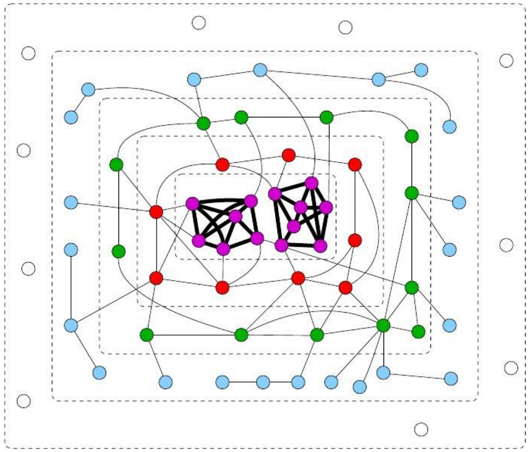
Graph Mining Tools for Community Detection & Evaluation in Social Networks & the Web



Christos Giatsidis
École Polytechnique, France

Fragkiskos D. Malliaros
École Polytechnique, France

Michalis Vazirgiannis
AUEB, Greece
École Polytechnique, France
Télécom ParisTech, France

22nd International World Wide Web Conference (WWW)

May 13-17, 2013 | Rio de Janeiro, Brazil

Outline

1. **Introduction & Motivation**
2. **Graph fundamentals**
3. **Community evaluation measures**
4. **Graph clustering algorithms**
5. **Clustering and community detection in directed graphs**
6. **Alternative Methods for Community Evaluation**
7. **New directions for research in the area of graph mining**

Outline

1. **Introduction & Motivation**
2. Graph fundamentals
3. Community evaluation measures
4. Graph clustering algorithms
5. Clustering and community detection in directed graphs
6. Alternative Methods for Community Evaluation
7. New directions for research in the area of graph mining

Social Networks growth..

- Social networking accounts for 1 of every 6 minutes spent online [http://blog.comscore.com/]
- One in every nine people on Earth is on Facebook
- Each Facebook user spends on average 15 hours and 33 minutes a month on the site
- 30 billion pieces of content is shared on Facebook each month
- 300,000 users *helped* translate Facebook into 70 languages
- People on Facebook install 20 million "Apps" every day

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

Social Networks Growth..

- YouTube has 490 million unique users who visit every month (02/2011)
- Users on YouTube spend a total of 2.9 billion hours per month (326,294 years)!
- Wikipedia hosts 17 million articles and has over 91,000 contributors
- People upload 3,000 images to Flickr every minute and hosts over 5 billion images!
- 190 million average Tweets per day occur on Twitter (May 2011)
- Twitter is handling 1.6 billion queries per day
- Google+ was the fastest social network to reach 10 million users at 16 days (Twitter took 780 days and Facebook 852 days)

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

Graphs are everywhere

- The WWW is a directed graph
- Social & citation Networks constitute inherently Graphs
- Such graphs can be directed (WWW) and or signed (trust networks)
- High dynamics: constantly changing in both “shape” and size”

Communities in social nets

- 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 . This property is called clustering or community structure and is of great interest

Community detection

- community detection in graphs aims to identify the modules and, possibly, their hierarchical organization, by only using the information encoded in the graph topology.
- First attempt dates back to 1955 by Weiss and Jacobson searching for work groups within a government agency.

Communities – application domains

- Social communities have been studied for a long time (Coleman, 1964; Freeman, 2004; Kottak, 2004; Moody and White, 2003).
- *In biology* - protein-protein interaction networks, communities are likely to group proteins having the same specific function within the cell (Chen, 2006; Rives and Galitski 2003; Spirin and Mirny, 2003),
- *World Wide Web*: communities correspond to groups of pages dealing with the same or related topics (Dourisboure et al., 2007; Flake et al., 2002),
- *metabolic networks* they may be related to functional modules such as cycles and pathways (Guimera and Amaral, 2005; Palla et al., 2005),
- *in food webs* they may identify compartments (Krause et al., 2003; Pimm, 1979)

Community evaluation

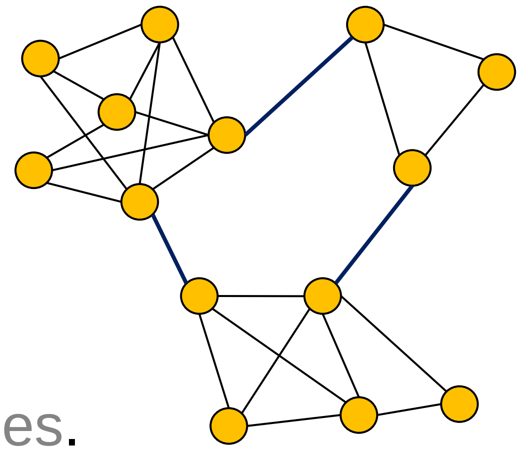
- *Community detection* and *evaluation* in graphs is a cornerstone issue .
- Different metrics/ measurements /methods are used
  - Hub/authorities
  - Modularity
  - Density/Diameter/Link distribution etc….
  - Centrality/Betweenness
  - Clustering coefficient
  - Structural cohesion
- A thorough state of the art review is offered by Fortunato

Santo Fortunato. Community detection in graphs. Physics Reports, 486(3-5):75-174, 2010.

Outline

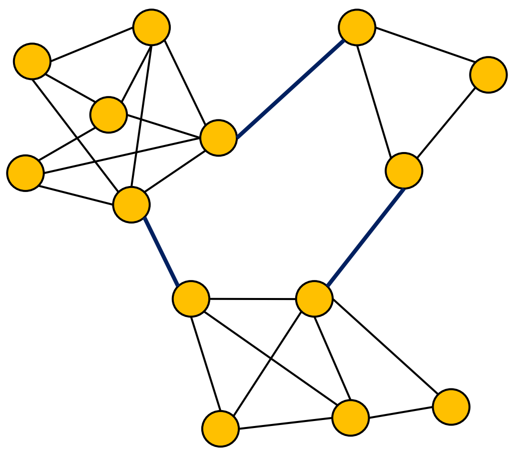
1. Introduction & Motivation
2. **Graph fundamentals**
3. Community evaluation measures
4. Graph clustering algorithms
5. Clustering and community detection in directed graphs
6. Alternative Methods for Community Evaluation
7. New directions for research in the area of graph mining

Basic graph notation



- A graph consists of vertices and edges.
- Edges can be directed/undirected, weighted/un weighted
- The adjacency matrix W represents the graph:
  - w_{ij} = 0 if *i* and *j* are not connected
  - w_{ij} > 0 if *i* and *j* are connected
- The degree f a vertex is the sum of all the adjacent edge weights: = $d_i = \sum_j w_{ij}$
- All vertices that can be reached pairwise by a path form a connected component.

Basic graph notation



- W = (w_{ij}): adjacency matrix
- $d_i = \sum_j w_{ij}$: degree of a vertex
- D = *diag(d1, …dn):* degree matrix
- |A| = # vertices in the graph A
- $Vol(A) = \sum_{i \in A} d_i$

Graph Clusters & Communities

- There is no widely accepted definition
- Generally a community is a cluster of nodes in a social network graph
- In general the graph has to be relatively sparse.
- If the graph is too dense then there is no meaning in search of a cluster using the structural properties of the graph.
- Many clustering algorithms or problems related to clustering are NP-hard

Common graph models

- Random graph (**Erdős–Rényi** model -1959)
  - model for generating random graphs,
  - an edge is created between each pair of nodes with equal probability, independently of the other edges.
- **scale-free network**
  - degree distribution follows a power law, at least asymptotically: the fraction $P(k)$ of nodes in the network having k connections is (for large values of k): $P(k) \sim k^{-\gamma}$, $2<\gamma<3$
  - Many real networks are conjectured to be scale-free, (World Wide Web links, biological networks, and social networks)
  - Preferential attachment and the fitness model have been proposed as mechanisms to explain conjectured power law degree distributions in real networks.

Outline

1. Introduction & Motivation
2. Graph fundamentals
3. **Community evaluation measures**
4. Graph clustering algorithms
5. Clustering and community detection in *directed graphs*
6. Alternative Methods for Community Evaluation
7. New directions for research in the area of graph mining

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)
- Set of nodes with **more/better/stronger** connections between its members, than to the rest of the network
- Why this happens?
  - Individuals are typically organized into social groups (e.g., family, associations, profession)
  - Web pages can form groups according to their topic
  - …

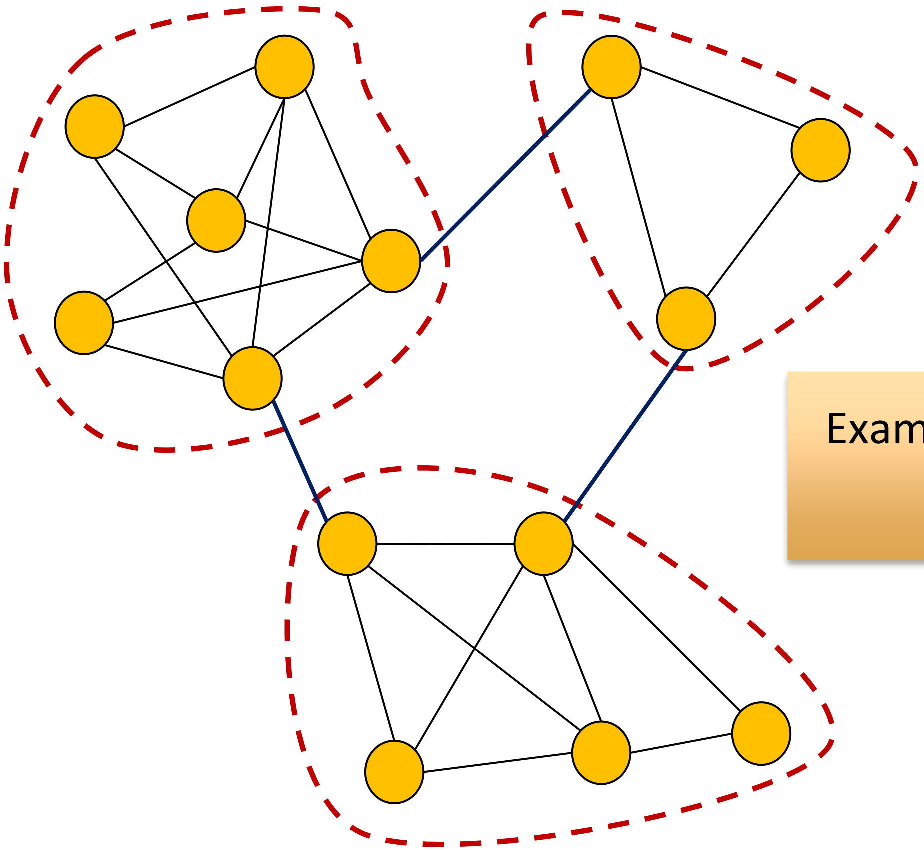
Definition/notion of communities

- How a community in graphs looks like?
- The property of community structure is **difficult** to be defined
  - There is no universal definition of the problem
  - It depends heavily on the application domain and the properties of the graph under consideration
- Most widely used notion/definition of communities is based on the number of edges within a group (density) compared to the number of edges between different groups

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

[Newman ‘03], [Newman and Girvan ‘04], [Schaeffer ‘07], [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

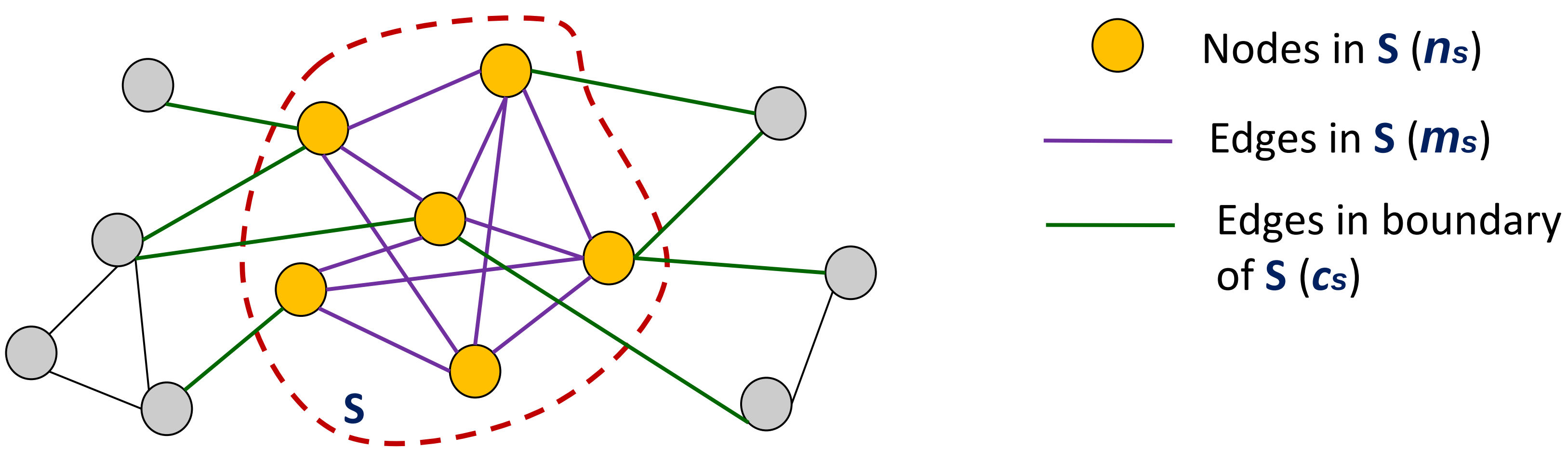
- Focus on
  - Intra-cluster edge density (# of edges within community),
  - Inter-cluster edge density (# of edges across communities)
  - Both two criteria

- 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***

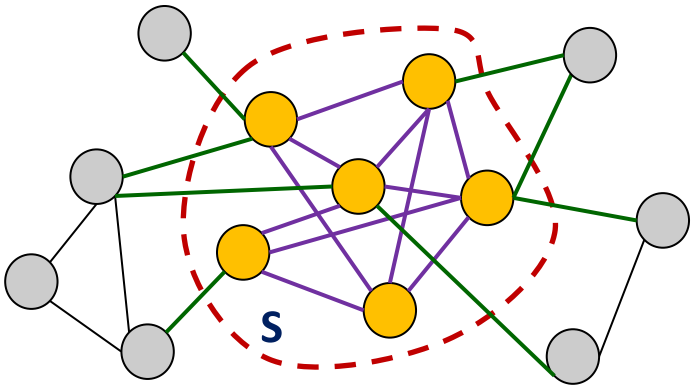


Evaluation based on internal connectivity (1)

- **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]

$$f(S) = m_s$$

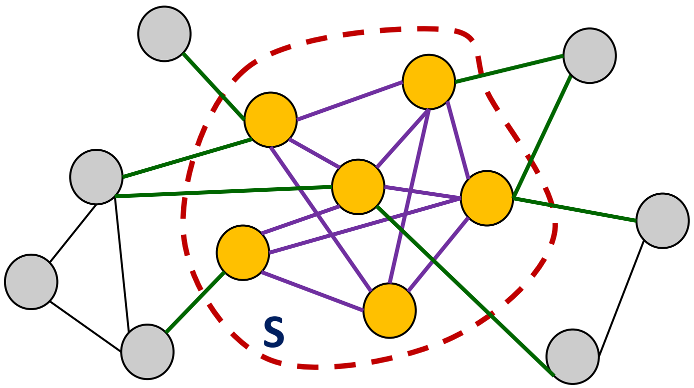
Number of edges between the nodes of **S**

Evaluation based on internal connectivity (2)

- **Average degree** [Radicchi et al. '04]

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

Average internal degree of nodes in **S**



- **Fraction over median degree (FOMD)** [Yang and Leskovec '12]

$$f(S) = \frac{\left| \left\{ u : u \in S, \left| \left\{ (u,v) : v \in S \right\} \right| > d_m \right\} \right|}{n_s}$$

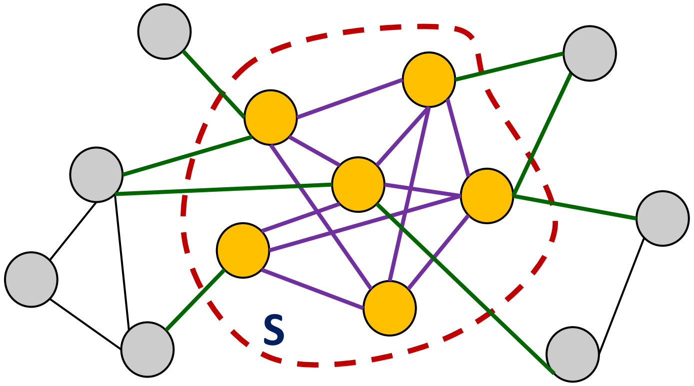
Fraction of nodes in **S** with internal degree greater than ***d**m*, where ***d**m* **= median (*d**u*)**

Evaluation based on internal connectivity (3)

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

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

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

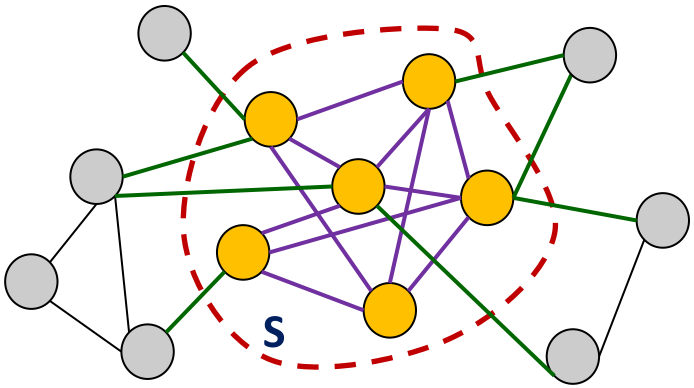


Evaluation based on external connectivity

- **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]

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

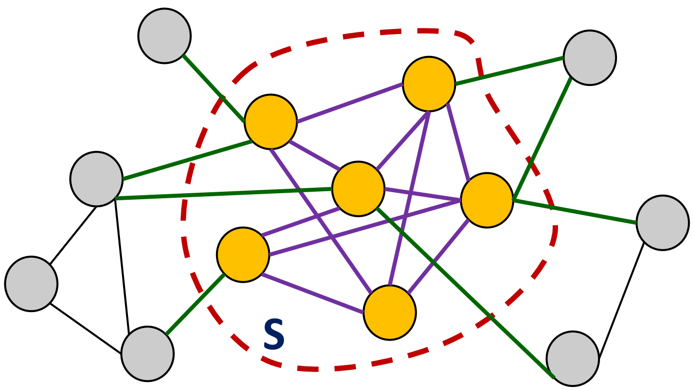
Fraction of existing edges – out of all possible edges – that leaving **S**

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 outside **S**



- **Normalized cut** [Shi and Malic '00]

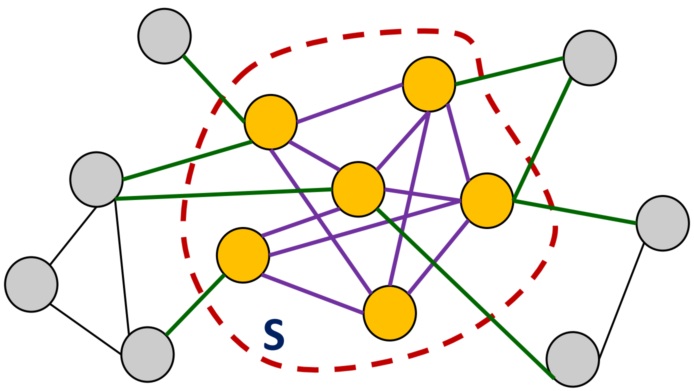
$$f(S) = \frac{c_s}{2m_s + c_s} + \frac{c_s}{2(m - m_s) + c_s}$$

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

Evaluation based on internal and external connectivity (2)

- **Maximum out degree fraction (Max ODF)** [Flake et al '00]

$$f(S) = \max_{u \in S} \frac{|\{(u,v) \in E : v \notin S\}|}{d_u}$$



Measures the maximum fraction of edges of a node in **S** that point outside **S**

- **Average out degree fraction (Avg ODF)** [Flake et al '00]

$$f(S) = \frac{1}{n_s} \sum_{u \in S} \frac{|\{(u,v) \in E : v \notin S\}|}{d_u}$$

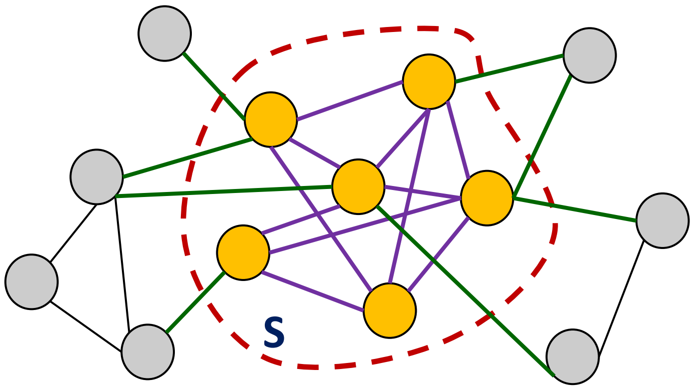
Measures the average fraction of edges of nodes in **S** that point outside **S**

Evaluation based on internal and external connectivity (3)

- **Flake's out degree fraction (Flake's ODF)** [Flake et al '00]

$$f(S) = \frac{\left| \left\{ u : u \in S, \left| \left\{ (u,v) \in E : v \in S \right\} \right| < d_u / 2 \right\} \right|}{n_s}$$

Measures the fraction of nodes in **S** that have fewer edges pointing inside than outside of **S**



