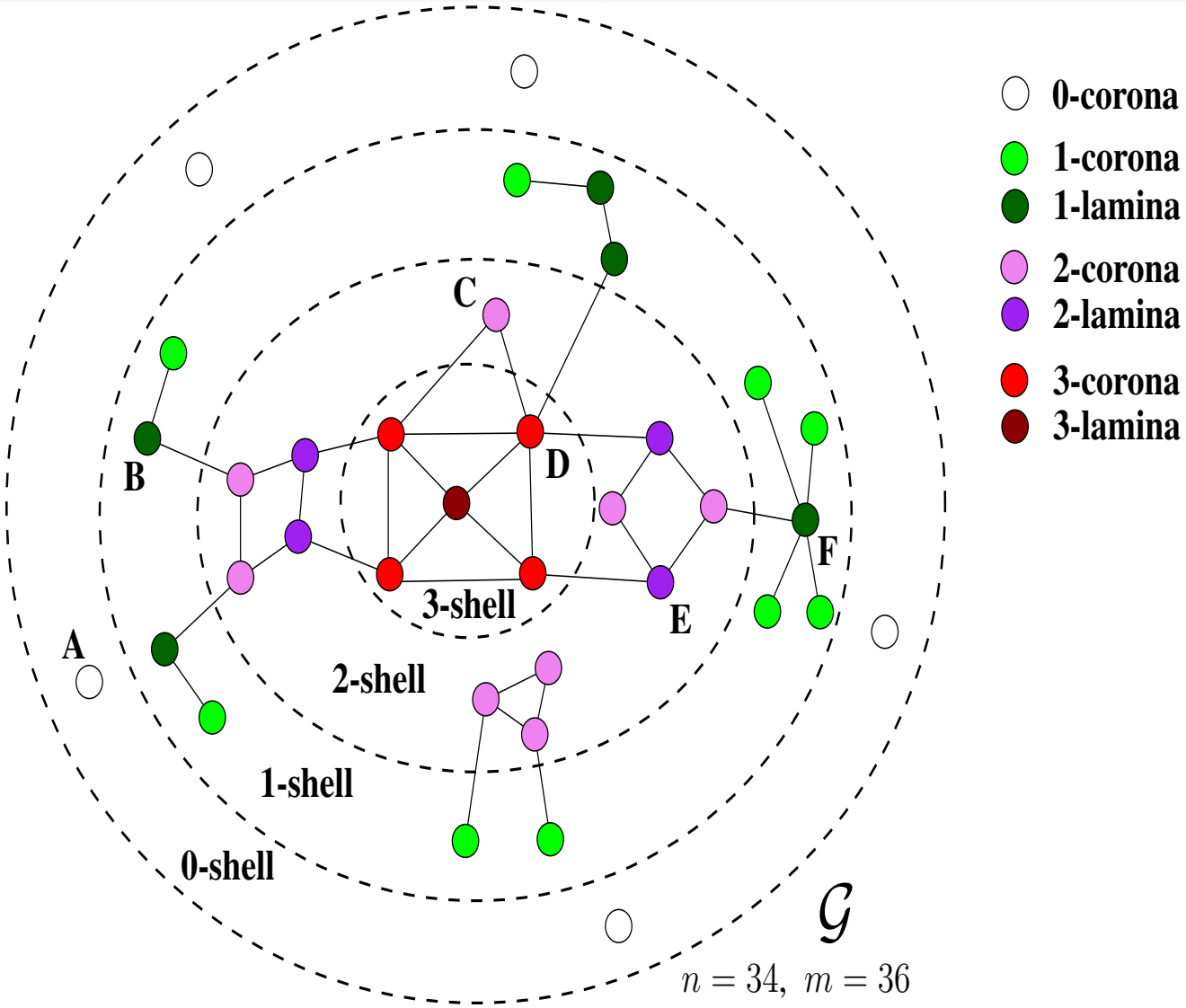
DaSciM
Data Science and Mining Team
École Polytechnique