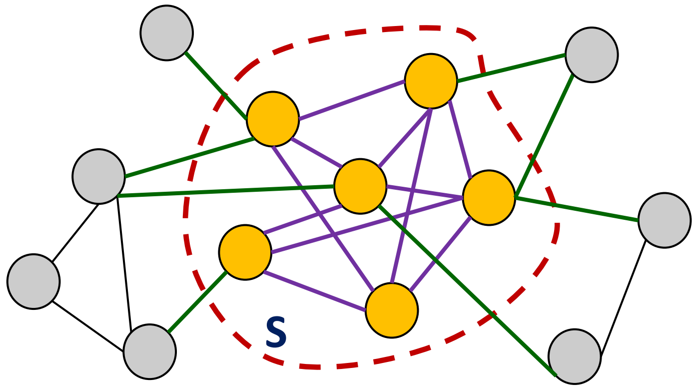
Evaluation based on network model

- **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

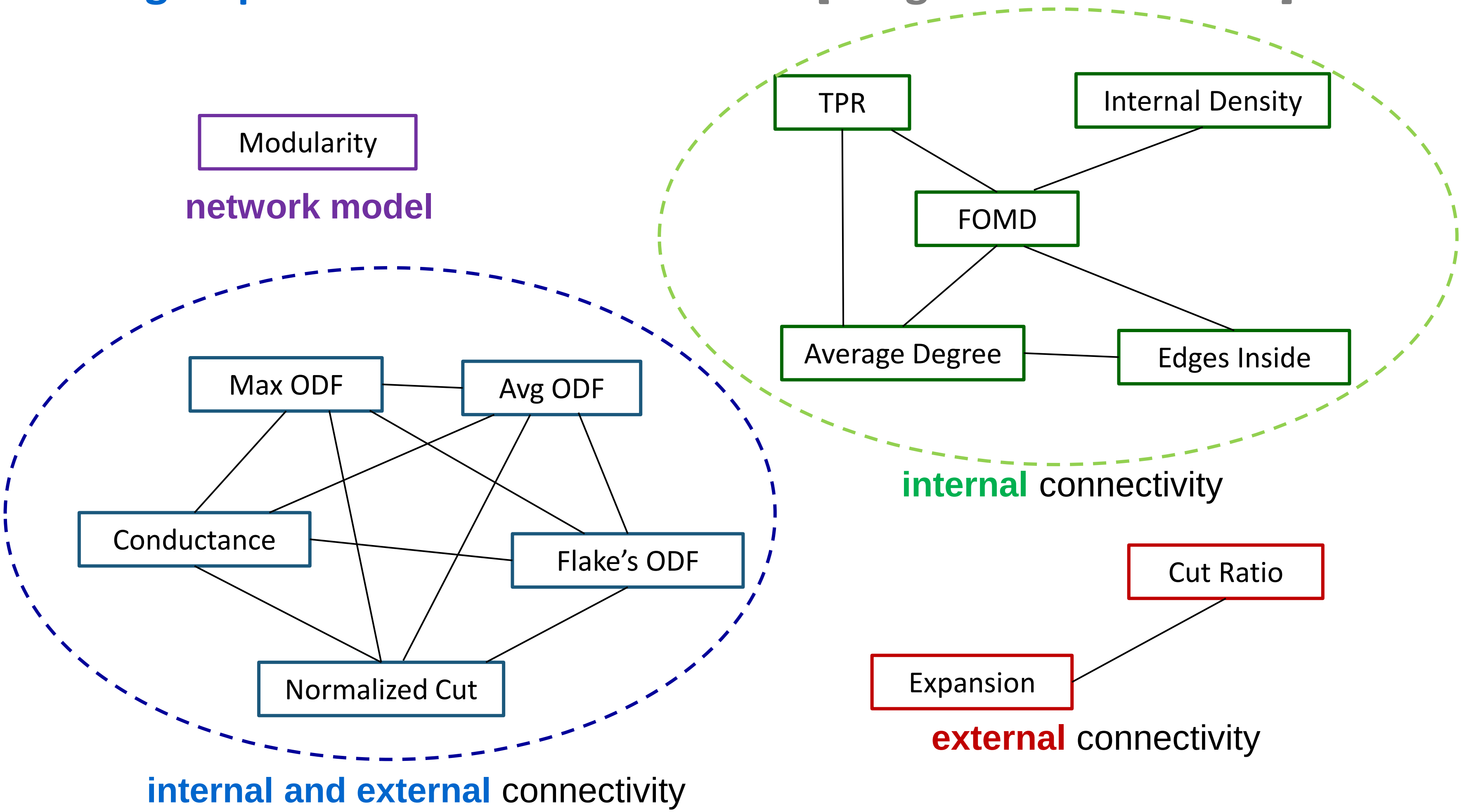


How different are the evaluation measures? (1)

- Several community evaluation measures (objective criteria) have been proposed
- Is there any **relationship** between them?
- Consider real graphs with **known node assignment to communities (ground-truth information)** and test the behavior of the objective measures **[Yang and Leskovec '12]**
  1. For each of the ground-truth communities **S**
  2. Compute the score of **S** using each of the previously described evaluation measures
  3. Form the **correlation matrix** of the objective measures based on the scores
  4. Apply a threshold in the correlation matrix
  5. Extract the correlations between community objective measures

How different are the evaluation measures? (2)

- **Observation:** Community evaluation measures form **four groups** based on their correlation [Yang and Leskovec '12]



How different are the evaluation measures? (3)

- The different structural definitions of communities are **heavily correlated** [Yang and Leskovec '12]
- Community evaluation measures form **four groups** based on their correlation
- These groups correspond to the four main notions of structural communities
  - Communities based on **internal** connectivity
  - Communities based on **external** connectivity
  - Communities based on **internal and external** connectivity
  - Communities based on a **network model** (modularity)

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.

References (community evaluation measures)

- J. Shi and J. Malik. Normalized Cuts and Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 2000.
- M.E.J. Newman. Modularity and community structure in networks. PNAS, 103(23), 2006.

Outline

1. Introduction & Motivation
2. Graph fundamentals
3. Community evaluation measures
4. **Graph clustering algorithms**
5. Clustering and community detection in directed graphs
6. Alternative Methods for Community Evaluation
7. New directions for research in the area of graph mining

Graph Clustering Algorithms

- **Taxonomy**
- Hierarchical methods
- Spectral Clustering
- Modularity Based Methods

Graph Clustering Algorithms

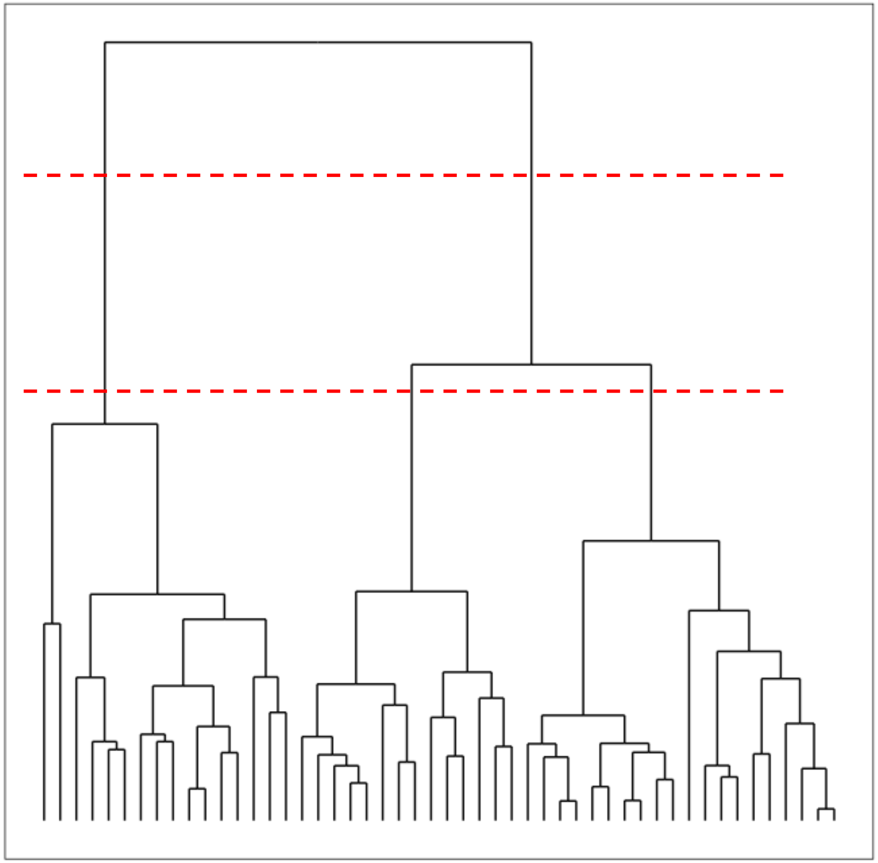
- Taxonomy
  - Hierarchical clustering
    - Divisive algorithms (the algorithm of Girvan and Newman)
  - Spectral clustering
  - Modularity-based methods

Hierarchical graph clustering algorithms

- Clusters form hierarchies
- Need for a cluster similarity measure
  - Single linkage clustering vs. complete linkage
- Agglomerative algorithms, clusters are iteratively merged if their similarity is sufficiently high
- Divisive algorithms, in which clusters are iteratively split by removing edges connecting vertices with low similarity **[Girvan and Newman]** (to be presented later)
- Hierarchical clustering does not require a preliminary knowledge on the number and size of the clusters

Hierarchical clustering

- Dendrogram: request multiple partitions of the data
- High complexity
  - $O(n^2) - O(n^2 \log(n))$



Graph Clustering Algorithms

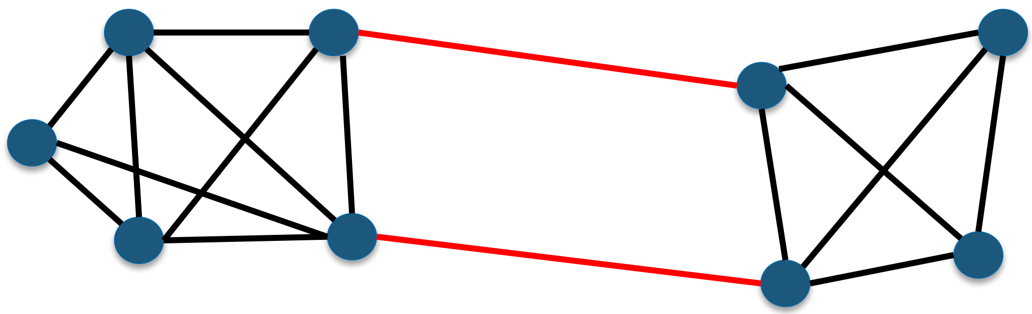
- Taxonomy
- Hierarchical methods
- **Spectral Clustering**
- Modularity Based Methods

Notations

- Given Graph G=(V,E) undirected:
  - Vertex Set $V=\{v_1,\ldots..v_n\}$, 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 $A \subset V$ $\overline{A} = V - A$
  - Given $\mathrm{A}, \mathrm{B} \subset V$ & $\mathrm{A} \cap B = \emptyset$ $w(A,B) = \sum_{v_i \in A, vj \in B} w_{ij}$
  - D: Diagonal matrix where $D(i,i)=d_i$
  - W: Adjacency matrix $W(i,j)=w_{ij}$

Graph-Cut

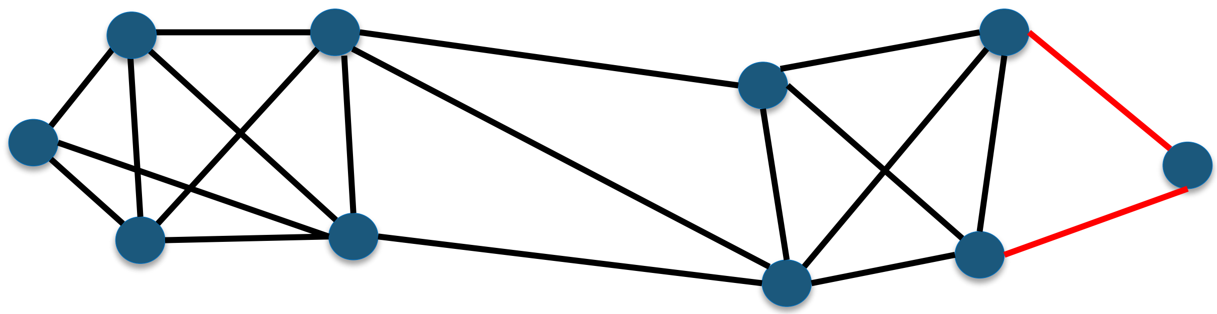
- For k clusters:
  - $cut(A_1, .., Ak) = 1/2 \sum_{i=1}^{k} w(Ai, \overline{A_i})$
    - undirected graph:1/2 we count twice each edge



- 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 $f: V \rightarrow \mathbb{R}$ 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:

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

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

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

Graph Laplacian

- How is the previous useful in Spectral clustering?

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

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

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

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

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

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

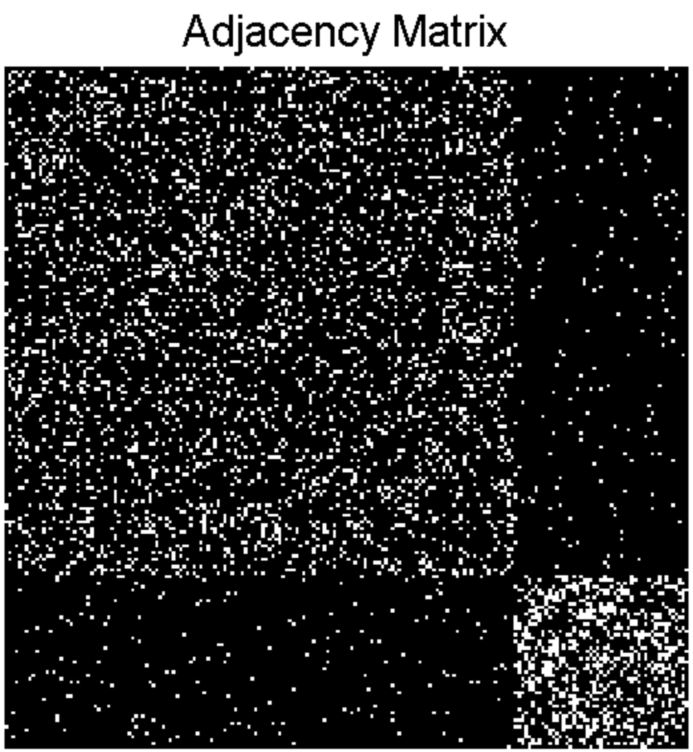
Properties of L

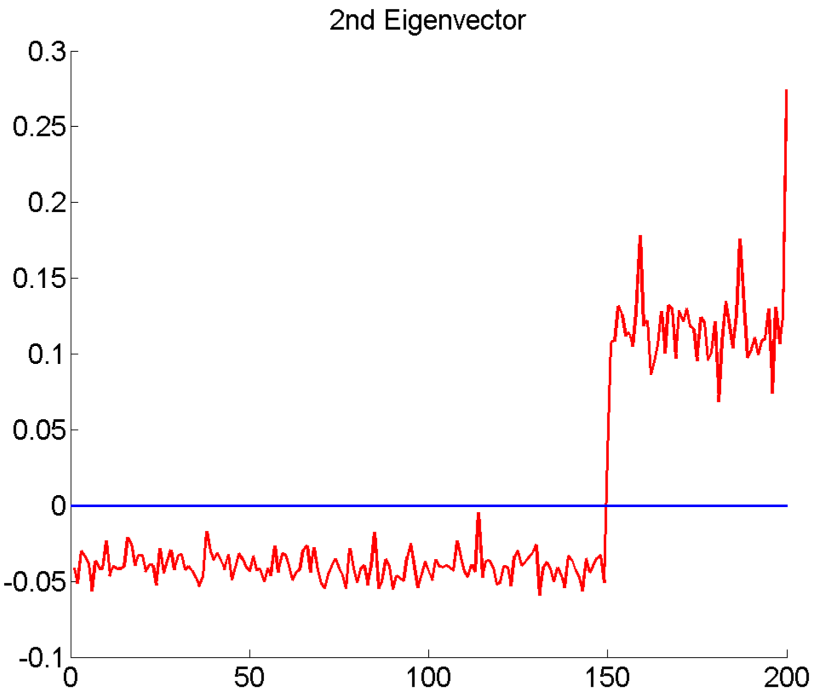
- L is
  - Symmetric
  - Positive
  - Semi-definite
- The smallest eigenvalue of L is 0
  - The corresponding eigenvector is $\mathbb{1}$
- L has n non-negative, real valued eigenvalues
  - $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

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

- $Ratiocut(A_1, \ldots Ak) = \sum_{i=1}^{k} \frac{cut(Ai,\overline{A}_i)}{|Ai|}$
  - 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

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

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

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

- 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 :

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

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

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

Normalized Cut

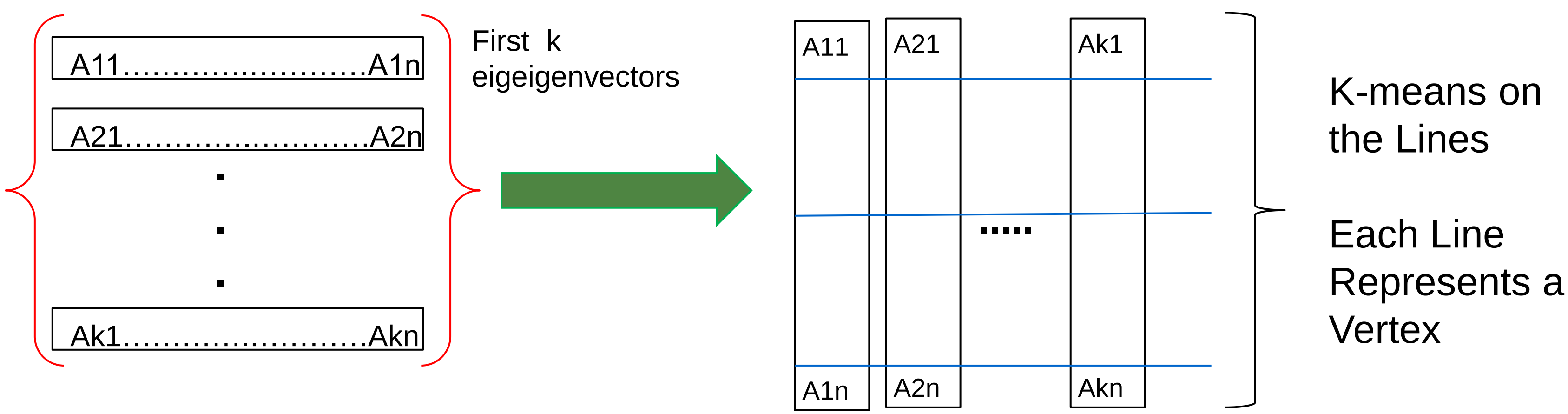
- Similarly we come to : $min_f f^T L f$
  subject to $f^T D 1 = 0,\ fTDf = v(V)$
- Assume $h = D^{1/2} f$
  - $min_h h^T D^{-1/2} \mathrm{L} D^{-1/2} \mathrm{h}$ subject to
    $h^T D^{1/2} 1 = 0, \qquad h^T h = v(V)$
  - The answer is in the eigenvector of the second smallest eigenvalue of $L_{sym} = D^{-1/2} \mathrm{L} D^{-1/2}$
    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

- Define $f_{ij} = \begin{cases} \frac{1}{\sqrt{|Aj|}} & vi \in Aj \\ 0 & othewise \end{cases}$
  - we have a vector indicating the cluster a vertex belongs to
- Similarly to the other equations we can deduce:
  - ${f_i}^T L f_i = cut(Ai, \overline{A_i}) / |Ai|$
  - $\sum_{i=1}^{k} {f_i}^T L f_i = \sum_{i=1}^{k} (F^T L F)_{ii} = Tr(F^T L F)$
    - Where Tr is the Trace of a Matrix
- So now the RatioCut becomes:
  $min(F^T L F)\ subject\ to\ FTF = I$

Multi-Way Graph Partition

- The solution can now be given by the first k eigenvectors of L as columns
- 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}



Algorithm for k>2

Compute Laplacian (L, L_{sym}).
Compute the first k eigenvectors $u_1, \dots, u_k$ of L.
Let $U \in \mathbb{R}^{nxk}$ the matrix containing the vectors u1, . . . , uk as columns.
For i = 1, . . . , n,

let $y_i \in \mathbb{R}^k$ the vector corresponding to the i-th row of U.
Cluster the points $y_i = 1, \dots . n \in \mathbb{R}^k$ with the k-means algorithm into clusters C1, . . . ,Ck.