Home | News | Members | Research | Publications | Teaching | External services | Contact

- Community detection & evaluation
  - Identifying groups of users highly collaborating among them
- Epidemics
  - How to better spread (or imped) a message in a network.
- Community-preserving anonymization of social networks
  - How can I perturb a graph maintaining macroscopic utilities.
- Graph kernels
  - Similarity measures among graphs for better graph classification (i.e. proteomics)
- Clustering acceleration

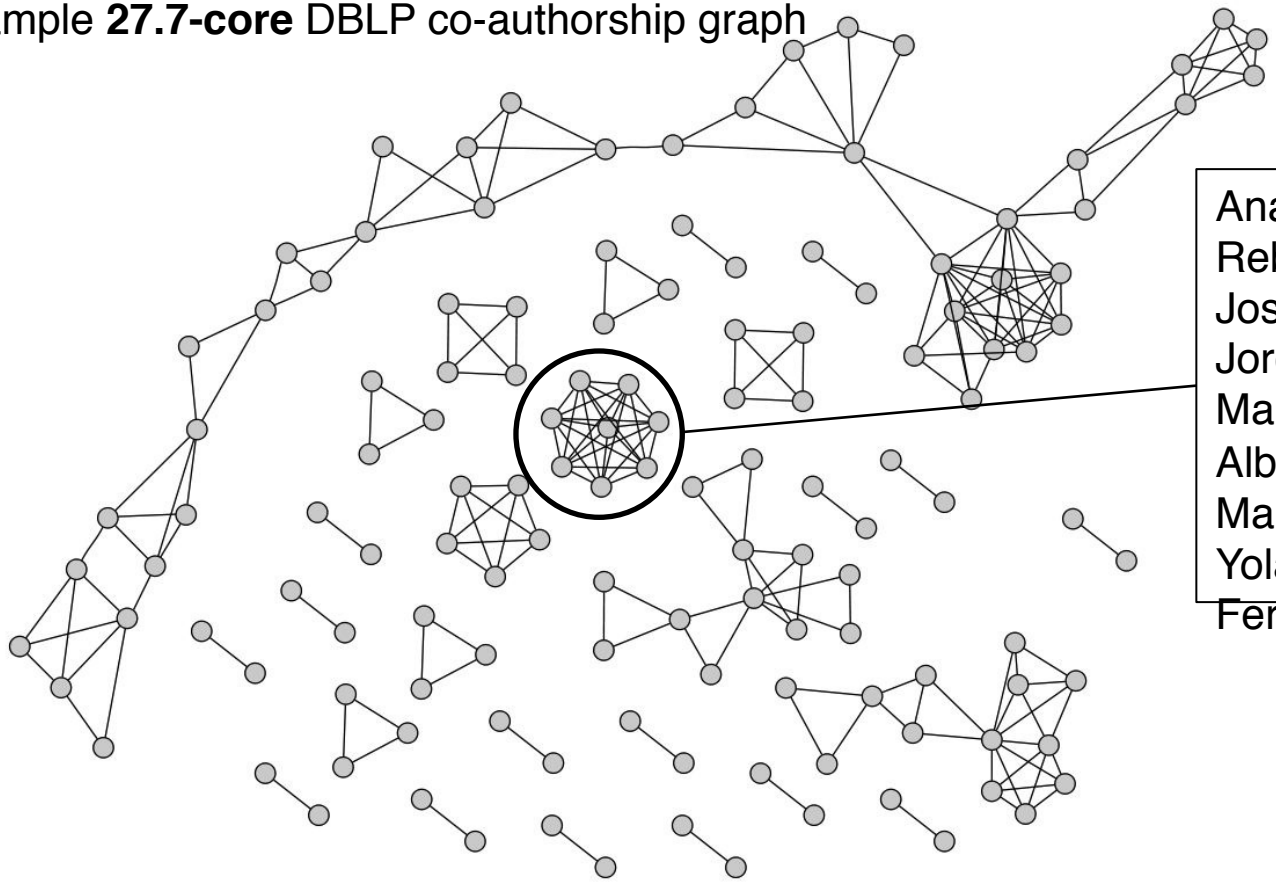


Graph Mining – k-core concept



Community detection and evaluation

Example **27.7-core** DBLP co-authorship graph

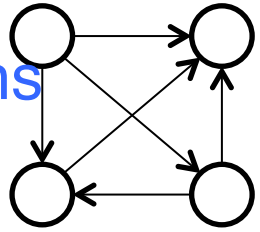


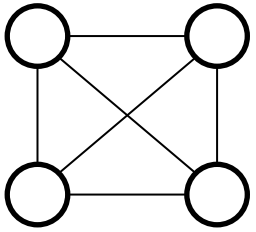
Ana Fernandez Vilas
Rebeca P. Diaz Redondo
Jose J. Pazos Arias
Jorge Garcia Duque
Martin Lopez Nores
Alberto Gil-Solla
Manuel Ramos Cabrer
Yolanda Blanco-Fernandez