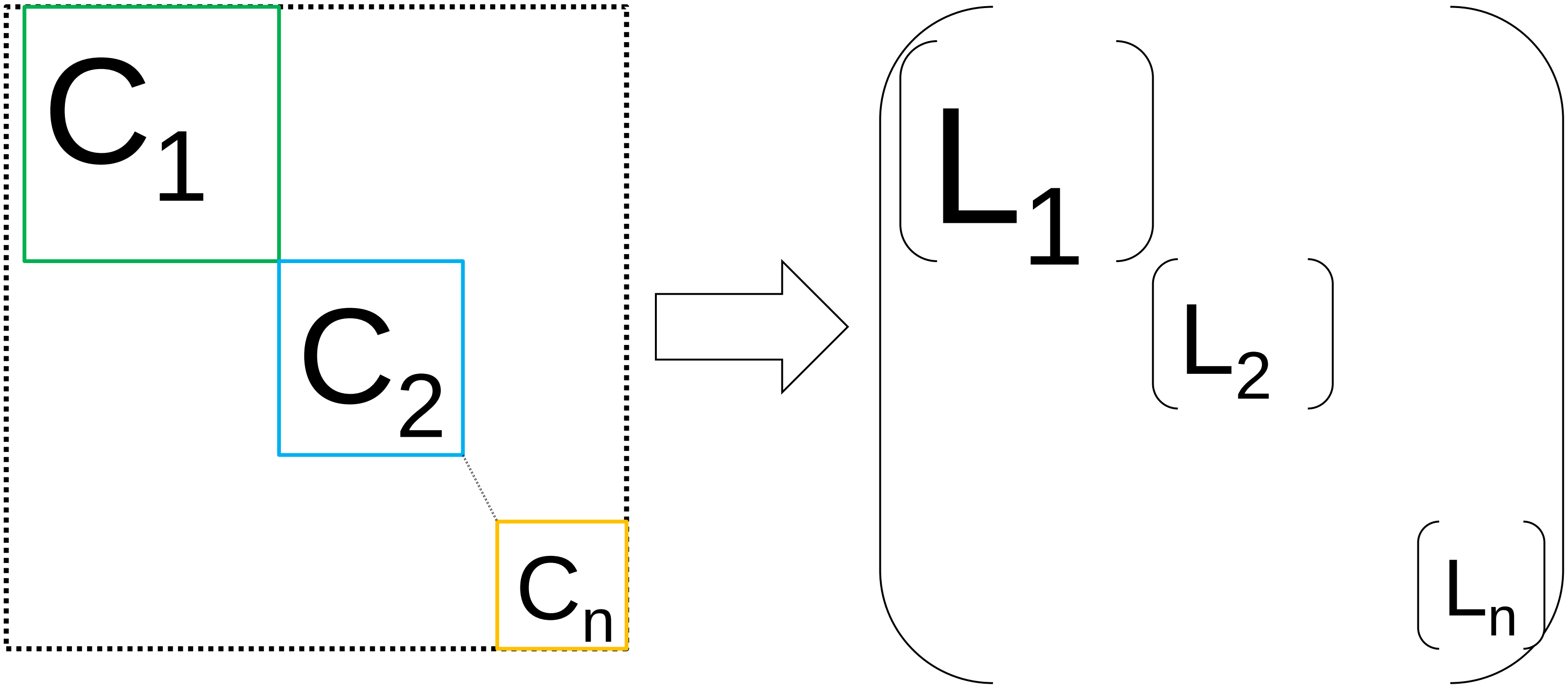
Output: Clusters $A_1, \dots, A_k$ with $Ai = \{j | v_j \in Ci\}$

- **HOW DO WE CHOOSE k?**
  - We choose the k that maximizes the eigengap:

$$\Delta_k = |\lambda_k - \lambda_{k-1}|$$ (**Davis-Kahan Theorem**)

Ideally: for k connected components the Laplacian has k 0-eigenvalues

Laplacian-Eigenvectors-EigenValues



Everything sorted according to cluster : block diagonal form Matrix
L follows the same form composed on $L_1 \ldots L_n$
Each L_i has the same properties as L: nx0 min eigenvalues etc..
Each "Second" eigenvector is a cut of C_i from the rest of the graph and holds a mapping (distance) of a vertex to the cluster i

Simple example

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

2 Eigenvectors

(1100) and (0011)

Mapping vertices in their clusters

Permutation does not change the result

The cut remains the same regardless of the ordering

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

2 Eigenvectors

(1010) and (0101)

Mapping vertices to the same clusters

References

- Ulrike 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 embeded 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

- Taxonomy
- Hierarchical methods
- Spectral Clustering
- **Modularity Based Methods**

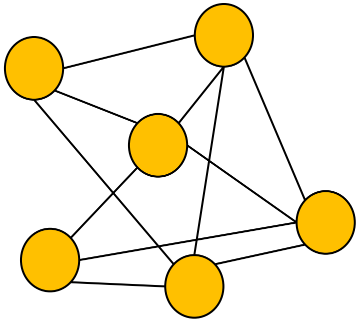
Basics

- Most of the community evaluation measures (e.g., conductance, cut-based measures), quantify the quality of a community based on
  - **Internal connectivity** (intra-community edges)
  - **External connectivity** (inter-community edges)
- **Question:** Is there any other way to distinguish groups of nodes with good community structure?
- **Random graphs** are not expected to present inherent community structure
- **Idea:** Compare the number of edges that lie **within a cluster** with the expected one in case of **random graphs** with the same degree distribution – **modularity measure**

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]

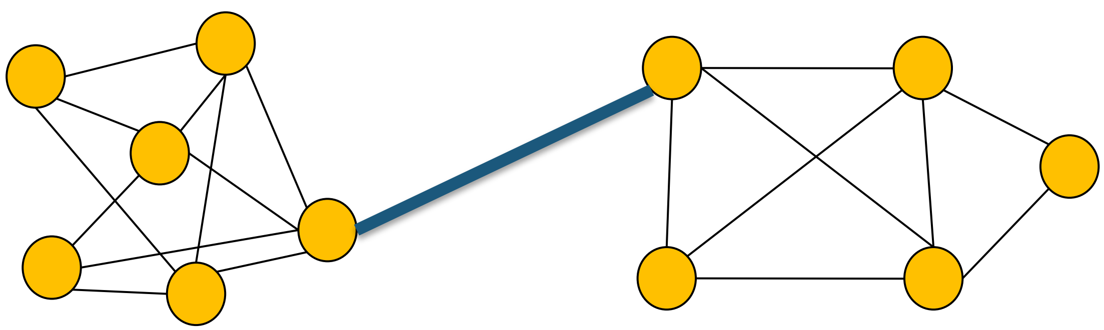
Applications of modularity

- 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



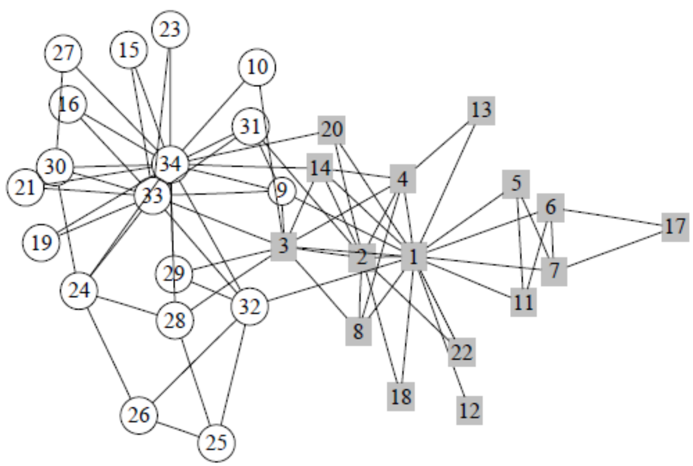
Edge betweenness is higher for edges that connect different communities

Newman-Girvan algorithm (1)

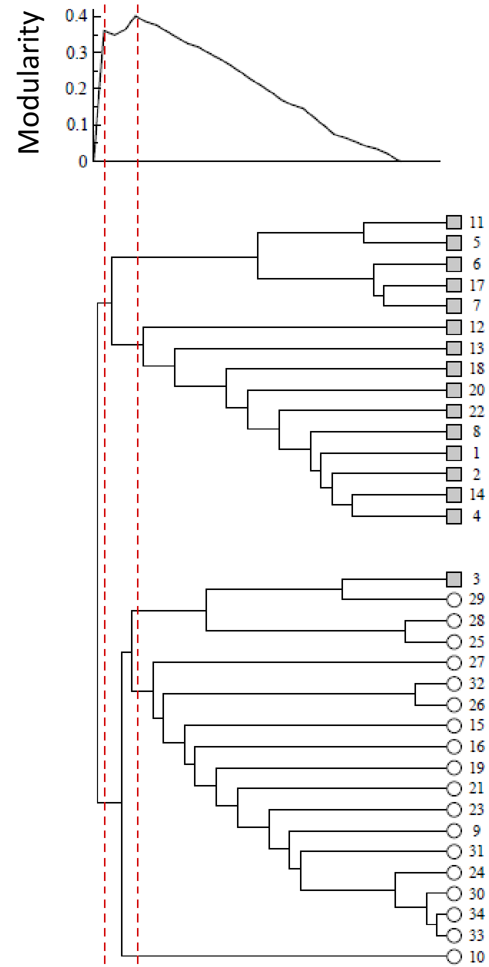
- **Basic steps:**
  1. Compute betweenness centrality for all edges in the graph
  2. Find and remove the edge with the highest score
  3. Recalculate betweenness centrality score for the remaining edges
  4. Go to step 2
- How do we know if the produced communities are **good ones** and stop the algorithm?
  - 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]

Greedy techniques (1)

- Newman's algorithm [Newman '04b]
  - Agglomerative (bottom-up) hierarchical clustering algorithm
  - **Idea:** Repeatedly join pairs of communities that achieve the greatest increase of modularity (dendrogram representation)

  1. Initially, each node of the graph belongs on its own cluster (**n**)
  2. Repeatedly, join communities in pairs by adding edges
     a. At each step, choose the pairs that achieve the **greatest increase** (or minimum decrease) of modularity
     b. Consider only pairs of communities **between which there exist edges** (merging communities that do not share edges, it can never improve modularity)

  - Complexity: **$O((m+n)\ n)$** (or **$O(n^2)$** on a sparse graph)

Greedy techniques (2)

- Can we improve the complexity of Newman's algorithm?
  - Greedy optimization algorithm by Clauset, Newman and Moore **[Clauset et al. '04]**
  - **Key point:** large graphs are **sparse**
  - Exploit sparsity by using appropriate **data structures** for sparse graphs (e.g., max-heaps)
    a. A sparse matrix for storing the variations of modularity $\Delta Q_{i,j}$ after joining two communities **i**, **j** (in the case they are connected by an edge)
    b. A max-heap data structure for the largest element of each row of matrix $\Delta Q_{i,j}$ (fast update time and constant time for finndmax() operation)
  - Complexity: **O(m d logn), d** is the depth of the dendrogram describing the performed partitions (the community structure)
    - Sparse graphs: **m ~ n**. Graphs with hierarchical structure: **d ~ logn**. Therefore, the complexity is $O(n \log^2 n)$ for such graphs

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**

$$
\begin{aligned}
Q &= \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(c_i, c_j) \\
&= \frac{1}{4m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)(s_i s_j + 1) \\
&= \frac{1}{4m}\sum_{ij} B_{ij} s_i s_j = \frac{1}{4m} s^T \mathrm{B} s
\end{aligned}
$$

- **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 **u**

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

- Modularity can now expressed as

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

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 **u_1** of **B** corresponding to the largest eigenvalue
  2. Assign the nodes of the graph in one of the two communities **X** (**s_i = +1**) and **Y** (**s_i = -1**) based on the **signs** of the corresponding components of the eigenvector

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

- 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 logn)** for sparse graphs

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

Simulated annealing

- **Simulated annealing** is a probabilistic method for global optimization of a given function in a large search space
  - Explore the search space looking for a good approximation of the global optimum of a function **f** (modularity in our case - maximum)
  - Set of states, correspond to points of the search space
  - Transitions from one state to another are achieved probabilistically
    1. With probability 1, if **f** increases after moving to the other state
    2. With probability **exp(β Δf)**, where **Δf** represents the amount of decrease of **f** when moving to the other state, and **β** is a parameter that helps to avoid getting trapped in local optima
- Modularity optimization using simulated annealing **[Guimera et al. '04]**
  - Two types of movements-transitions
  1. Individual node movements, from one community to another (randomly)
  2. Collective node movements, either by merging two communitie, or splitting one community
  - Mostly for small graphs (~ 10^4 nodes)

[Fortunato '10]

Extremal optimization

- Optimization heuristic search method
- **Basic idea:** optimize a global function, by optimizing local variables **[Duch and Arenas '05]**
  - **Global function:** modularity **Q**
  - **Local variables:** the contribution of individual nodes to the modularity $\mathbf{q_i}$
- The modularity in the graph can be expressed as the sum over the nodes, based on their contribution: $Q = \frac{1}{2m}\sum_i q_i$

1. Start from a random partition of the graph into two parts
2. At every iteration, the node with **the lower value of local variable** is moved to the other partition, until the global modularity is not changing
3. Delete all links between both partitions
4. Repeat recursively at every part of the remaining graph

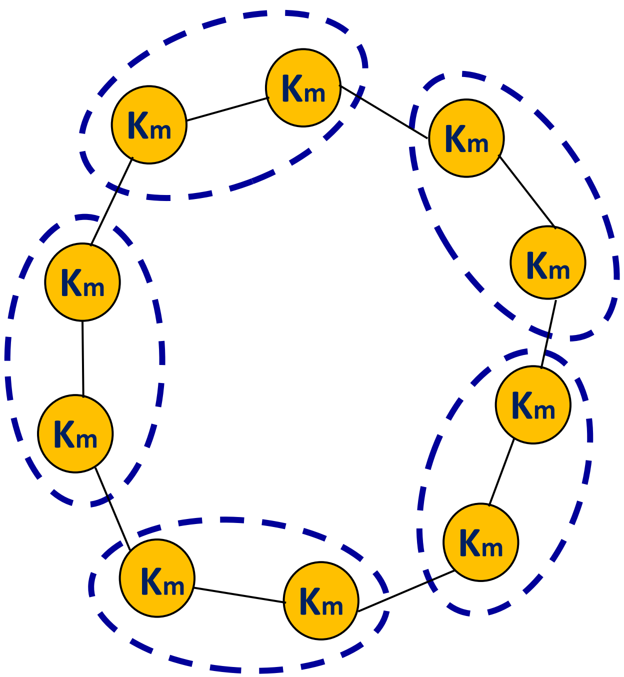
- Complexity: $\mathbf{O(n^2 \log n)}$

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 ≤ 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)

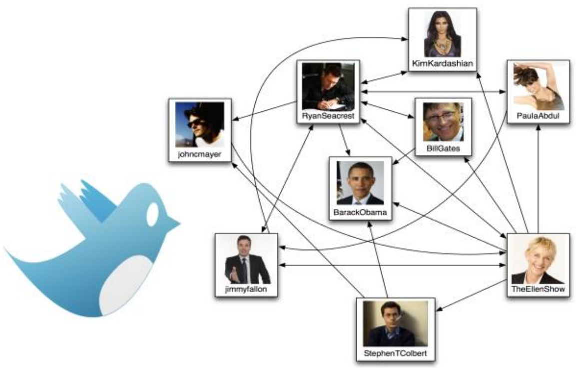
- M.E.J. Newman. Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E 74, 2006.
- R. Guimera, M. Sales-Pardo, L.A.N. Amaral. Modularity from Fluctuations in Random Graphs and Complex Networks. Phys. Rev. E 70, 2004.
- J. Duch and A. Arenas. Community detection in complex 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

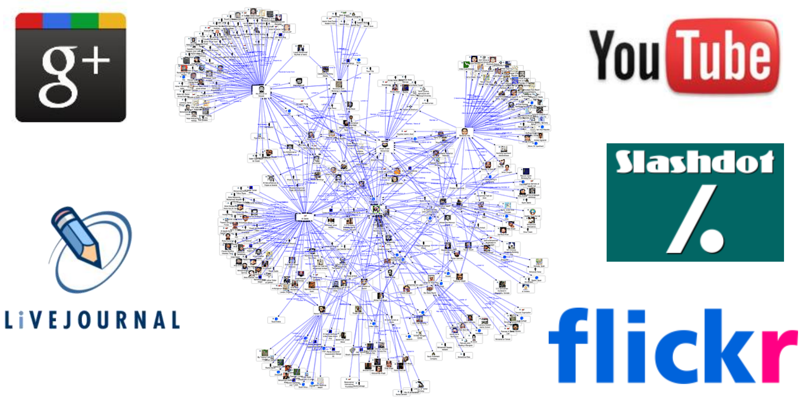
1. Introduction & Motivation
2. Graph fundamentals
3. Community evaluation measures
4. Graph clustering algorithms
5. **Clustering and community detection in directed graphs**
6. Alternative Methods for Community Evaluation
7. New directions for research in the area of graph mining

Directed graphs – why should we care (1)?

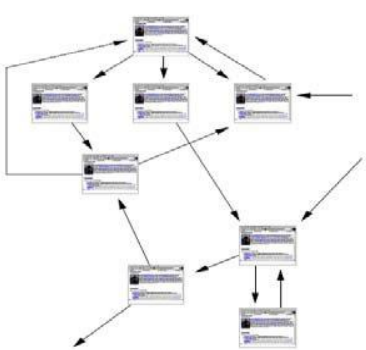
- A plethora of network data from several applications is from their nature **directed**



Twitter
[Image: http://sites.davidson.edu/mathmovement/]

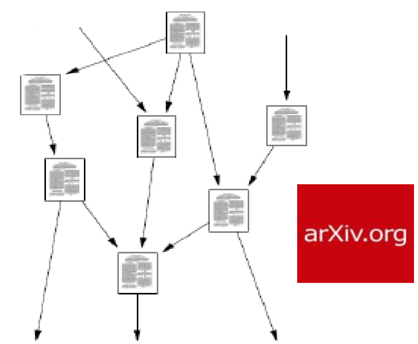


Online Social Networks



Web Graph

Wikipedia



Citation Graph

Directed graphs – why should we care (2)?

- **Social and Information Networks:**
  - Communities in the **directed hyperlink structure of the Web** correspond to sets of web pages that possibly share common topics
  - Communities in SNs with **non-symmetric links** (e.g., Twitter) → individuals with common interests or friendship relationships
- **Biology:** In prokaryote genome sequence data, the donor-recipient relations between genomes are modeled by **directed graphs** (Lateral Gene Transfer - LTG)
  - Community detection enables to **test hypotheses** relevant to LTG patterns and mechanisms operating in nature
- **Neuroscience:** Neuron interactions are represented by **directed graphs**
  - Community detection methods help us to comprehend the **functional architecture** of the brain

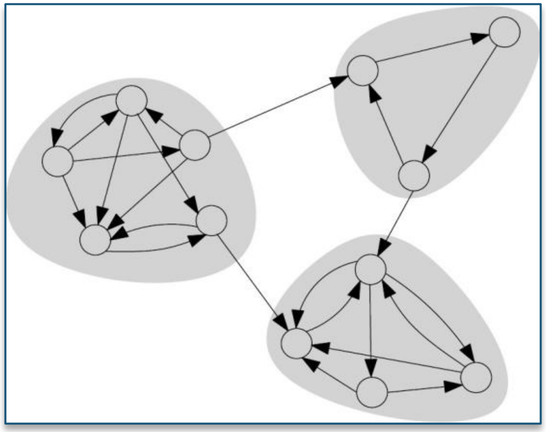
Directed graphs – why should we care (3)?

- **Clustering non-graph data:**
  - Apply graph clustering algorithms on data with no inherent graph structure (e.g., points in a d-dimensional Euclidean space)
- **How?**
  1. Construct a **similarity graph based** on the topological relationships and distance between data points
  2. Then, the problem of clustering the set of data points is transformed to a graph clustering problem

Depending on the way the similarity graph is constructed, the final graph can be **directed** (e.g., using **k**-Nearest Neighbor graphs) **[von Luxburg '07]**

Basics

- Similar to the undirected case, the community detection is the task of grouping the vertices of a **directed network** into clusters (communities), in such a way that
  - There should be many edges within each cluster ...
  - ... and relatively few edges among different clusters



- **However**, the problem has mainly been considered and studied for the case of **undirected** networks
- A large number of diverse algorithms have been proposed
- **[Fortunato, Phys. Reports '10]**

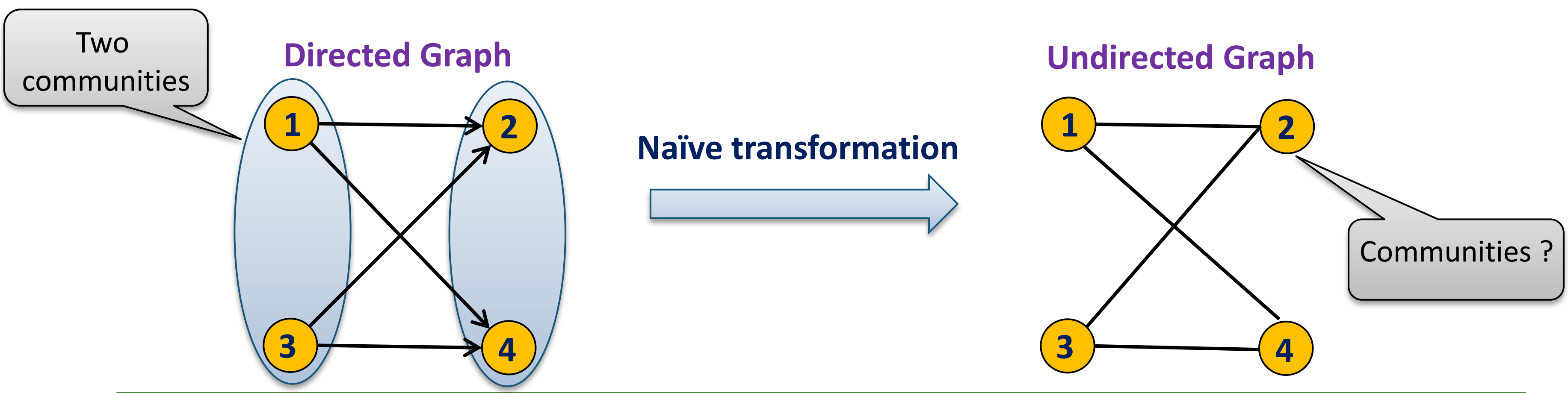
Edge directionality should be considered properly in the community detection task

Challenges in clustering directed graphs (1)

- The problem is generally a more **hard and challenging** task compared to the undirected one
- Existence of **asymmetric relationships** among entities (non reciprocal) → the nature of interactions are fundamentally different from the one in the undirected case
- **Graph concepts** for community evaluation (e.g., density)
  - Well theoretically founded for undirected graphs
  - Not enough effort has been put on how to extend these concepts on directed graphs
- **Theoretical tools**
  - Mainly **graph theoretic and linear algebraic tools**
  - Have mainly been considered for undirected graphs
  - Not straightforward extension to the directed case

Challenges in clustering directed graphs (2)

- No precise and common **definition** for the problem
- The presence of directed links is possible to imply the existence of other **more sophisticated** types of clusters that
  - Do not exist in undirected networks
  - Can not be captured using only density and edge concentration characteristics
- **Ignoring directionality** and naively transform the graph to undirected in not a good practice



Topics on clustering in directed graphs

- Notions - Intuitive definitions
  - **Density-based** communities
  - **Pattern-based** communities
- Approaches for identifying communities in directed networks
  - Naïve graph transformation
  - Transformations maintaining directionality
  - Extending clustering objective functions and methodologies to directed networks
  - Alternative approaches