http://www.graphdegeneracy.org/

Community detection and evaluation

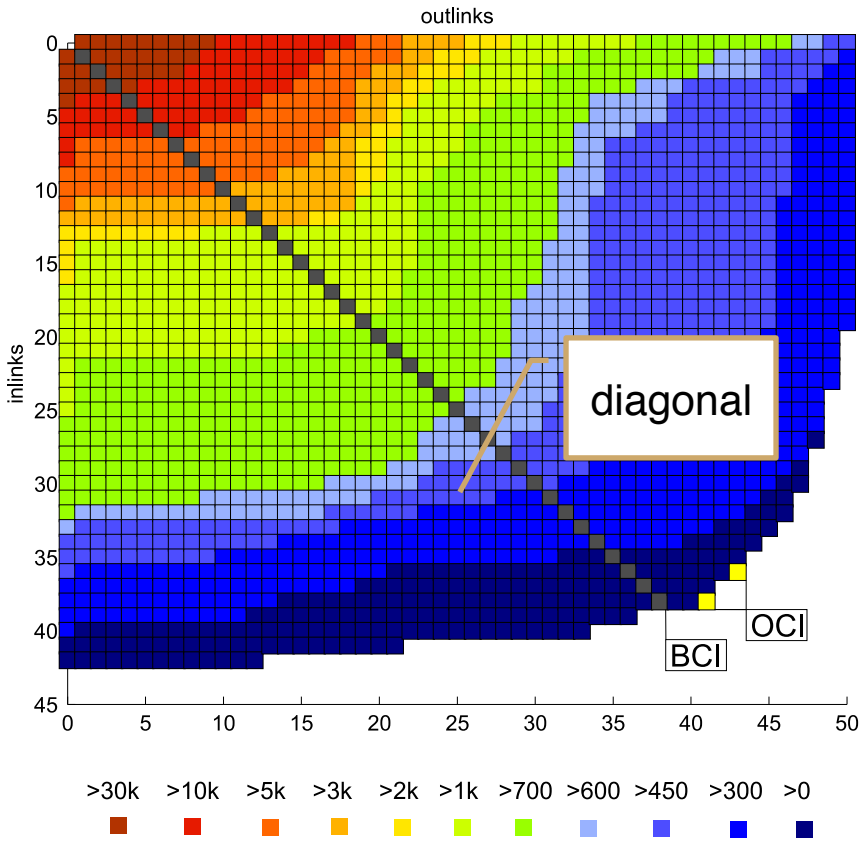
Degeneracy in directed graphs



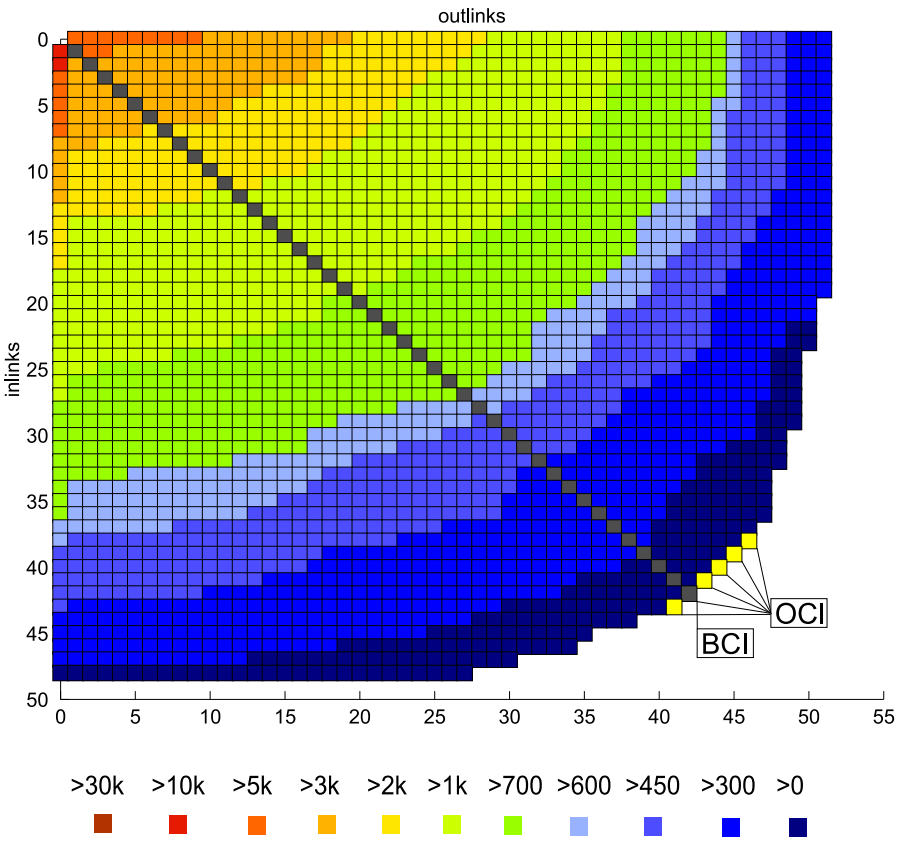


- Directed graphs:
  - WIKI - graph
  - DBLP & ARXIV – Citation graph
- Is there a degeneracy notion for directed graphs?
- We extend the k-core concept in directed graphs by applying a limit on in/out edges respectively
- Trade off between in/out edges can give us a more specific view of the cohesiveness and the “social" behavior

D-core matrix Wikipedia & DBLP



Wikipedia
The extreme D-core(38,41) contains 237 pages



DBLP
One of the extreme D-cores(38,46) contains 188 authors

The Extreme DBLP citation graph D-core

José A. Blakeley
Hector Garcia-Molina
Abraham Silberschatz
Umeshwar Dayal
Eric N. Hanson
Jennifer Widom
Klaus R. Dittrich
Nathan Goodman
Won Kim
Alfons Kemper
Guido Moerkotte
Clement T. Yu
M. Tamer Ã Zsu
Amit P. Sheth
Ming-Chien Shan
Richard T. Snodgrass
David Maier
Michael J. Carey
David J. DeWitt
Joel E. Richardson
Eugene J. Shekita
Waqar Hasan
Marie-Anne Neimat
Darrell Woelk
Roger King
Stanley B. Zdonik
Lawrence A. Rowe
Michael Stonebraker
Serge Abiteboul
Richard Hull
Victor Vianu
Jeffrey D. Ullman
Michael Kifer
Philip A. Bernstein
Vassos Hadzilacos
Elisa Bertino
Stefano Ceri
Georges Gardarin

Patrick Valduriez
Ramez Elmasri
Richard R. Muntz
David B. Lomet
Betty Salzberg
Shamkant B. Navathe
Arie Segev
Gio Wiederhold
Witold Litwin
Theo Härder
François Bancilhon
Raghu Ramakrishnan
Michael J. Franklin
Yannis E. Ioannidis
Henry F. Korth
S. Sudarshan
Patrick E. O'Neil
Dennis Shasha
Shamim A. Naqvi
Shalom Tsur
Christos H. Papadimitriou
Georg Lausen
Gerhard Weikum
Kotagiri Ramamohanarao
Maurizio Lenzerini
Domenico Saccà
Giuseppe Pelagatti
Paris C. Kanellakis
Jeffrey Scott Vitter
Letizia Tanca
Sophie Cluet
Timos K. Sellis
Alberto O. Mendelzon
Dennis McLeod
Calton Pu
C. Mohan
Malcolm P. Atkinson
Doron Rotem

Michel E. Adiba
Kyuseok Shim
Goetz Graefe
Jiawei Han
Edward Sciore
Rakesh Agrawal
Carlo Zaniolo
V. S. Subrahmanian
Claude Delobel
Christophe Lecluse
Michel Scholl
Peter C. Lockemann
Peter M. Schwarz
Laura M. Haas
Arnon Rosenthal
Erich J. Neuhold
Hans-Jorg Schek
Dirk Van Gucht
Hamid Pirahesh
Marc H. Scholl
Peter M. G. Apers
Allen Van Gelder
Tomasz Imielinski
Yehoshua Sagiv
Narain H. Gehani
H. V. Jagadish
Eric Simon
Peter Buneman
Dan Suciu
Christos Faloutsos
Donald D. Chamberlin
Setrag Khoshafian
Toby J. Teorey
Randy H. Katz
Miron Livny
Philip S. Yu
Stanley Y. W. Su
Henk M. Blanken

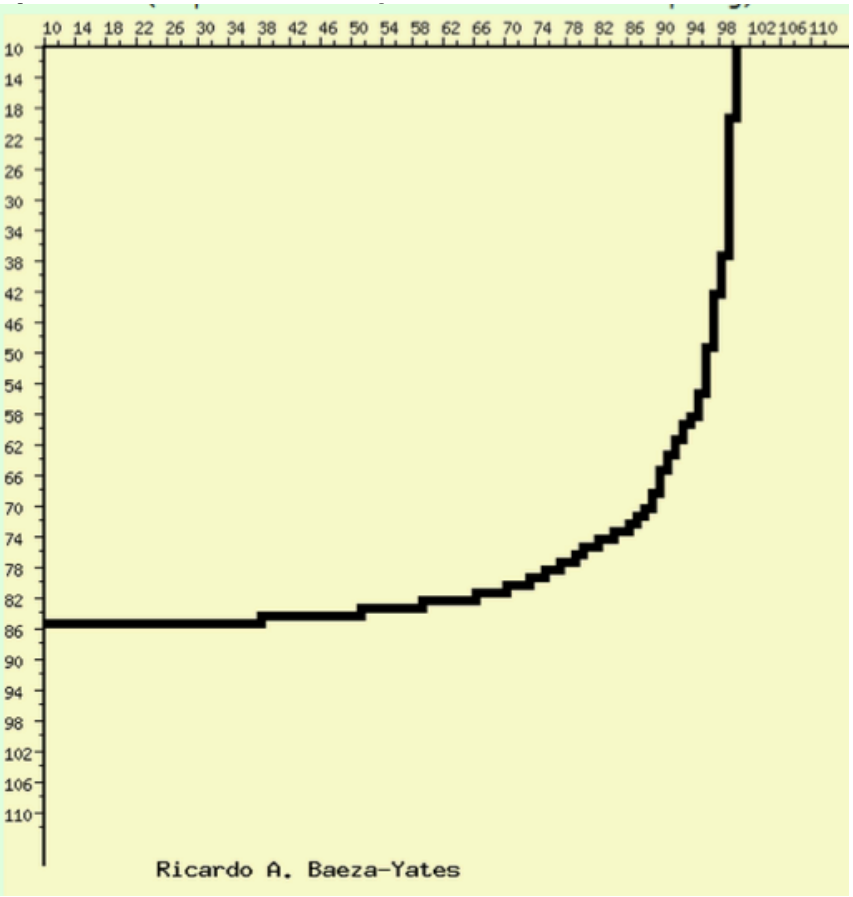
Peter Pistor
Matthias Jarke
Moshe Y. Vardi
Daniel BarbarÃ¡
Uwe Deppisch
H.-Bernhard Paul
Don S. Batory
Marco A. Casanova
Joachim W. Schmidt
Guy M. Lohman
Bruce G. Lindsay
Paul F. Wilms
Z. Meral Özsoyoglu
Gultekin Özsoyoglu
Kyu-Young Whang
Shahram Ghandeharizadeh
Tova Milo
Alon Y. Levy
Georg Gottlob
Johann Christoph Freytag
Klaus Küspert
Louiqa Raschid
John Mylopoulos
Alexander Borgida
Anand Rajaraman
Joseph M. Hellerstein
Masaru Kitsuregawa
Sumit Ganguly
Rudolf Bayer
Raymond T. Ng
Daniela Florescu
Per-Åke Larson
Hongjun Lu
Ravi Krishnamurthy
Arthur M. Keller
Catriel Beeri
Inderpal Singh Mumick
Oded Shmueli