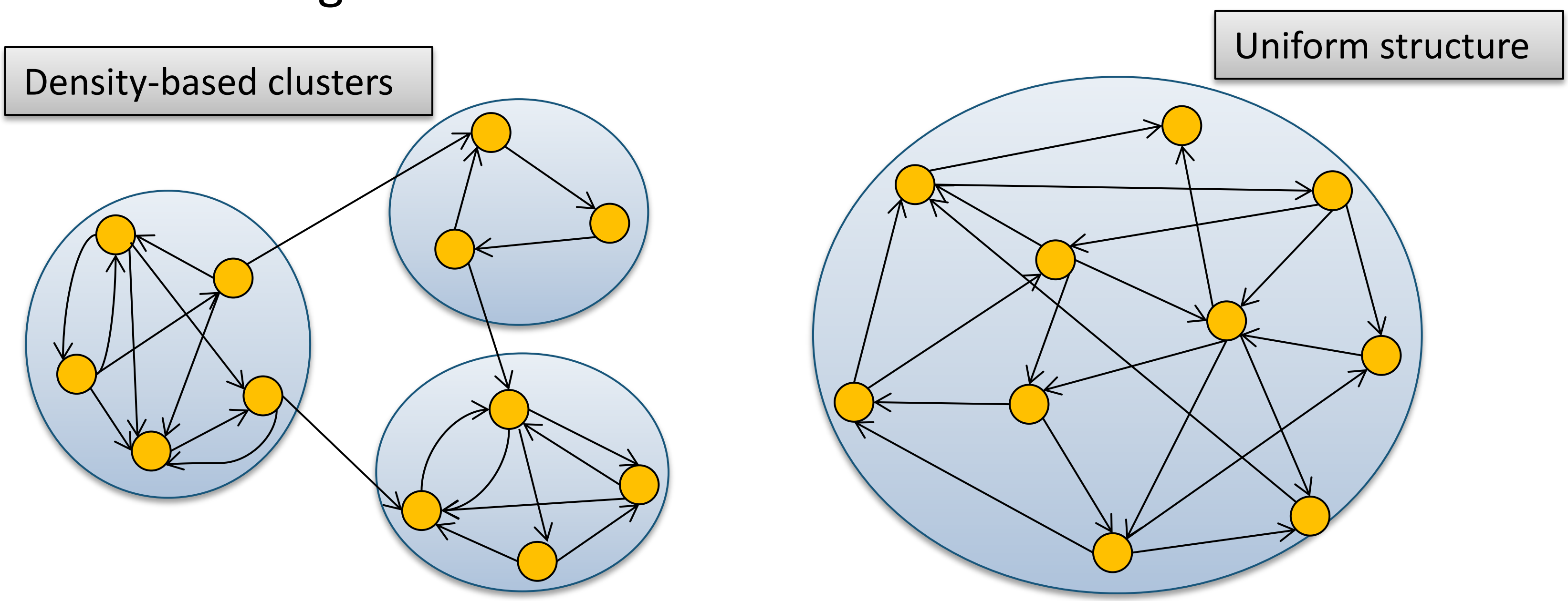
Communities in directed networks

> A cluster or community in a graph can be considered as a set of nodes that share common or similar features (characteristics)

- Two main definitions/notions (or categories) of clusters in directed networks
  - Density-based clusters
  - Pattern-based clusters

Density-based clusters

- Follow the typical clustering definition based on edge density characteristics
- Entirely based on the **distribution-density** of the edges inside the network
- Group of nodes with more intra-cluster edges than inter-cluster edges



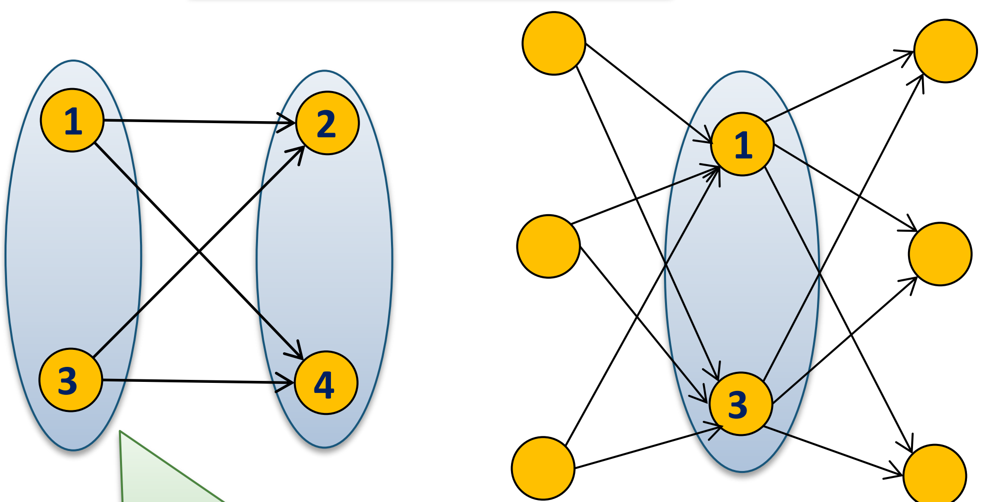
Is it a “trivial” task?

- Extending the notion of density-based clusters to directed networks is not always a trivial procedure
- Meaningful extension of the **objective criteria** (e.g., modularity) used for community evaluation
- Simple graph concepts become more complex
  - E.g., each cluster should be **connected** [Schaeffer ‘07]
  - Three types of connectivity in directed graphs
    - **Weak connectivity**
    - **Connectivity**
    - **Strong connectivity**

Pattern-based clusters

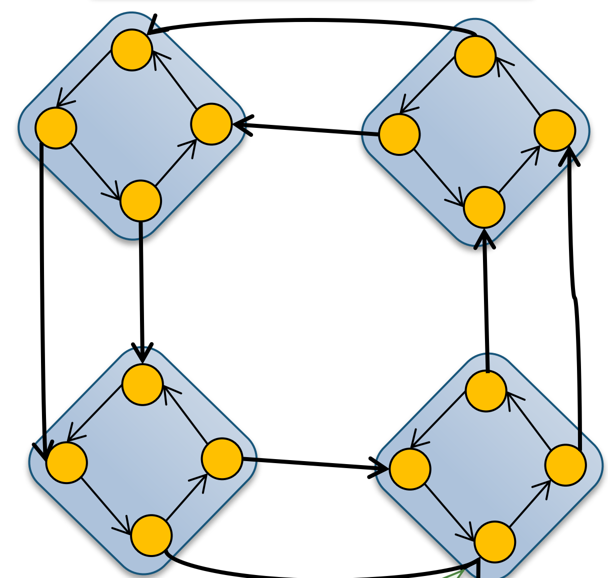
- The density-based definition cannot capture more **sophisticated** clustering and connectivity patterns
  - Edge density alone may not represent the major clustering criterion
  - Patterns beyond edge density

Citation-based clusters



Nodes point to and pointed by the same group

Flow-based cluster



Patterns of flow – groups with persistent flow

Remarks on clustering notions

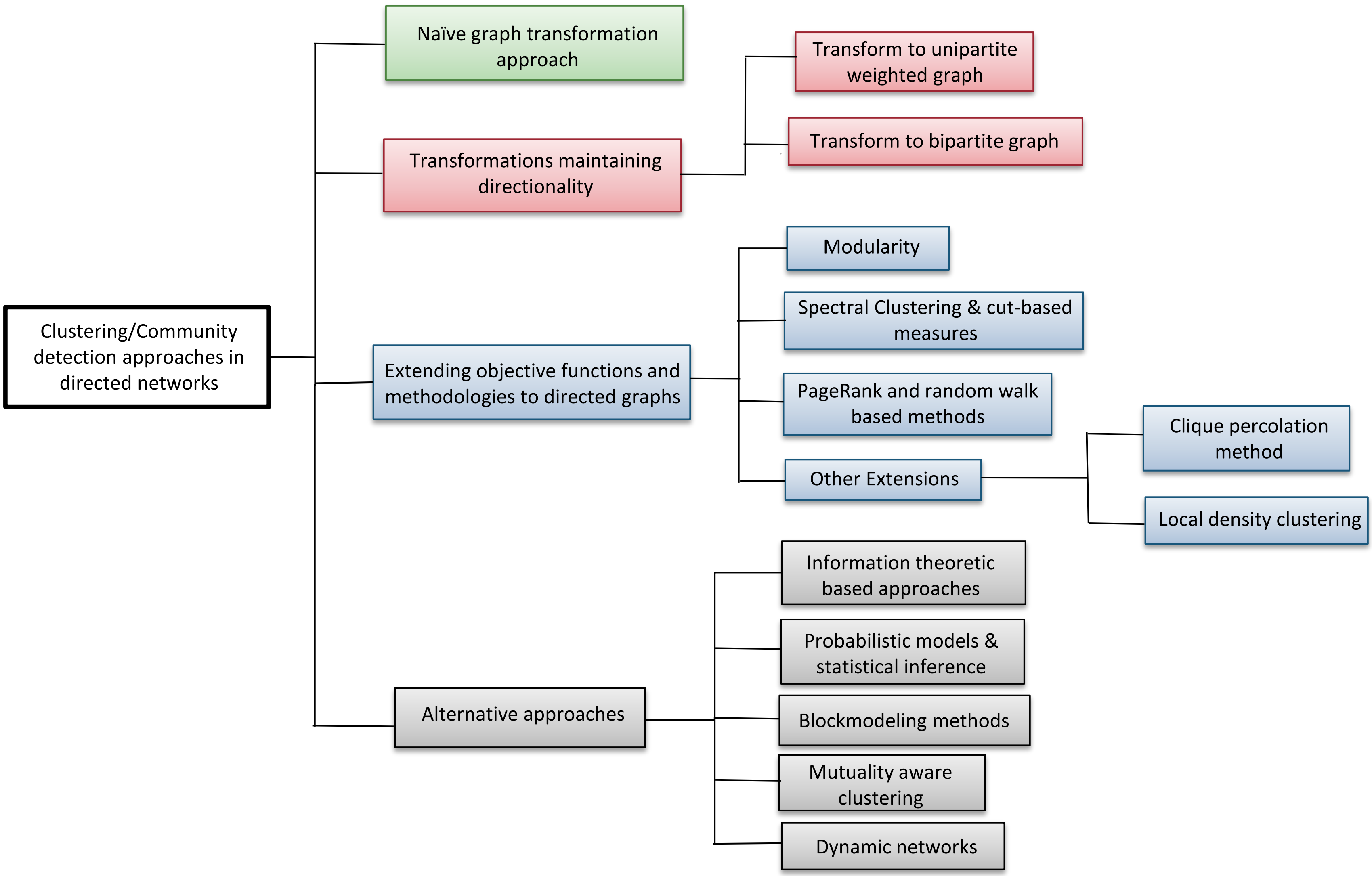
- Both types of clusters may co-exist in a directed graph
  - **Combined** density-based and pattern-based clusters
- E.g., many methods adopt the citation-based clustering notion **and** are also able to identify density-based clusters
- **Key point:**
  - Apply appropriate transformations to **enhance** a density-based method with pattern-based clustering features

Community detection in directed graphs

Approaches for identifying communities in directed networks w.r.t. the undirected case of the problem

- Naïve graph transformation
- Transformations maintaining directionality
- Extending clustering objective functions and methodologies to directed networks
- Alternative approaches

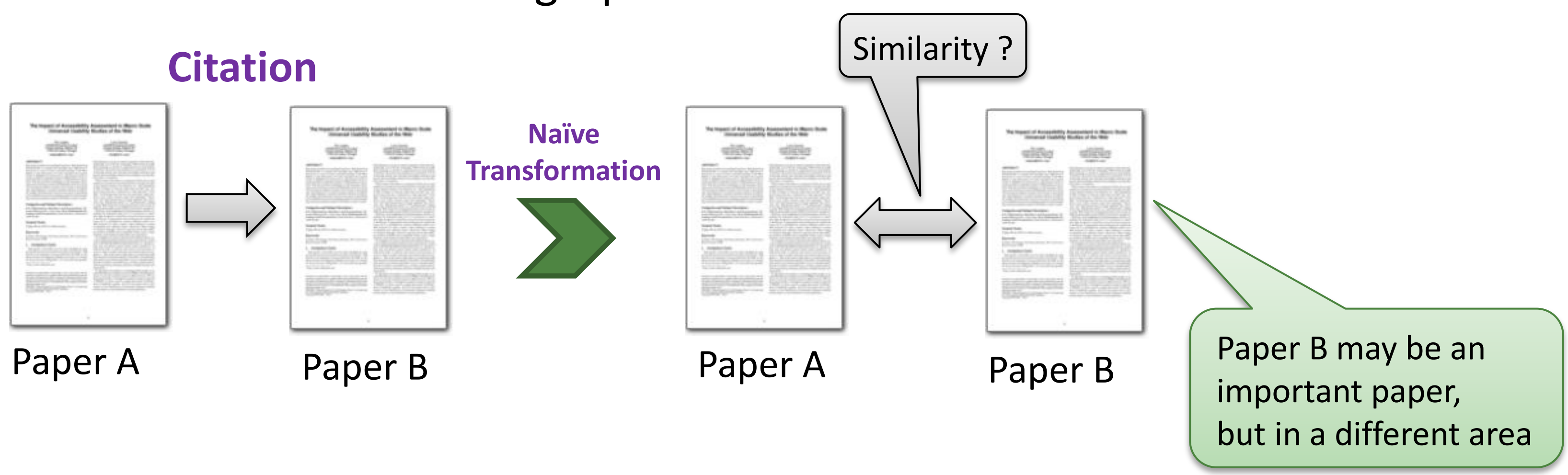
Taxonomy



Clustering/Community detection approaches in directed networks
Naïve graph transformation approach
Transformations maintaining directionality
Transform to unipartite weighted graph
Transform to bipartite graph
Extending objective functions and methodologies to directed graphs
Modularity
Spectral Clustering & cut-based measures
PageRank and random walk based methods
Other Extensions
Clique percolation method
Local density clustering
Alternative approaches
Information theoretic based approaches
Probabilistic models & statistical inference
Blockmodeling methods
Mutuality aware clustering
Dynamic networks

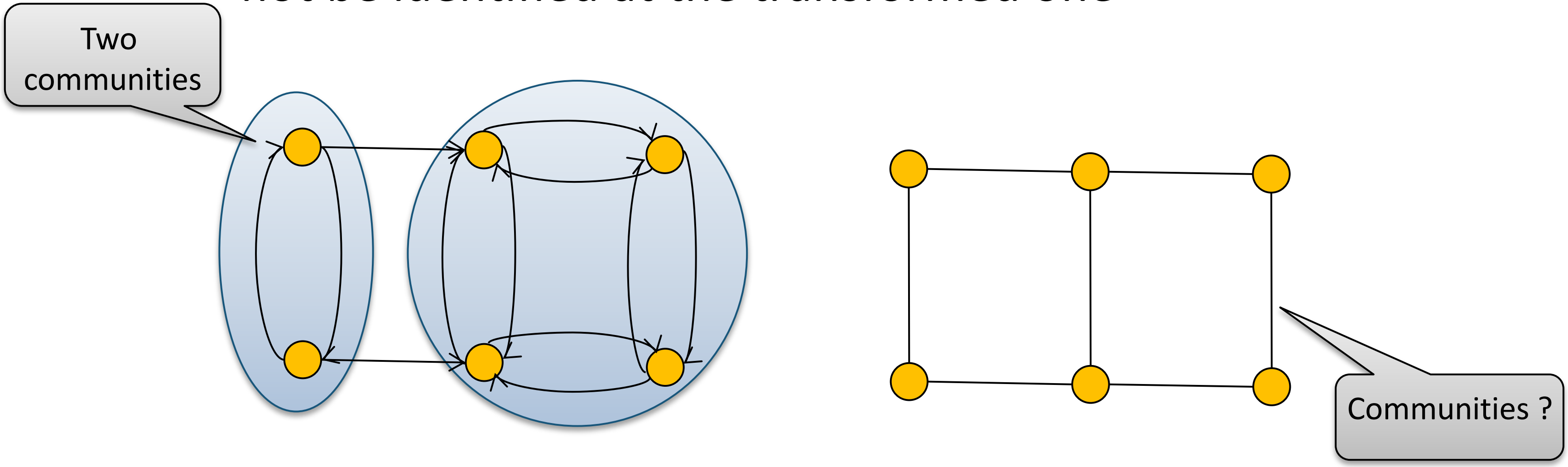
Naïve graph transformation approach (1)

- Discard edge directionality and treat graphs as undirected → Apply algorithms for undirected graphs
- **Several drawbacks:** information represented by edges' direction is ignored
- **Data ambiguities**
  - Ambiguities and to some degree incorrect information are introduced in the graph

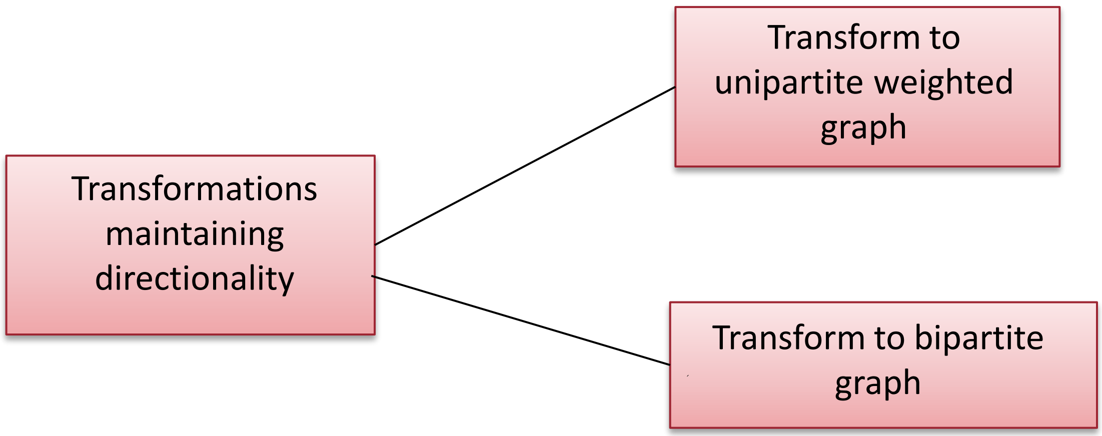


Naïve graph transformation approach (2)

- **Deviations in clustering results**
  - Ambiguities introduced in the data, may have impact to the final outcome of the clustering algorithm
  - Valuable information is not utilized in the clustering process
  - E.g., clusters that exist in the initial directed network, may not be identified at the transformed one

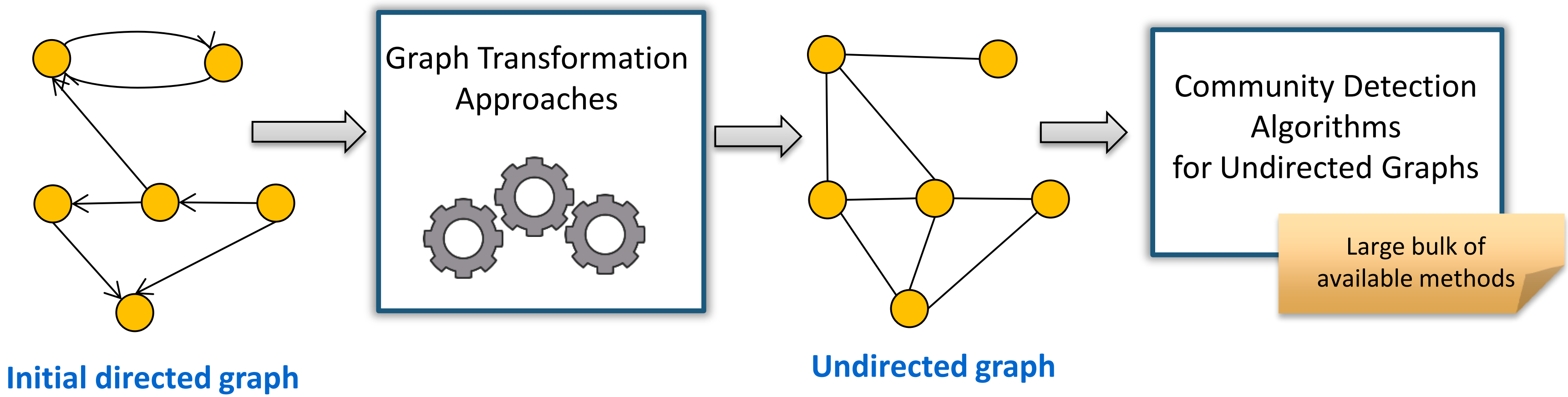


Transformations maintaining directionality



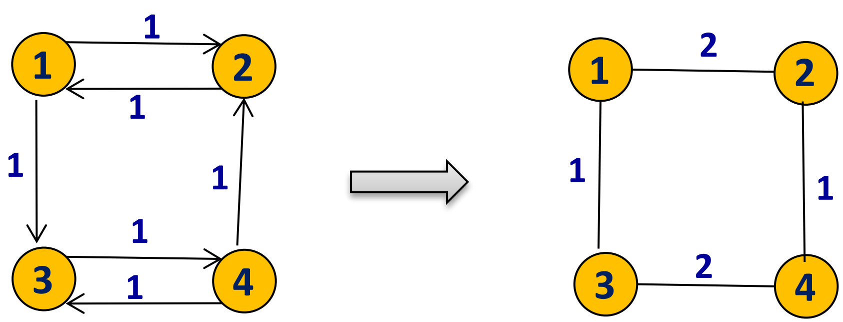
Transformations maintaining directionality

1. Transform the directed graph to undirected (unipartite / bipartite)
2. Edges' direction information is retained as much as possible (e.g., by introducing weights on the edges of the transformed graph)
3. Apply already proposed community detection algorithms designed for undirected graphs
4. The extracted communities will also correspond to the communities of the initial graph



Transformation to unipartite weighted graph (1)

- **Idea:** transform the directed graph to undirected
  - Information about directionality is incorporated via edge weights
- Graph symmetrizations [Satuluri and Parthasarathy '11]
- $\mathbf{A}_U = \mathbf{A} + \mathbf{A}^T$ symmetrization



Directed graph (adj. matrix: **A**) Transformed graph

- Same number of edges
- Edges in both direction:
  - Add as **edge weight** the **sum** of the weights in the initial graph