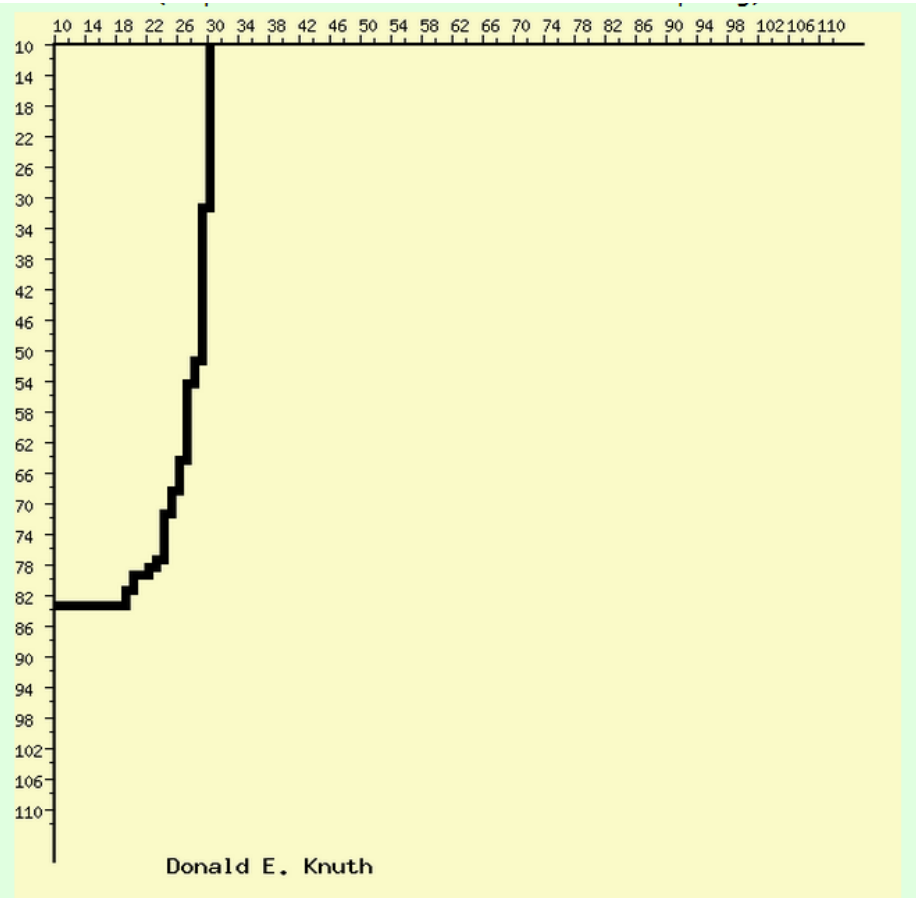
George P. Copeland
Peter Dadam
Susan B. Davidson
Donald Kossmann
Christophe de Maindreville
Yannis Papakonstantinou
Kenneth C. Sevcik
Gabriel M. Kuper
Peter J. Haas
Jeffrey F. Naughton
Nick Roussopoulos
Bernhard Seeger
Georg Walch
R. Erbe
Balakrishna R. Iyer
Ashish Gupta
Praveen Seshadri
Walter Chang
Surajit Chaudhuri
Divesh Srivastava
Kenneth A. Ross
Arun N. Swami
Donovan A. Schneider
S. Seshadri
Edward L. Wimmers
Kenneth Salem
Scott L. Vandenberg
Dallan Quass
Michael V. Mannino
John McPherson
Shaul Dar
Sheldon J. Finkelstein
Leonard D. Shapiro
Anant Jhingran
George Lapis