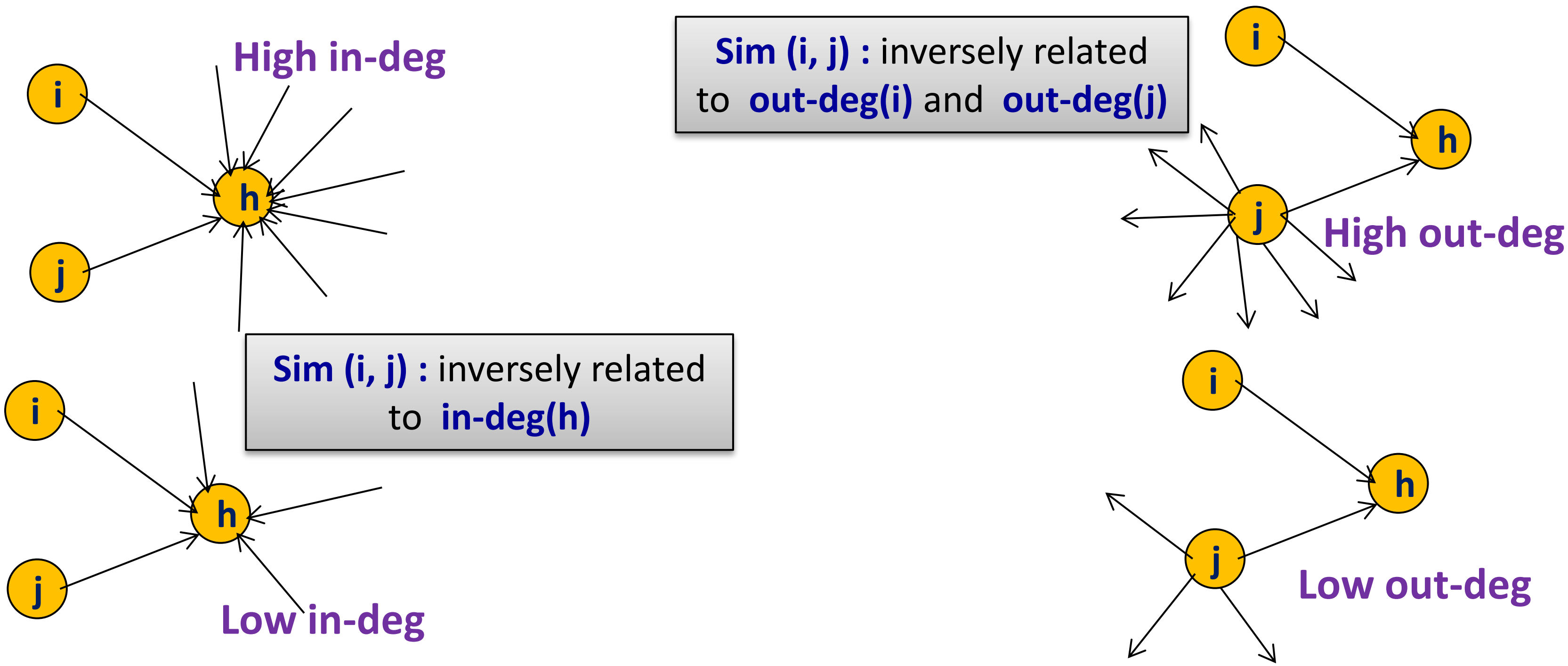
Transformation to unipartite weighted graph (2)

- The previous symmetrization maintains intact the edge set (discard directions – new edge weights)
- **Observation:** Meaningful clusters can be groups of nodes that share similar incoming and/or outgoing edges
  - Edges should appear between similar nodes (**in-link** and **out-link** node similarity)
- Bibliometric symmetrization **[Satuluri and Parthasarathy '11]**
- $\mathbf{A}_U = \mathbf{A}\mathbf{A}^T + \mathbf{A}^T\mathbf{A}$
  - $B = \mathbf{A}\mathbf{A}^T$: **Bibliographic coupling matrix** (captures the number of common outgoing edges between each pair of nodes)
  - $C = \mathbf{A}^T\mathbf{A}$: **Co-citation strength matrix** (captures the number of common incoming edges between each pair of nodes)

> - Introduce **new edges** based on
>   - Number of common outgoing edges and incoming edges

Transformation to unipartite weighted graph (3)

- The degree distribution of real-world networks is **heavy-tailed**
- Nodes with high degree would share a lot of common edges with other nodes (higher similarity)
- How can we define a **similarity measure** between the nodes of a directed graph, taking into account in- and out- degree?



Transformation to unipartite weighted graph (4)

- **Degree discounted** symmetrization

$$B = D_{out}^{-a} A D_{in}^{-\beta} A^{T} D_{out}^{-a}$$

Bibliographic coupling matrix

$$C = D_{in}^{-\beta} A^{T} D_{out}^{-a} A\, D_{in}^{-\beta}$$

Co-citation matrix

- Adjacency matrix of symmetrized undirected graph
  - $A_U = B + C$
- Typically, α = β = 0.5 **[Satuluri and Parthasarathy '11]**

Transformation to unipartite weighted graph (5)

- **Random-walk based** transformation **[Satuluri and Parthasarathy '11], [Lai et al., Physica A '10], [Lai et al., J. Stat Mech. '10]**
- The normalized cut criterion will be preserved
- Two neighborhood nodes are more probable to belong on the same community, if they can be mutually visited by random walks starting from these nodes
  - Use edge directionality to classify edges
- The edge between those nodes is more likely to be **an intra-community** edges → It will receive higher weight than **inter-community edges** **[Lai et al., J. Stat Mech. '10]**

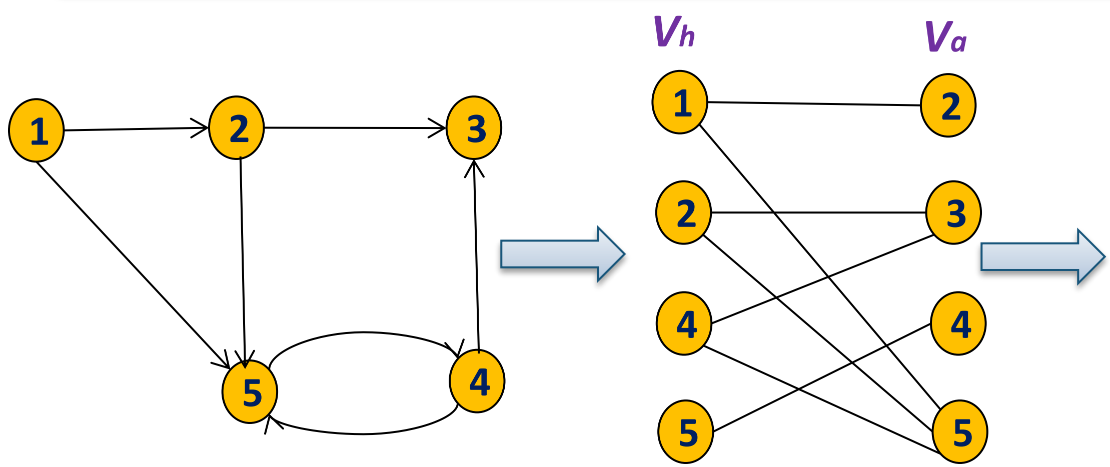
Transformation to bipartite graph (1)

- The directed graph $G = (V, E)$ is transformed to a bipartite undirected one $G_B = (V_h, V_a, E_b)$:

$$V_h = \{i_h \mid i \in V \text{ and } D_{out}(i) > 0\}$$

$$V_a = \{i_a \mid i \in V \text{ and } D_{in}(i) > 0\}$$

Each directed edge $(i, j) \in E$ between two nodes of the directed graph G will be represented by an edge $(i_h, j_a) \in E_b$ of the produced bipartite graph

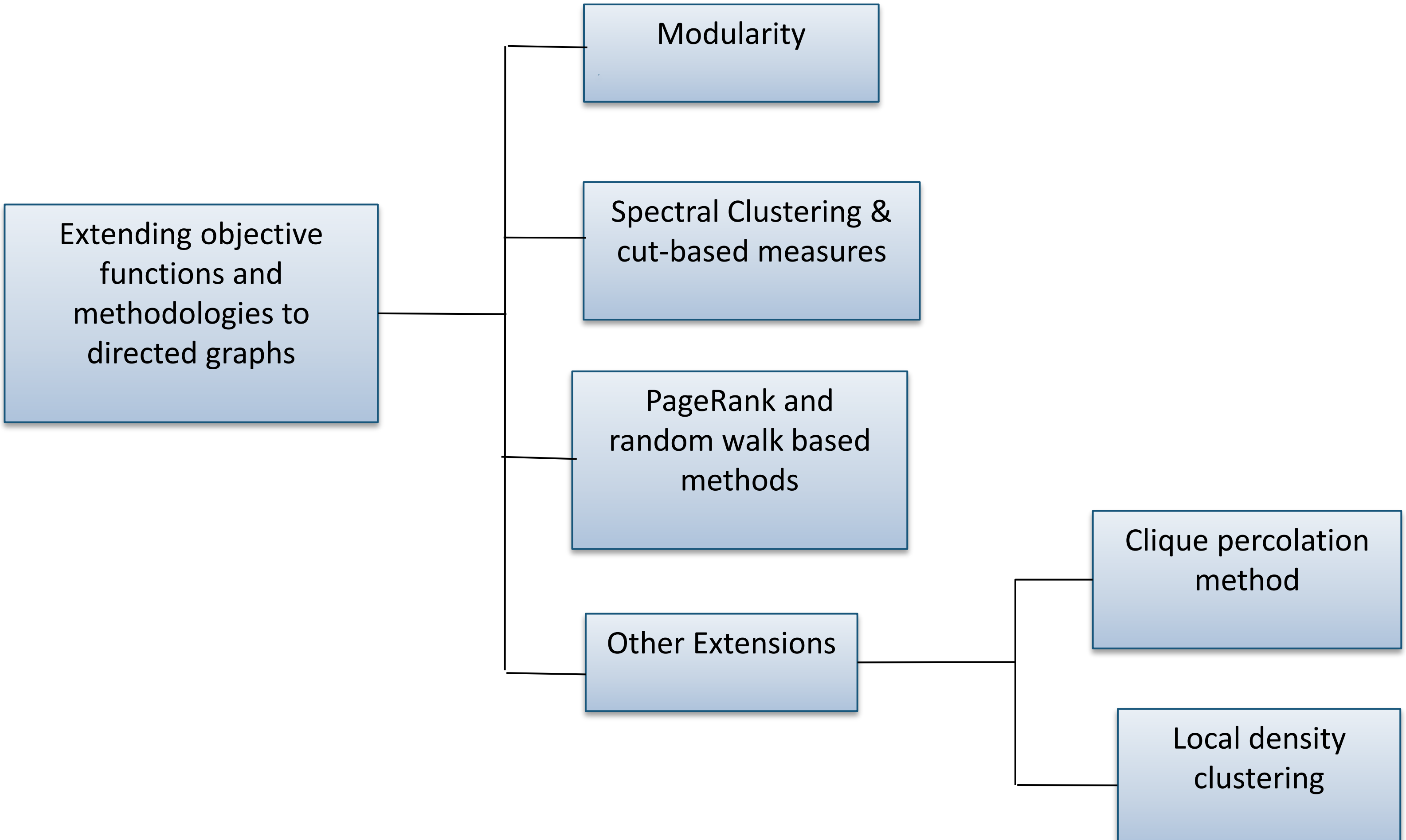


Apply community detection methods for bipartite graphs

Transformation to bipartite graph (2)

- Approach inspired by Kleinberg's **hub** and **authority** web model
- **Idea:** detect clusters of nodes with similar outgoing and incoming links
- Consider that the partitions represent actors (**h**) and teams (**a**)
  - Identify groups of actors that are closely connected to each other through **co-participation** in many teams **[Guimera et al. '07], [Zhan et al. '11]**
- Other approach: semi-supervised learning framework for directed graphs **[Zhou et al. '05]**
  - Node classification in directed graphs (positive or negative labels)
  - Absence of labeled node instances → graph clustering tool
  - **Idea:** category similarity of co-linked nodes
  - Node similarity based on the existence of common parents and common children structures → highlight co-linked nodes structures

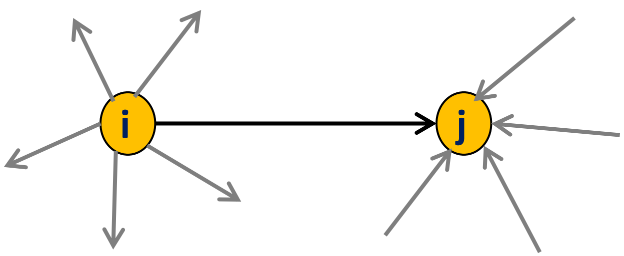
Extending objective functions and methodologies



Modularity for directed graphs

- Initially introduced for the case of undirected graphs
- Q = (fraction of edges within communities) - (expected fraction of edges)
- In directed graphs, the existence of a directed edge ***(i, j)*** between nodes ***i*** and ***j*** depends on the **out-degree** of ***i*** and **in-degree** of ***j***

$$Q_d = \frac{1}{m}\sum_{i,j}\left[\mathbf{A}_{ij} - \frac{k_i^{out}k_j^{in}}{m}\right]\delta(c_i, c_j)$$



- Consider that:
  - *i* has high out-deg and low in-deg
  - *j* has high in-deg and low out-deg
- More probable to observe edge ***(i, j)*** than edge ***(j, i)***

[Arenas et al. '07], [Leicht and Newman '08]

Modularity optimization

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

$$Q_d = \frac{1}{m}\sum_{ij}\left(A_{ij} - \frac{k_i^{out}k_j^{in}}{m}\right)\delta(c_i, c_j)$$
$$= \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i^{out}k_j^{in}}{m}\right)(s_i s_j + 1)$$
$$= \frac{1}{2m}\sum_{ij} B_{ij} s_i s_j = \frac{1}{2m} s^T \mathrm{B} s$$

$$B_{ij} = A_{ij} - \frac{k_i^{out}\, k_j^{in}}{m}$$

Modularity matrix (not symmetric)

[Leicht and Newman '08]

- Transpose Q_d (scalar) and take the average

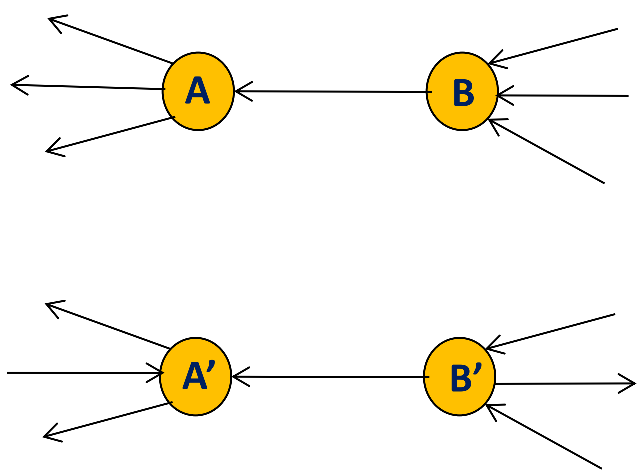
$$Q_d = \frac{1}{4m} s^T (B + B^T) s$$

- Now, $B + B^T$ is symmetric

- **Spectral optimization** of modularity
- Compute the eigenvector that corresponds to the largest positive eigenvalue of $B + B^T$
- Assign the nodes to communities **X** and **Y** according to the signs of the corresponding components in the eigenvector
- Repeated bisection (for more than two communities)

A few interesting points of directed modularity

- Modularity exhibits two limitations
  - It cannot properly distinguish the directionality of the edges
  - It cannot be used to detect clusters representing patterns of movement between nodes



- Nodes **A** and **A'** as well as **B** and **B'** have the same in-deg and out-deg respectively
- **B→ A** : more precise (strong) directed flow that the one **B' → A'**
- Modularity cannot distinguish these different situations

[Kim et al. '10]

Other extensions of directed modularity

- Random walk based formulation
  - LinkRank method **[Kim et al. '10]**
  - Indicates the importance of the edges in the graph based on random walk concepts
  - $Q_{linkrank}$ = (fraction of time spent by a random surfer while walking within communities) – (expected value of this time)
  - A community is a group of nodes where a random surfer is more likely to stay
  - It can distinguish properly the direction of the edges

- Directed modularity for overlapping communities
  - Allow nodes to be assigned in more than one community
  - Extend the configuration model **[Nicosia et al. '09]**

Spectral clustering and cut-based methods

- **Spectral clustering:** partition the nodes of the graph using information related to the spectrum of a matrix representation of the graph (e.g., Laplacian or adjacency)
- Optimizing cut-based objective measures, can be achieved using spectral techniques
- We can say that spectral methods have a dual use:
  - Clustering framework itself
  - Optimization framework of objective functions
  - Close connection between those two points
- Laplacian matrix for directed graphs
  - Spectral clustering algorithm
- Extension of cut-based measures to directed graphs

Laplacian matrix for directed graphs

- **Undirected networks:** use the eigenvector that corresponds to the second smallest non-zero eigenvalue of the Laplacian matrix (Fiedler vector) to obtain a bipartition of the nodes
  - Solution to the **normalized cut** objective function
- What about directed directed graphs?
- **Laplacian matrix** for directed graphs

$$L_d = I - \frac{\Pi^{1/2} P \Pi^{-1/2} + \Pi^{-1/2} P^T \Pi^{1/2}}{2}$$

- ***P*** is the transition matrix and $\Pi = \mathrm{diag}(\pi_1, \pi_2, \ldots, \pi_n)$ the stationary distribution of the random walk
  - Cheeger inequality holds for L_d
- **[Chung '07], [Zhou et al. '05], [Li and Zhang '10]**

Directed spectral clustering algorithm

Input: Directed graph $G = (V, E)$

Output: A partition of the vertex set V into two parts

1. Define a random walk over G with transition matrix **P**
2. Form the normalized Laplacian matrix L_d
3. Compute the eigenvector u_2 of L_d that corresponds to the second smallest (non zero) eigenvalue
4. Partition the vertex set V into two parts
   a. $S = \{i \in V \mid u_2(i) \geq 0\}$
   b. $S' = \{i \in V \mid u_2(i) < 0\}$

- The algorithm can be extended in the case of a k-partition
  - Eigenvectors of the k smallest eigenvalues of L_d
  - **[Zhou et al. '10], [Gleich '06]**

Cut-based measures

- The Laplacian matrix provide a solution to the normalized cut problem
- What about other cut-based measures?
- **Weighted cuts** **[Meila and Pentney '07]**

$$\mathrm{WCut}(S,\bar{S}) = \frac{\sum_{i\in S, j\in \bar{S}} T'_i A_{ij}}{\sum_{i\in S} T_i} + \frac{\sum_{j\in \bar{S}, i\in S} T'_j A_{ji}}{\sum_{j\in \bar{S}} T_j}$$

  - Balanced size node clusters (vector ***T***)
  - Vector ***T'*** is used as a normalization factor
  - The optimization of **WCut** can be relaxed to a symmetric problem
- Other generalization of **NCut** (in image processing) **[Yu and Shi '01]**

PageRank and random walk based methods

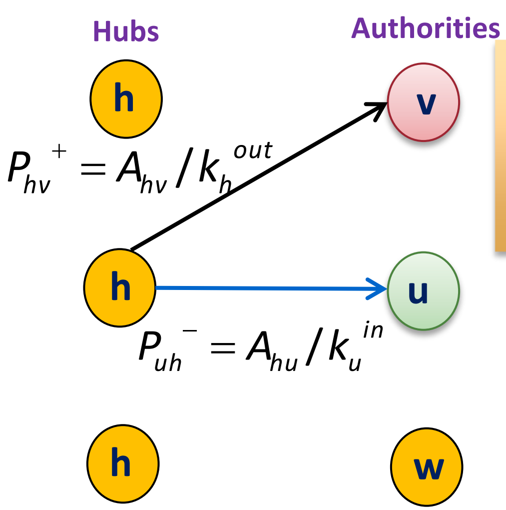
- Random walks are closely related to spectral clustering
  - Cut-based measures can be expressed in terms of random walks

$$\mathrm{NCut}(S, \bar{S}) = \frac{\Pr(S \to \bar{S})}{\Pr(S)} + \frac{\Pr(\bar{S} \to S)}{\Pr(\bar{S})}$$

- The minimization of the number of edges that crossing a cut can be described as a similar process where the random walker is forced to stay more time within a cluster
- **Other random walk based approach:**
  - Consider the transition matrix ***P*** of a random walk
  - Look for piecewise constant components in the top k eigenvectors of ***P*** **[Pentney and Meila '07]**
  - Look for correlation between components in the eigenvectors **[Capocci et al. '05]**
    - The components correspond to nodes of the same cluster will show high correlation

Community detection in the Web graph (1)

- The random walk should ensure that Web pages that share a common topic or interest should be grouped together
  - Even if they are not directly connected
  - Co-citation and co-reference information (pattern-based clusters)
- Use a **two-step** PageRank random walk treating nodes as hubs/authorities **[Huang et al. '06]**



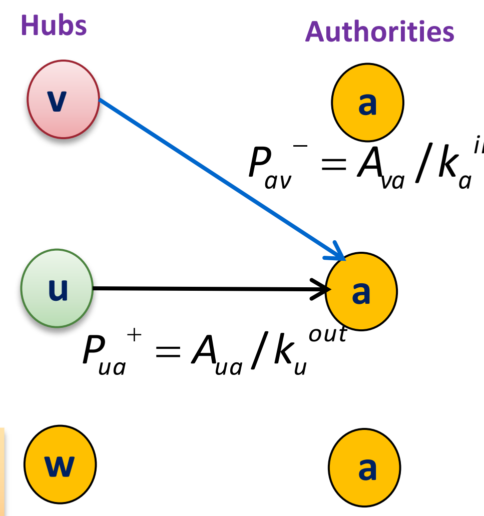
Jump one step backward to **h** and then one step forward to **v**

Two-step transition probability between authorities **u** and **v**

$$P_{uv}^{A} = \sum_h P_{uh}^{-} P_{hu}^{+}$$

Two-step transition probability between hubs **u** and **v**

$$P_{uv}^{H} = \sum_a P_{ua}^{+} P_{av}^{-}$$



Jump one step forward to **a** and then one step backwardt o **v**

Community detection in the Web graph (2)

- The transition matrix of the random walk can be defined as

$$P = \beta\, P^{A} + (1-\beta)\, P^{H}$$

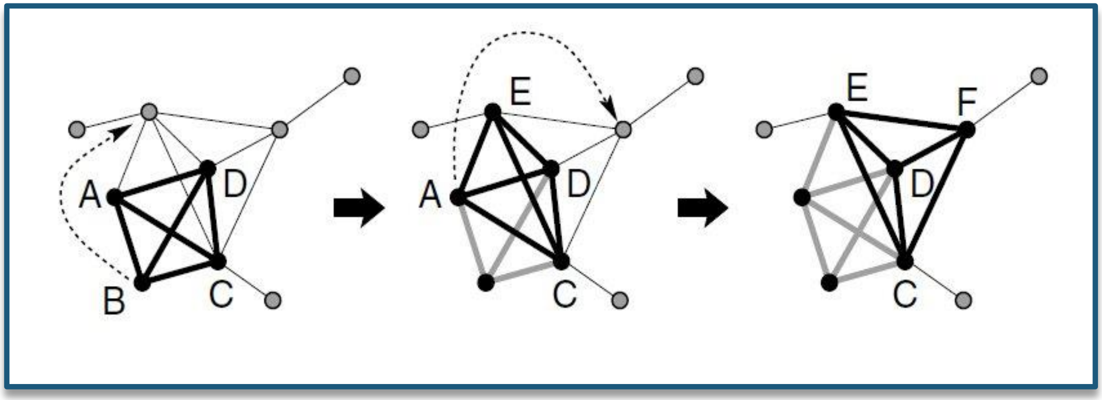
- It combines both backward and forward two step random walks
  - Co-citation and co-reference node similarity
- Parameter **β** controls the co-citation and co-reference effects
- Apply the modified transition matrix to the Laplacian matrix and use spectral methods to extract the communities