D-Core frontier for individuals

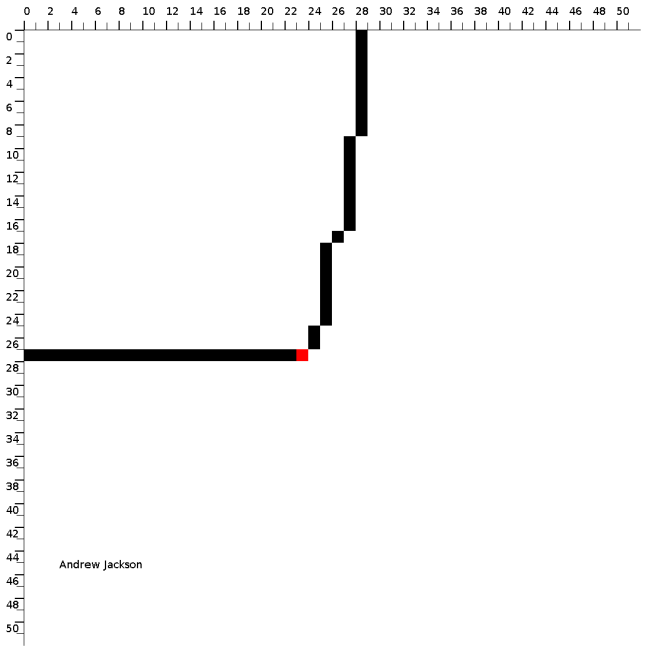
- The frontier of an individual: defined by

outlinks

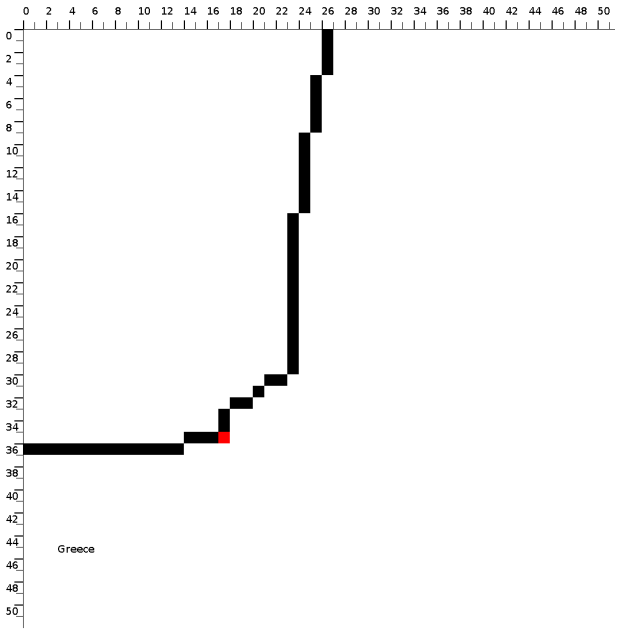




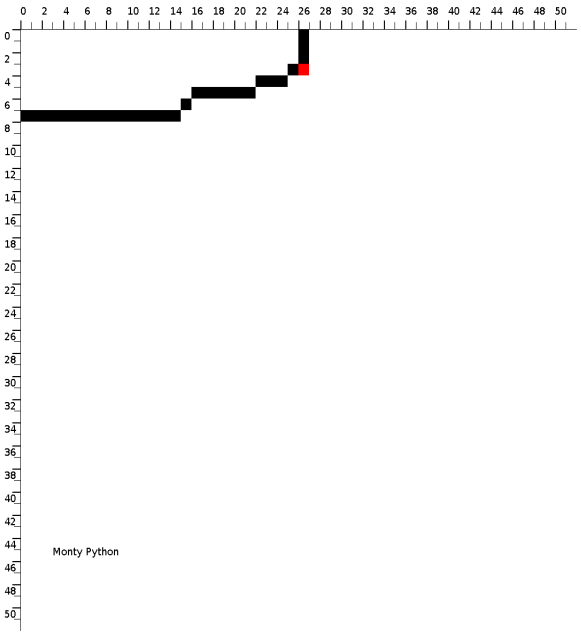
Thematic D-core frontiers - Wikipedia



“Andrew Jackson”



“Greece”



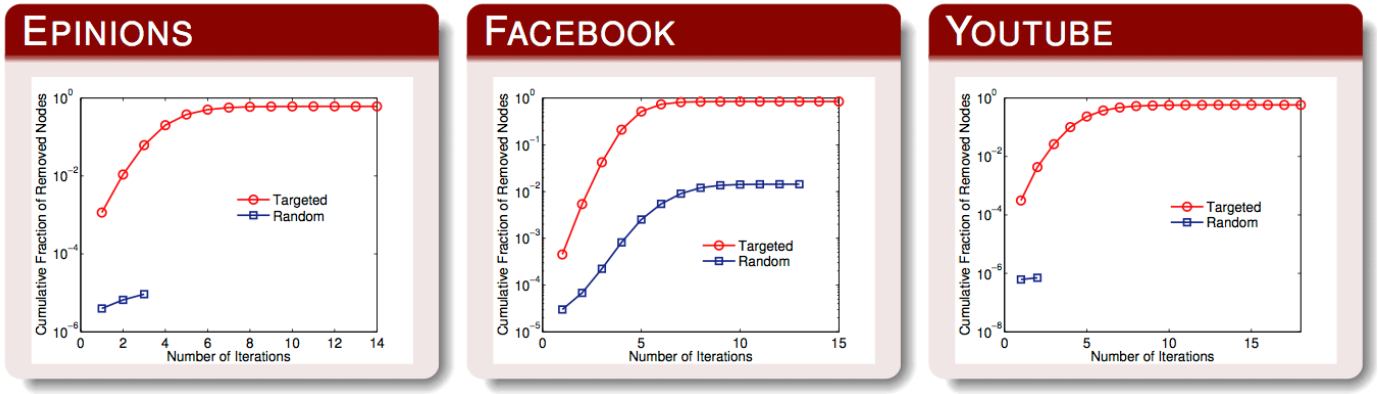
“Monty Python”

Engagement/departure dynamics and vulnerability in social networks

- Evaluating the engagement of a user in a network (i.e. low probability of departure) - churn
- Evaluate departure effect to the network – i.e. how can you make a SC collapse

Network vulnerability

- Nodes may depart depart the graph, causing an epidemic or cascading departures
- What is the effect on the graph?
- Which nodes departures are more significant?
- Strategies for selecting departing node: random vs. targeted departure
- **Cascading Departure (CasD) model**
  - ***k***-core decomposition based model to capture the cascading (epidemic) disengagement effect due to the departure of a node



- Social networks are
  - extremely robust under departures of random nodes
  - highly vulnerable under cascades starting from targeted departures