Local clustering using random walks

- **Goal:** find a good local clustering structure near a specified seed node
- Examine only a small portion of the input graph
- **Idea:** combine information from local and global structure **[Andersen et al. '07]**
  - **Local:** Personalized PageRank score of a node ***v*** (seed node)
  - **Global:** PageRank score of node ***v***
- For the seed node ***v***
  - Compute the Personalized PageRank score with a single starting node (seed node)
  - Compute the global PageRank score with a uniform starting distribution over all nodes
  - Take the ratio of the entries in the Personalized PR and global PR and sort the nodes according to the ratio

Other extensions to directed graphs (1)

- Clique percolation method **[Palla et al. '07]**
  - Detect network modules (dense connected groups of nodes)
  - **Idea:** consider the definition of **k**-cliques (complete subgraph with k nodes)
  - **Adjacent k-cliques:** they share **k-1** nodes
  - **Module:** the union of k-cliques that can be reached from each other traversing the nodes of adjacent **k**-cliques



- Template **k**-clique: A-B-C-D
- The template is gradually rolled to adjacent k-cliques
- Final module: A-B-C-D-E-F

  - **Directed k-cliques:** complete subgraphs of size **k**, where the nodes can be ordered, i.e., directed edges connect higher order node to lower ones

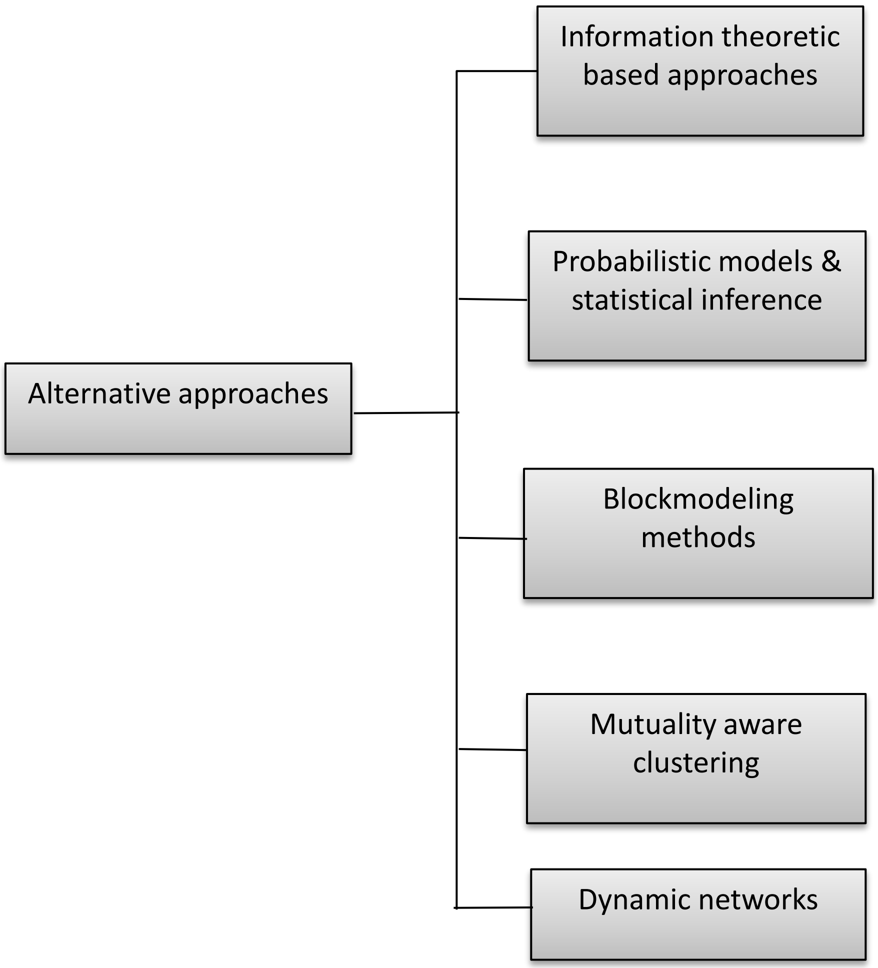
Other extensions to directed graphs

- Local density clustering [Schaeffer '05], [Virtanen '03]
  - Extend the concept of local cluster density to directed graphs
  - Find a good local cluster that contains a specific seed node
  - **Internal degree** of cluster **int-deg(C)**: # of edges with both endpoints in **C**
  - **External degree** of cluster **ext-deg(C)**: # of edges with only the start node in **C**
  - **Density** of graph **G = (V, E)**: $\delta = |E| / |V|(|V|-1)$
  - **Local density** of cluster **C**: $\delta_{local}(C) = \text{int-deg}(C) / |C|(|C|-1)$
  - **Relative density** of **C**: $\delta_r(C) = \text{int-deg}(C) / (\text{int-deg}(C) + \text{ext-deg}(C))$
  - A cluster should have both high local and relative density
  - Cluster quality measure: $f(C) = \delta_{local}(C) \times \delta_r(C)$

Local clustering: find subgraph **C** with **k** nodes that contains a given node **v** (seed node), maximizing **f(C)**

- Optimization using a local search approach starting from **v**

Alternative approaches



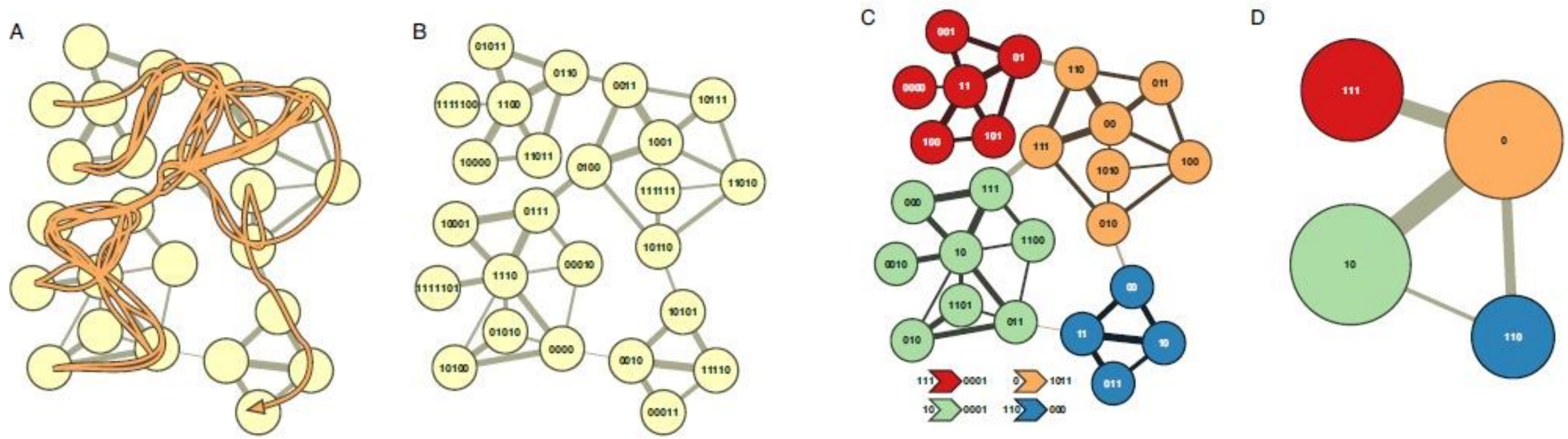
Information-theoretic based approaches (1)

- Communities in graphs represent patterns and regularities
  - The can be used to efficiently compress the data
- **Isomap** method **[Rosvall and Bergstrom '08]**
  - Combine **random walks** and **compression principles**
  - **Intuition:** communities can be identified based on how fast information flows on them
  - Apply the concept of random walks to describe the process of inf. flow
  - We have seen that a community corresponds to a group of nodes where the random surfer is more likely to be trapped in
    - The random surfer will visit more time nodes of the same group than nodes outside of that
  - **Idea:** communities would correspond to groups of nodes in which the random walk can be compressed better

> Reformulation as a **coding** problem:
>
> Select a partition **M** of **|V|** nodes into **c** communities, minimizing the description length of the random walk

Information-theoretic based approaches (2)

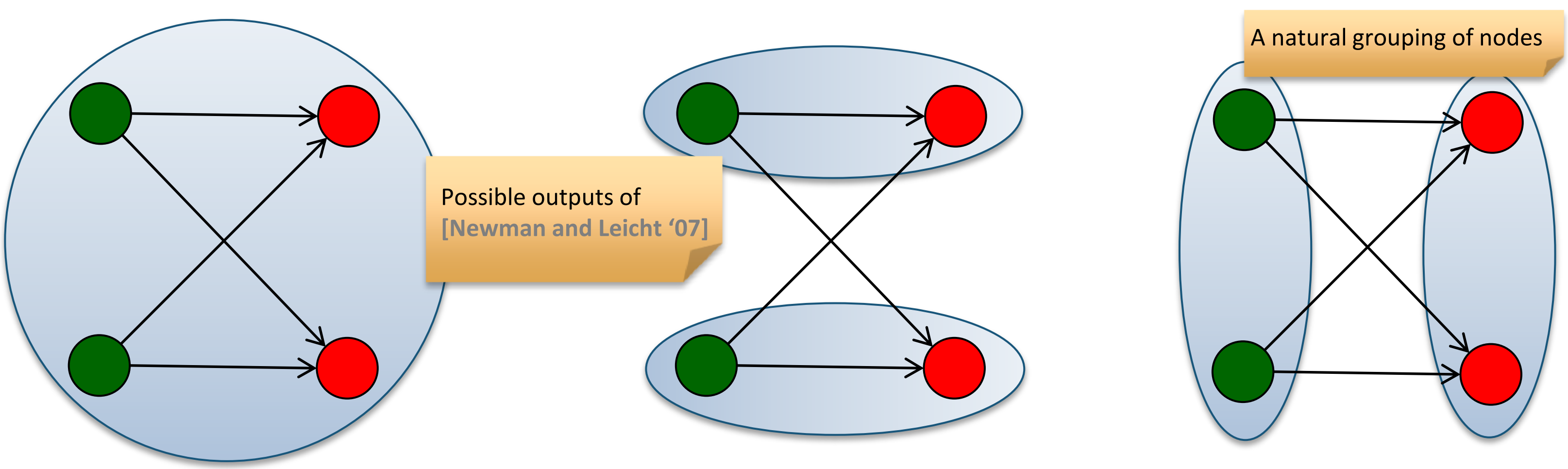
Illustration of **Isomap** [Rosvall and Bergstrom '08]



A. Trajectory of the random walks
B. Use Huffman coding to assign codewords to the nodes based on the trajectory
   - Shorter codewords are assigned to more frequently visited nodes
C. Two level description:
   - Unique codewords (names) for major clusters
   - Codewords of nodes within clusters are reused
D. Coarse grained description: report only the codewords of clusters (high level description)

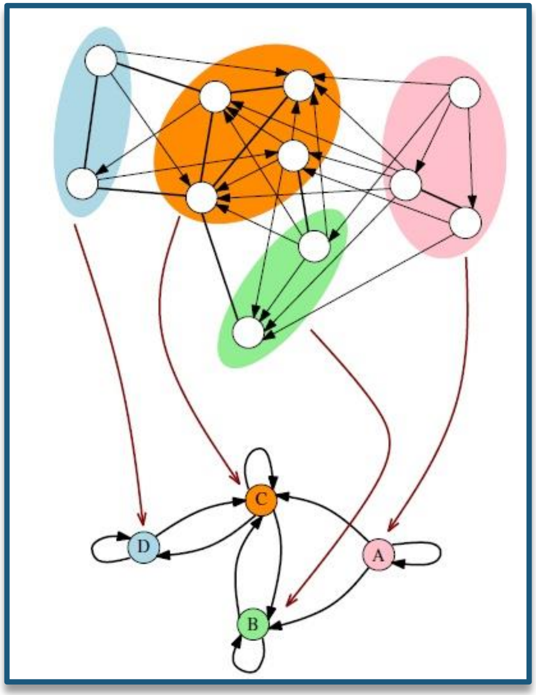
Probabilistic models and statistical inference

- **Mixture models** for inferring the group (community) membership of nodes in a directed network
- Formulate the community detection problem as a likelihood maximization problem **[Newman and Leicht '07]**
  - Apply an Expectation-Maximization algorithm to infer the probabilities q_{ir} , i.e., the probability that node **i** belongs to community **r**
  - **Note:** Each community should have at least one node with non-zero out degree (due to the formulation of the mixture model)
- Extensions **[Ramasco and Mungan '08], [Wang and Lai '08]**



Blockmodeling methods

- **Blockmodeling:** represent a large and possibly incoherent graph by a smaller structure that can be interpreted more easily

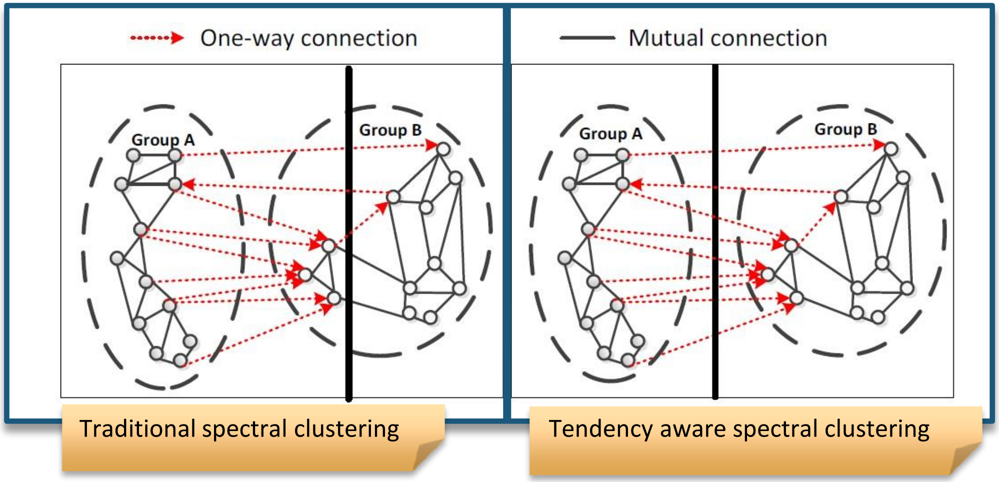


[Batagelj and Mrvar '02]

- Similar to a co-clustering procedure
  - Reordering scheme of the adjacency matrix
  - Formation of a block-wise structure
- The blockmodel for the example graph can be described by matrix B_{cxc}
  - $B_{gq} = 1$ if there exists an edge between communities **g** and **q**
- [Holland et al. '83], [Wang and Wong '87], [Yang et al. '10], [Airoldi et al. '08], [Rohe and Yu '12]

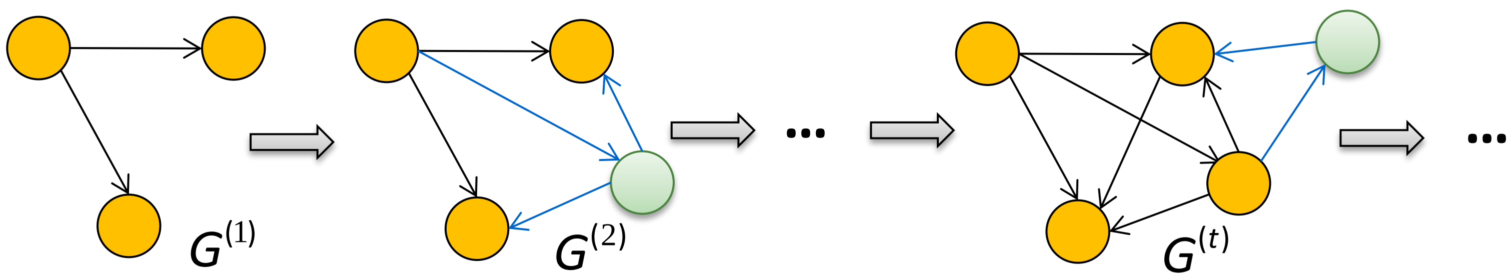
Mutuality-tendency aware community detection

- Existence of **mutual** (both-way) and **one-way** connections in directed networks
  - Most approaches do not explicitly distinguish them
  - By minimizing the number of inter-community edges, possible tendencies between nodes are not captured
  - **Importance:** Cluster stability depends on the existence of mutual connections
- Tendency aware spectral clustering **[Li et al. '12]**
  - Tendencies of node pairs to form reciprocal connections
  - **Criterion:** Maximization of **intra-cluster mutuality tendency** and minimization of **the inter-cluster mutuality tendency**



Dynamic networks

- Dynamic nature of real-world networks
- **Graph stream** (sequence of graphs) $G := \{ G^{(1)}, G^{(2)}, \ldots, G^{(t)}, \ldots \}$
- Incrementally find communities in dynamic graphs



- Two sub-problems need to be addressed
  - **Community discovery:** node assignment into communities of static snapshots
  - **Change point detection:** quantify and detect the change of the community structure over time – similarity between different partitions over time
    - Significant change in the already identified community structure
  - **[Sun et al. '07], [Duan et al. '09]**

References (directed graphs)

- U. von Luxburg. A tutorial on spectral clustering. Statistics and Computing 17(4), 2007.
- S.E. Schaeffer. Graph clustering. Computer Science Review 1(1), 2007.
- V. Satuluri and S. Parthasarathy. Symmetrizations for clustering directed graphs. In: EDBT, 2011.
- D. Lai, H. Lu, and C. Nardini. Finding communities in directed networks by pagerank random walk induced network embedding. Physica A 389, 2010.
- D. Lai, H. Lu, and C. Nardini. Extracting weights from edge directions to find communities in directed networks. J. Stat. Mech., 2010.
- R. Guimera, M.S. Pardo, and L.A. Nunes Amaral. Module identification in bipartite and directed networks. Physical Review E 76(3), 2007.
- W. Zhan, Z. Zhang, J. Guan, and S. Zhou. Evolutionary method for finding communities in bipartite networks. Physical Review E 83(6), 2011.
- D. Zhou, B. Scholkopf, T. Hofmann. Semi-supervised learning on directed graphs. In: NIPS, 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.

References (directed graphs)

- Y. Kim, S.-W. Son, and H. Jeong. Finding communities in directed networks. Phys. Rev. E 81, 2010.
- 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.
- F. Chung. Laplacians and the cheeger inequality for directed graphs. Annals of Combinatorics 9, 2005.
- D. Zhou, J. Huang, and B. Scholkopf. Learning from labeled and unlabeled data on a directed graph. In: ICML, 2005.
- Y. Li and Z.-L. Zhang. Random walks on digraphs, the generalized Laplacian and the degree of asymmetry. In: WAW, 2010.
- D. Gleich. Hierarchical directed spectral graph partitioning. TR, Stanford U, 2006.
- M. Meila and W. Pentney. Clustering by weighted cuts in directed graphs. In: SDM, 2007.
- S.X. Yu and J. Shi. Grouping with directed relationships. In: EMMCVPR, 2001.
- W. Pentney and M. Meila. Spectral clustering of biological sequence data. In: AAAI, 2005.
- A. Capocci, V.D.P. Servedio, G. Galdarelli, and F. Colaiori. Detecting communities in large networks. Physica A 352(2-4), 2005.

References (directed graphs)

- J. Huang, T. Zhu, and D. Schuurmans. Web communities identification from random walks. In: PKDD, 2006.
- R. Andersen, F. Chung, and K. Lang. Local partitioning for directed graphs using pagerank. In: WAW, 2007.
- G. Palla, I.J. Farkas, P. Pollner, I. Derenyi, and T. Vicsek. Directed network modules. New Journal of Physics 9(6), 2007.
- S.E. Schaeffer. Stochastic local clustering for massive graphs. In: PAKDD, 2005.
- S. Virtanen. Clustering the chilean web. In: LA-WEB, 2003.
- M. Rosvall and C.T. Bergstrom. Maps of random walks on complex networks reveal community structure. PNAS, 2008.
- M.E.J. Newman and E.A. Leicht. Mixture models for exploratory analysis in networks. PNAS 104(23), 2007.
- J.J. Ramasco and M. Mungan. Inversion method for content-based networks. Physical Review E 77(3), 2008.
- J. Wang and C.H. Lai. Detecting groups of similar components in complex networks. New J. Phys. 10(12), 2008.
- V. Batagelj and A. Mrvar. Pajek – analysis and visualization of large networks. In: Graph Drawing, 2002.

References (directed graphs)

- P. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: Some first steps. Social Networks 5, 1983.
- Y. J. Wang and G. Y. Wong. Stochastic blockmodels for directed graphs. Journal of the American Statistical Association, 1987.
- T. Yang, Y. Chi, S. Zhu, and R. Jin. Directed network community detection: A popularity and productivity link model. In: SDM, 2010. F.D. Malliaros, V. Megalooikonomou, and C. Faloutsos. Fast robustness estimation in large social graphs: communities and anomaly detection. In: SDM, 2012.
- B.A. Prakash, A. Sridharan, M. Seshadri, S. Machiraju, and C. Faloutsos. Eigenspokes: surprising patterns and scalable community chipping in large graphs. In: PAKDD, 2010.
- E.M. Airoldi, D.M. Blei, S.E. Fienberg, and E.P. Xing. Mixed membership stochastic blockmodels. J. Mach. Learn. Res. 9, 2008.
- K. Rohe and B. Yu. Co-clustering for directed graphs; the stochastic co-blockmodel and a spectral algorithm. TR, 2012.
- Y. Li, Z.-L. Zhang, and J. Bao. Mutual or unrequited love: Identifying stable clusters in social networks with uni- and bi-directional links. In: WAW, 2012.
- J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In: KDD, 2007.

References (directed graphs)

- D. Duan, Y. Li, Y. Jin, and Z. Lu. Community mining on dynamic weighted directed graphs. In: CNIKM, 2009.
- A. Lancichinetti and S. Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs. Phys. Rev. E 80(1), 2009.
- J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. Internet Mathematics 6 (1), 2009.
- F. D. Malliaros, M. Vazirgiannis. Clustering and Community Detection in Directed Networks: A survey. To appear in Physics Reports Journal, Elsevier, 2013.

Outline

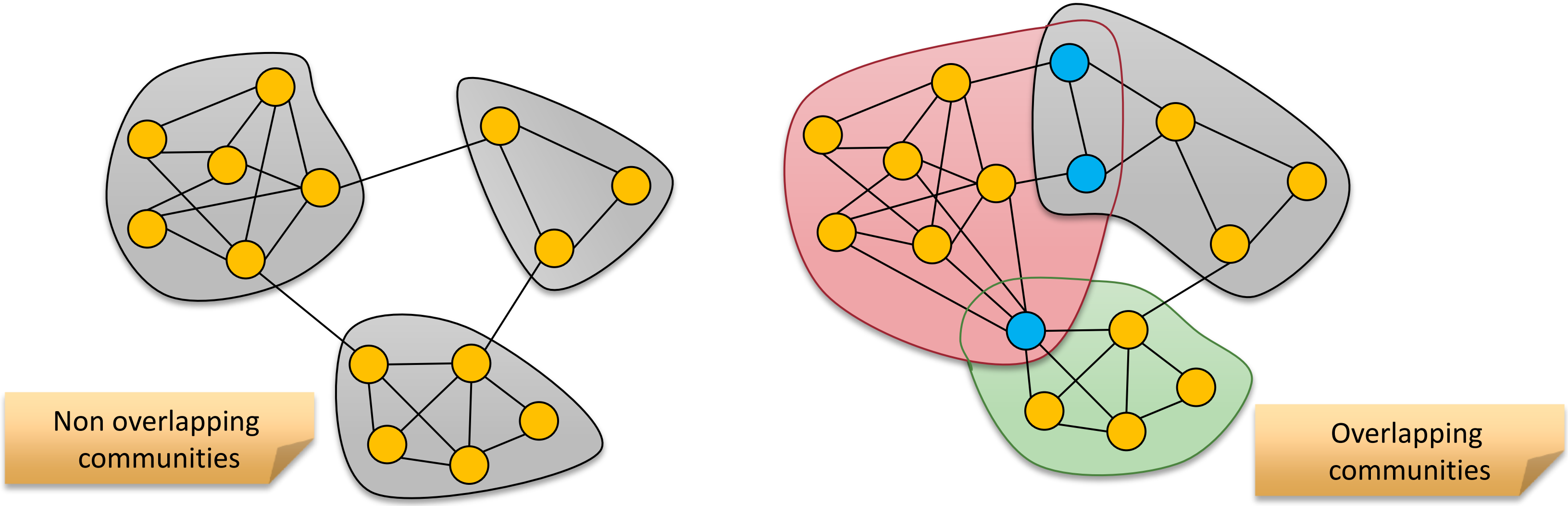
1. Introduction & Motivation
2. Graph fundamentals
3. Community evaluation measures
4. Graph clustering algorithms
5. Clustering and community detection in directed graphs
6. **Alternative Methods for Community Evaluation**
7. New directions for research in the area of graph mining

Topics on community detection and evaluation

- **Overlapping community detection**
- Global vs. local methods for community detection
- Community detection from seed nodes
- Observations on structural properties of large graphs
- Degeneracy-based community evaluation

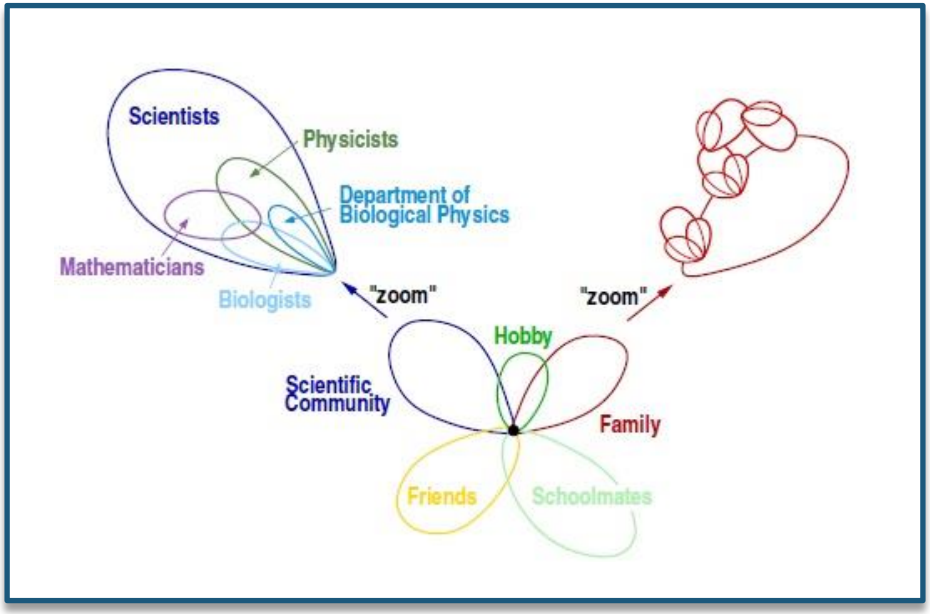
Overlapping community detection (1)

- Most of the methods presented so far perform **hard clustering**
  - The graph is divided into communities (clusters, modules)
  - Each node is assigned to a **single community**
- In many cases, nodes can simultaneously belong to more than one communities
  - **Overlapping communities**



Overlapping community detection (2)

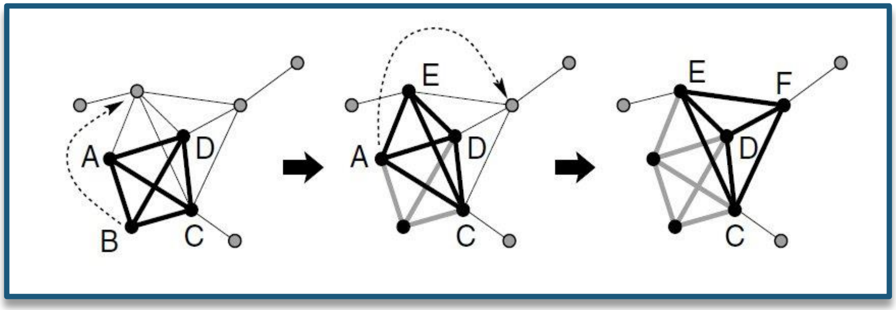
- **Why overlapping communities?**
  - E.g., in a social network, individuals have several simultaneous memberships (family, profession, friends, …)



[Palla et al. '05]

Overlapping community detection (3)

- Clique Percolation Method (CPM) **[Palla et al. '05]**
  - **Idea:** consider the definition of **k**-cliques (complete subgraph with k nodes)
  - **Adjacent k-cliques:** they share **k-1** nodes
  - **Communities:** the union of k-cliques that can be reached from each other traversing the nodes of adjacent **k**-cliques



- Template **k**-clique: A-B-C-D
- The template is gradually rolled to adjacent k-cliques
- Final module: A-B-C-D-E-F

- **CFinder** free software tool for CPM (http://www.cfinder.org/)

Overlapping community detection (4)

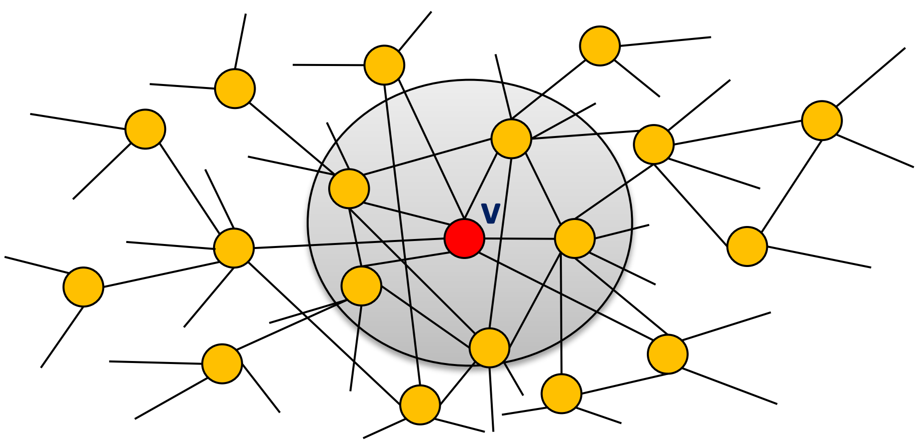
- Several extensions of Clique Percolation Method
  - Weighted graphs [Farkas et al. ’07]
  - Bipartite graphs (overlapping bicliques) [Lehmann et al. ’08]
  - Scalable (“fast”) implementation of CPM [Kumpula et al. ’08]
  - Parallel implementation of CFinder [Pollner et al. ’12]
- Drawbacks of CPM [Fortunato ’10]
  - Assumes that the graph has a large number of cliques
  - It may fail to detect communities in graphs with a small number of cliques
  - In case of graphs with many cliques → a single community that covers the whole graph
  - How to set parameter **k** to identify meaningful communities?

Topics on community detection and evaluation

- Overlapping community detection
- **Global vs. local methods for community detection**
- Community detection from seed nodes
- Observations on structural properties of large graphs
- Degeneracy-based community evaluation

Local vs. global communities

- Most of the proposed methods presented so far are **global**
  - Every node of the graph is finally assigned to a community
  - Complexity issues → we need to process the whole graph
- In many cases, we are interested in evaluating communities as **individual entities** **[Fortunato '10], [Schaeffer '07]**
  - Independent of the full graph
  - Using possibly limited amount of information → improving complexity
- Find local communities around **seed nodes**
  - For a given node **v**, extract the community that **v** belongs to



Topics on community detection and evaluation

- Overlapping community detection
- Global vs. local methods for community detection
- **Community detection from seed nodes**
- Observations on structural properties of large graphs
- Degeneracy-based community evaluation

Communities from seed nodes

- **Problem:** Given a seed node **v**, find a community around **v**
  - Based on a quality measure (e.g., conductance, expansion, cut ratio)
  - The desired size of the community may also be given as input
  - To problem can be extended to communities from a node set **s**

- **Question 1:** How to grow (expand) the seed set?
- **Question 2:** How to select the seed nodes?

Seed expanding strategies

Several strategies for expanding a seed set:

- Use random walks to expand a seed node set into a low-conductance community [Andersen and Lang '06]
  - Examine only a small neighborhood of the graph
- Local graph partitioning using **personalized PageRank** [Andersen et al. '06]
  - Find community around seed node **v**
  - Compute the personalized PageRank score (at the teleportation step, move to node **v**)
  - Sweep over the PageRank vector to find a good conductance set
  - Cheeger inequality for PageRank vectors (the basic tool in spectral clustering)
- Other extensions
  - Dense subgraphs around a seed node in bipartite graphs [Andersen '08]
  - Use of Markov chains (Evolving Set Process) [Andersen and Peres '09]

How to select seed nodes or sets? (1)

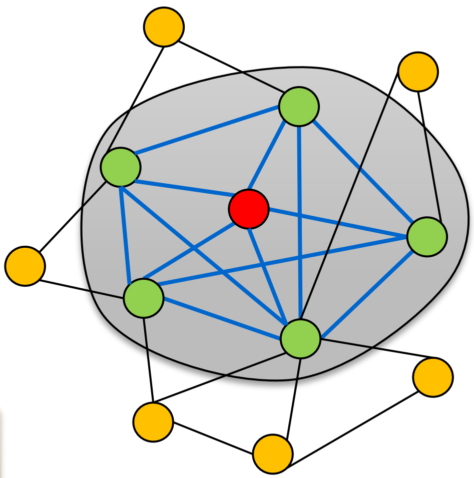
- It mainly depends on the scope of the community detection task
- Consider the node that we are interested in as a seed node
  - E.g., in a co-authorship graph
  - Find the best community of A.-L. Barabasi
- Important nodes
  - Based on centrality measures (e.g., degree, betweenness)
- Randomly
  - Just pick a random node and let the grow strategy to reveal the best cluster around this node

How to select seed nodes or sets? (2)

- In the case of seed sets, most of the methods require that the **seed set itself has good community properties**
  - E.g., low conductance node sets **[Andersen and Lang ’06], [Andersen et al. ’06]**
  - Difficult to find good seeds
- **Neighborhoods are good communities** **[Gleich and Seshadhri ’12]**
  1. High global clustering coefficient
  2. Heavy-tailed degree distribution

Node neighborhoods (egonets) with good conductance scores

Good seed sets



Sampling community structure

- **Idea** [Maiya and Berger-Wolf ’10]
  1. Produce subgraphs representative of community structure in the full graph
  2. Use these subgraph samples to infer the community membership for the rest of node in the graph
- **Sampling process** over the graph with respect to the community structure
  - The produced sample subgraphs should consist of members from most (or all) of the communities in the original graph
- How to sample representative subgraphs?
  - Based on the notion of **expander graphs**
  - Add nodes by **maximizing expansion** $f(S) = \frac{c_s}{n_s}$

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

Do we need all these methods?

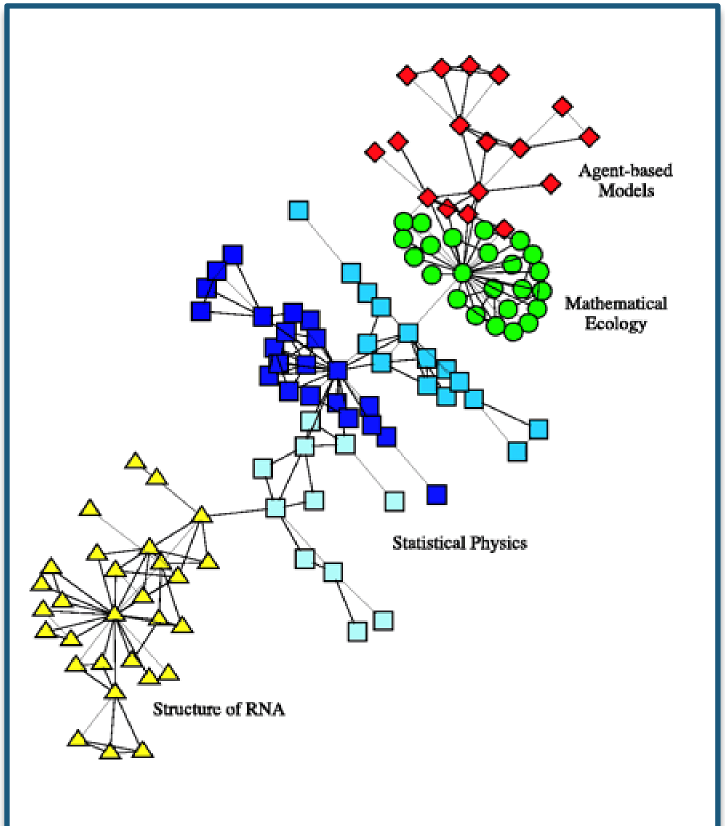
- **Question:** Do we really need all these diverse methodologies?
- **Answer:** Mainly, yes

1. **Overlapping communities:**
   - In some application domains we may not want to hardly assign nodes into only one community
   - The nodes of the graph may naturally have multiple memberships into communities
2. **Local vs. global methods:**
   - Computational issues
   - Applications where we are only looking for a community around some seed nodes or we need to partition the whole graph
3. **Structural observations:**
   - In **large scale real graphs**, it is difficult to find **good communities** **[Leskovec et al. '09]**

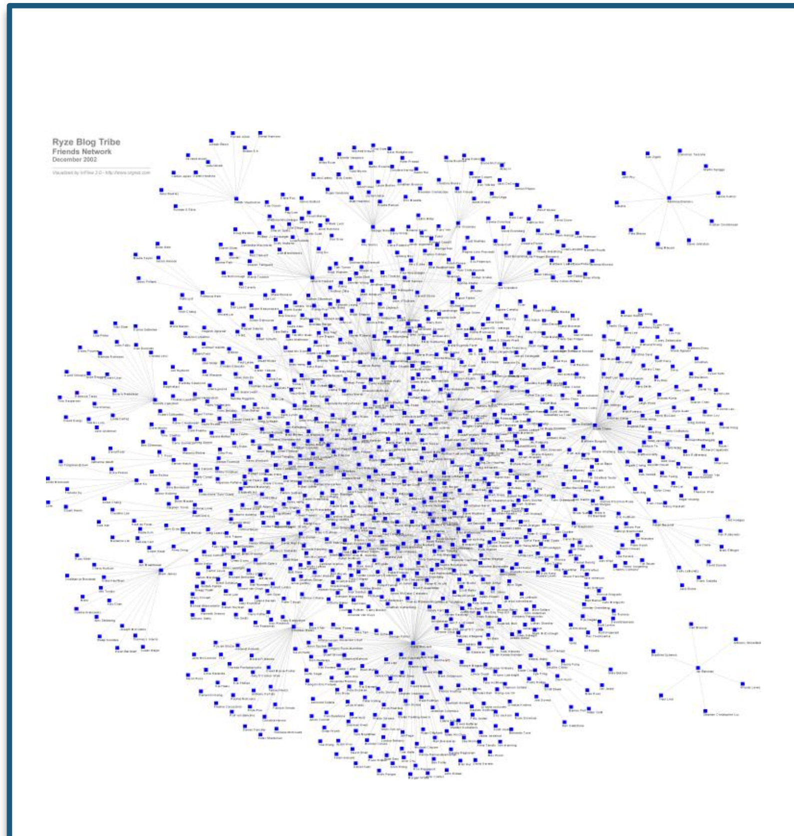
Topics on community detection and evaluation

- Overlapping community detection
- Global vs. local methods for community detection
- Community detection from seed nodes
- **Observations on structural properties of large graphs**
- Degeneracy-based community evaluation

Community structure in small vs. large graphs



Small scale collaboration network (Newman)

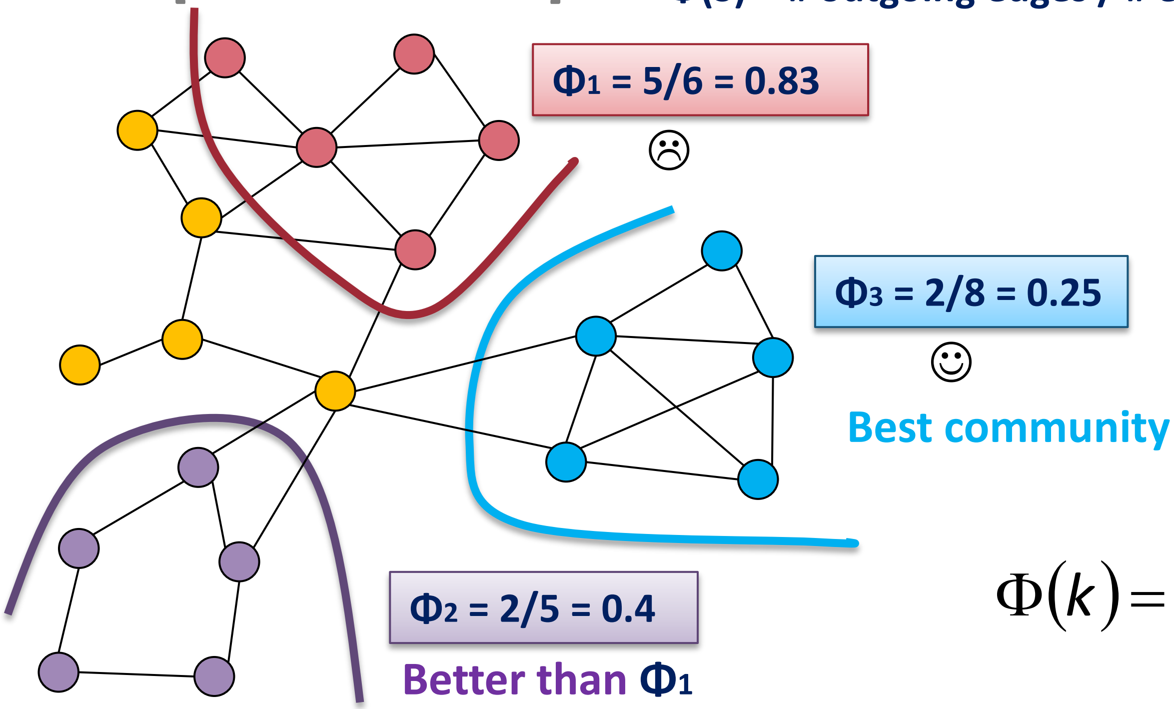


Blog network
http://www.ryze.com

Examine the structural differences

- How can we examine and compare the structural differences – **in terms of community structure** – at different scale graphs?
- Use **conductance Φ(S)** as a community evaluation measure
  - Smaller value for conductance implies better community-like properties [Leskovec et al. '09]

Φ(S) = # outgoing edges / # edges within



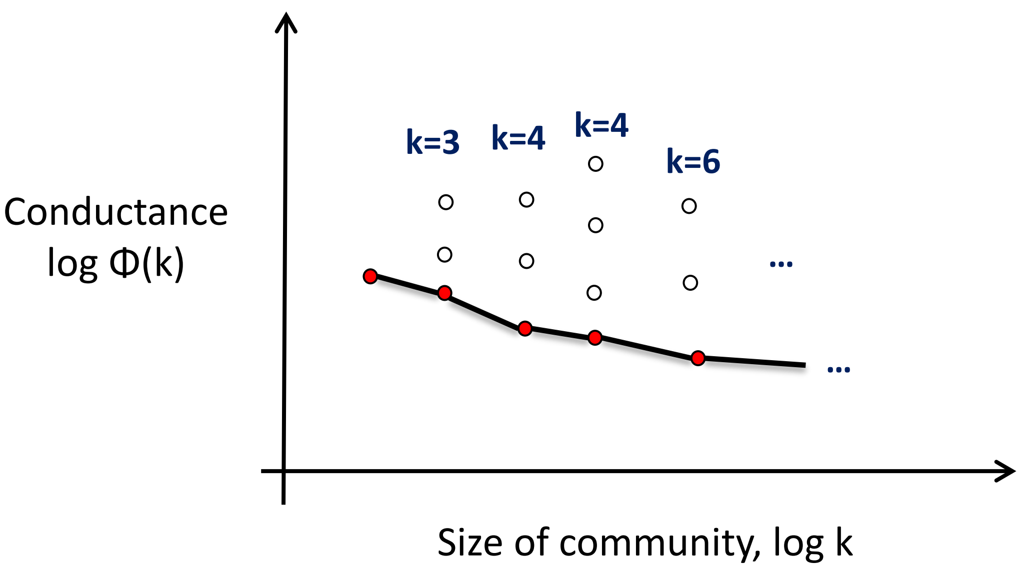
Find the best community of 5 nodes

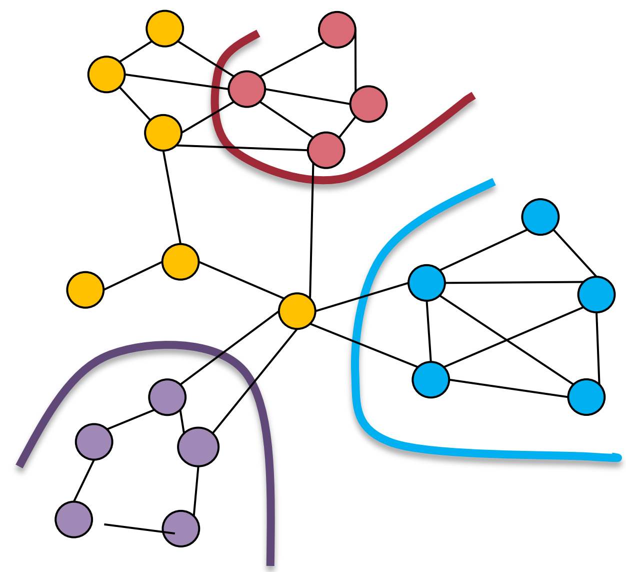
$$\Phi(k) = \min_{S \subset V, |S| = k} \Phi(S)$$

Example by J. Leskovec, ICML 2009

Network Community Profile plot

- **Network Community Profile** (NCP) plot **[Leskovec et al. '09]**
  - Plot the best conductance score (minimum) **Φ(k)** for each community size **k**

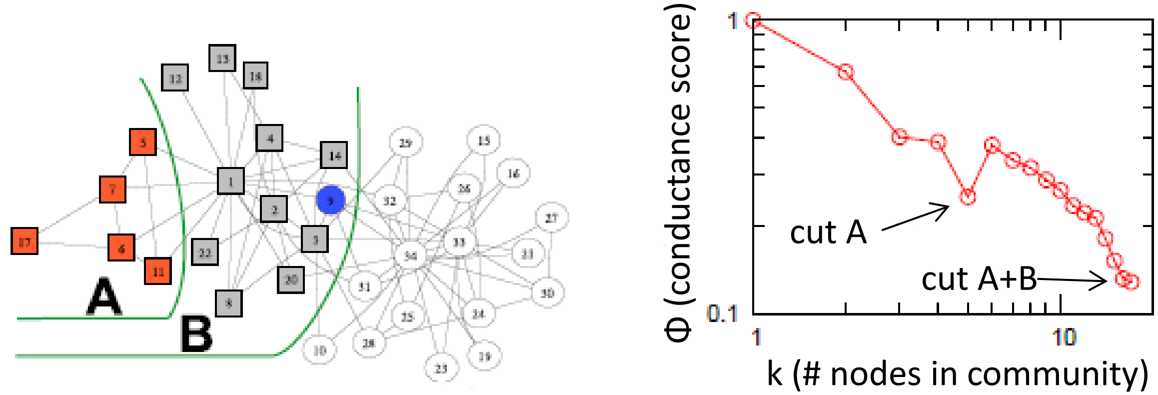




NCP plot of real graphs

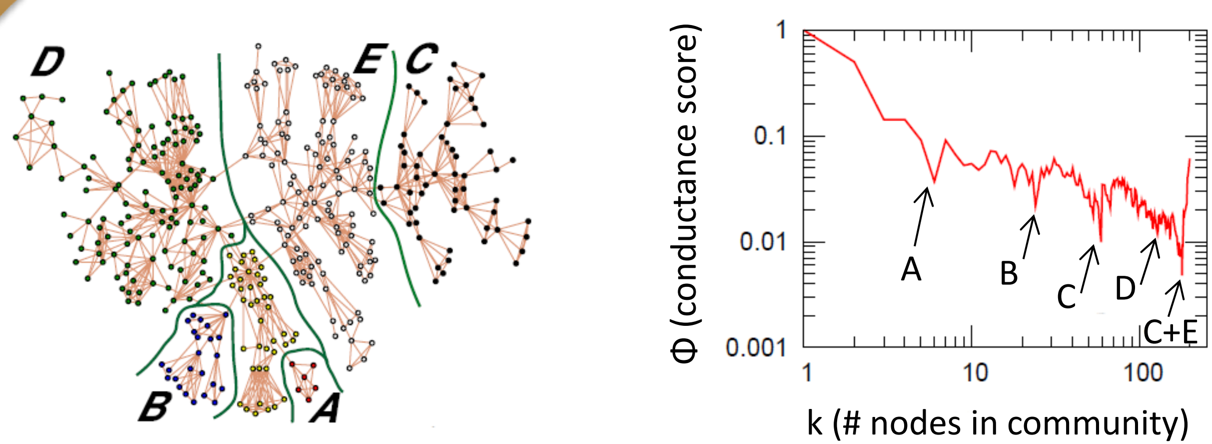
?

NCP plot examples



Zachary's karate club social network

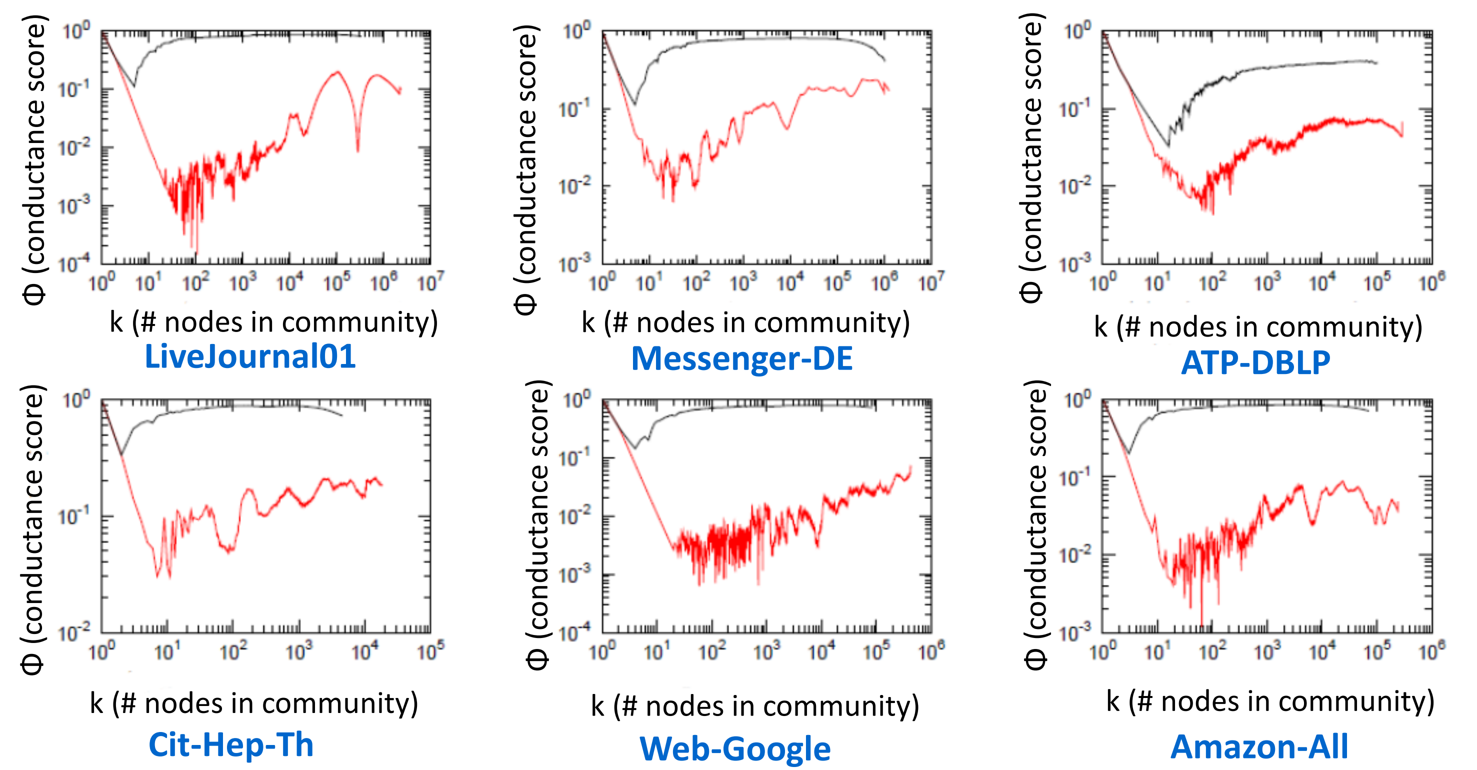
Small scale networks



Newman's collaboration network

[Leskovec et al. '09]

NCP plot of large real-world graphs



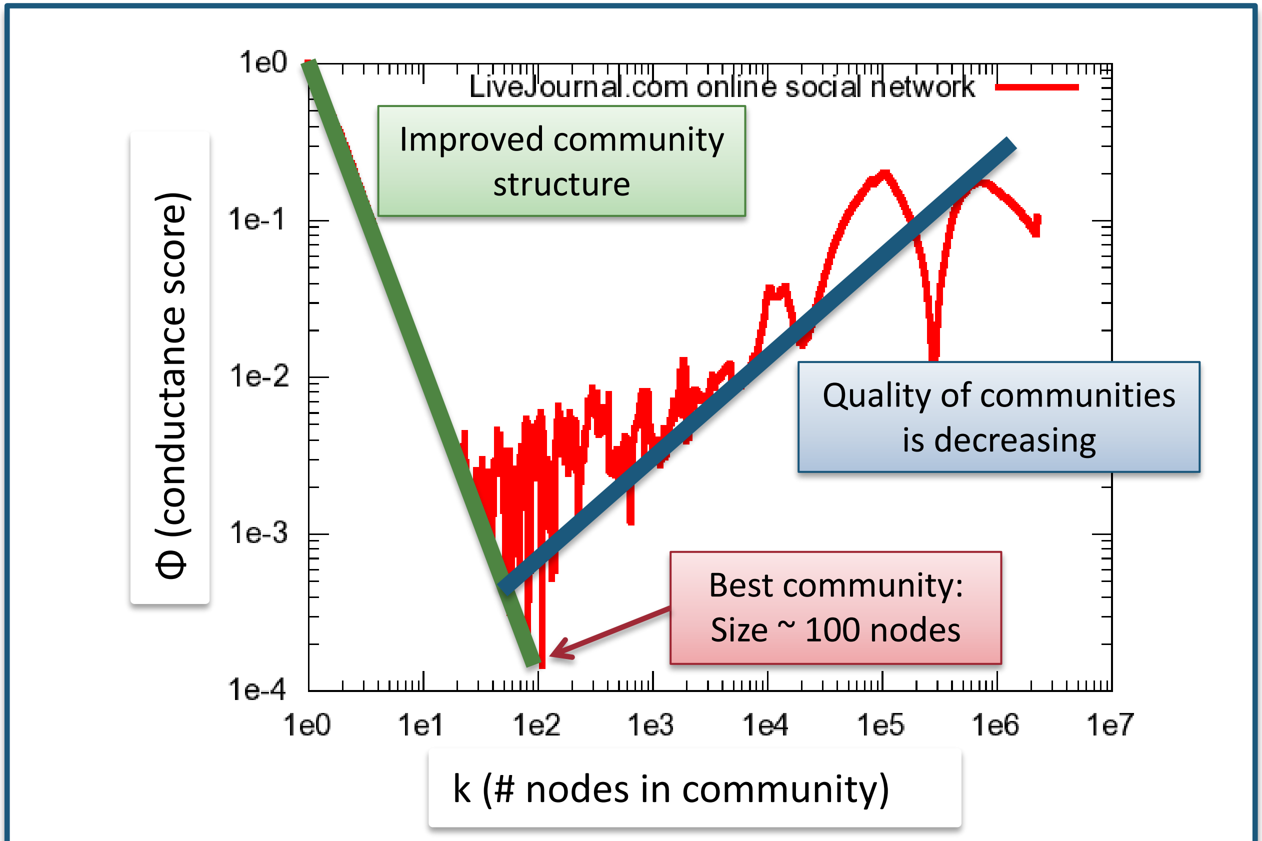
Large scale networks

Any common property?

[Leskovec et al. ’09]

Figure: J. Leskovec, ICML 2009

NCP plot: Observation in large graphs



LiveJournal social network

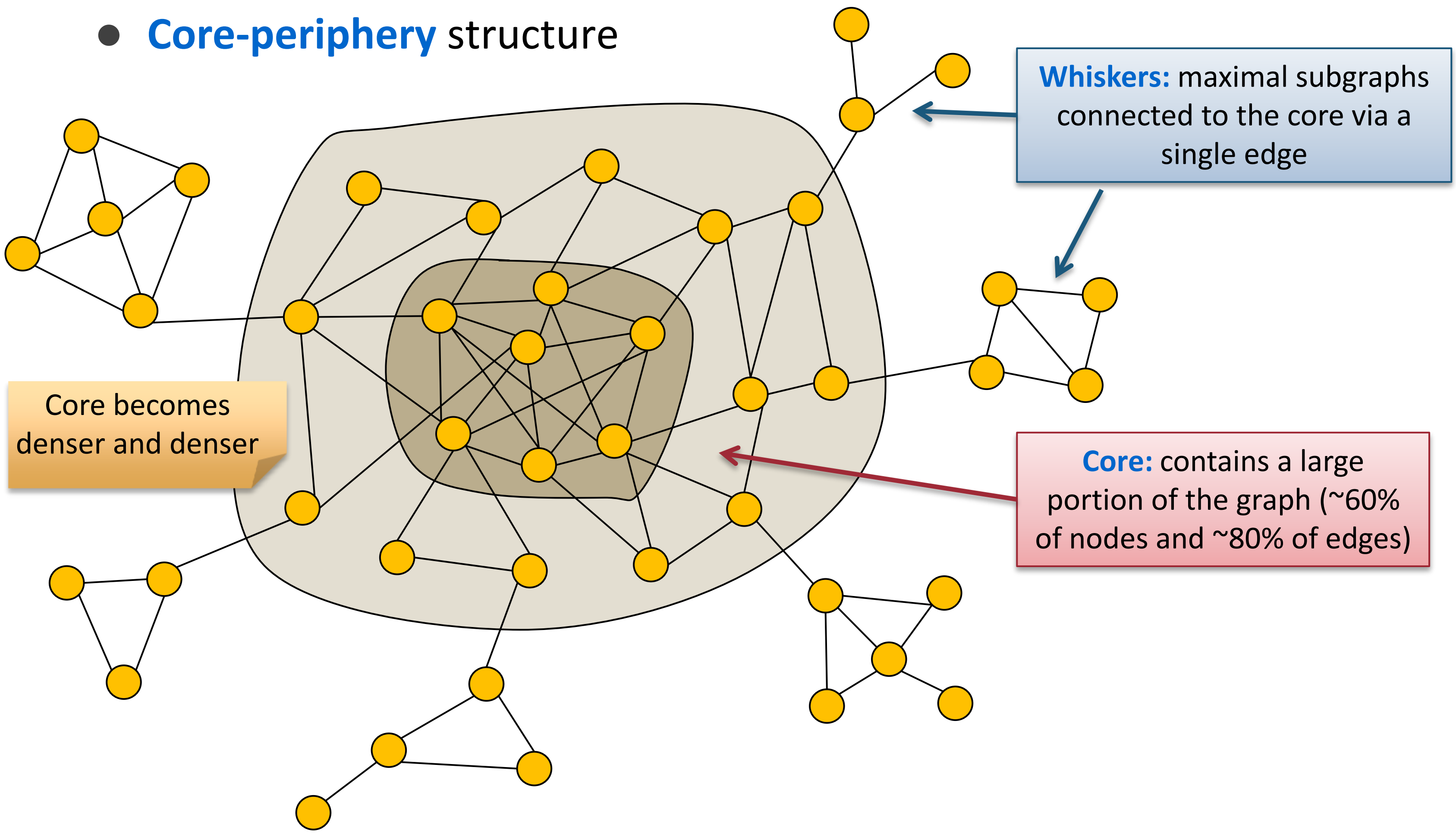
|V| = 5M, |E| = 42 M

Figure: http://snap.stanford.edu/ncp/
Slide by J. Leskovec, ICML 2009

[Leskovec et al. '09]

Explanation: Core-Periphery structure

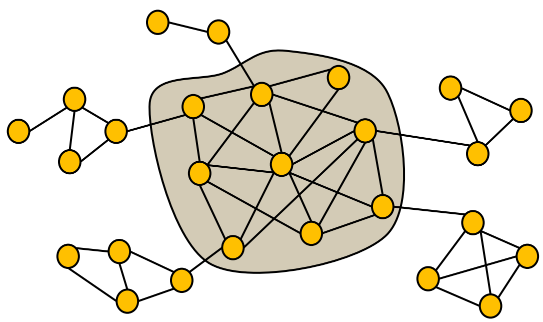
- How can we explain the observed structure of large graphs?
  - **Core-periphery** structure



[Leskovec et al. '09]

Core-Periphery structure

- **Core-periphery** structure
  - Core
  - Whiskers



- **Whiskers**
  - Non-trivial structure → more than random (shape and size)
- **Question:** What is happening if we remove whiskers (periphery) from graphs?
  - Almost nothing. The whiskers are replaced by **2-whiskers** (subgraphs connected to the core with 2 edges)
- The core itself has core-periphery structure
- **Important point:** Whiskers are also responsible for the best communities in large graphs (lowest point of NCP plot)

[Leskovec et al. ’09]

Similar structural observations

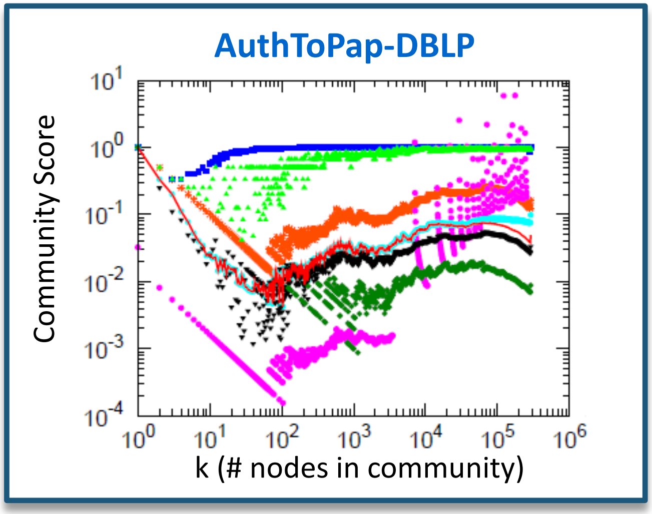
- **Jellyfish model** for the Internet topology **[Tauro et al. '01]**

- **Min-cut** plots **[Chakrabarty et al. '04]**
  - Perform min-cut recursively
  - Plot the relative size of the minimum cut

- **Robustness** of large scale social networks **[Malliaros et al. '12]**
  - Robustness estimation based on the expansion properties of graphs
  - Social networks are expected to show **low robustness** due to the existence of communities → the (small number of) inter-community edges will act as bottlenecks
  - Large scale social graphs tend to be extremely robust
  - **Structural differences** (in terms of robustness and community structure) between **different scale graphs**

Clustering algorithms and objective criteria

- **Question 1:** Is the observed property an effect of the used community detection algorithm (Metis + flow based method)?
  - **A:** No. The qualitative shape of the NCP plot is the same, regardless of the community detection algorithm **[Leskovec et al. ’09]**
- **Question 2:** Is the observed property an effect of the **conductance** community evaluation measure?
  - **A:** No. All the objective criteria that based on both internal and external connectivity, show a qualitatively almost similar behavior **[Leskovec et al. ’10]**
  - A *V*-like slope in the NCP plot



Conclusions

- **Large scale real-world graphs**
  - Core-periphery structure
  - No large, well defined communities
  - Structural differences between different scale graphs
- Community detection algorithms should take into account these structural observation
  - Whiskers correspond to the best (conductance-based) communities
  - Need larger high-quality clusters?
  - **Bag of whiskers:** union of disjoint (disconnected) whiskers are mainly responsible for the best high-quality clusters of larger size (above 100)

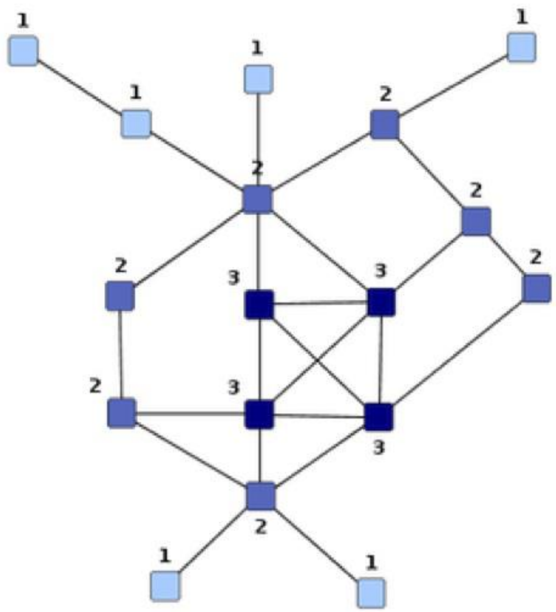
Topics on community detection and evaluation

- Overlapping community detection
- Global vs. local methods for community detection
- Community detection from seed nodes
- Observations on structural properties of large graphs
- **Degeneracy-based community evaluation**

Graph Degeneracy

- Degeneracy, for an **undirected** Graph G :
  - also known as the k-core number
  - *“the k-core of G is the largest sub-graph of G in which every vertex has degree of at least k within the sub-graph”*
- k-core decomposition:
  - find the k-core of G for all k
  - can be used as heuristics for maximum clique finding since a clique of size k
  - can give a (1/2)-approximation algorithm for the densest sub-graph problem

K-core



G_0 = G

G_0 : 1-core of G

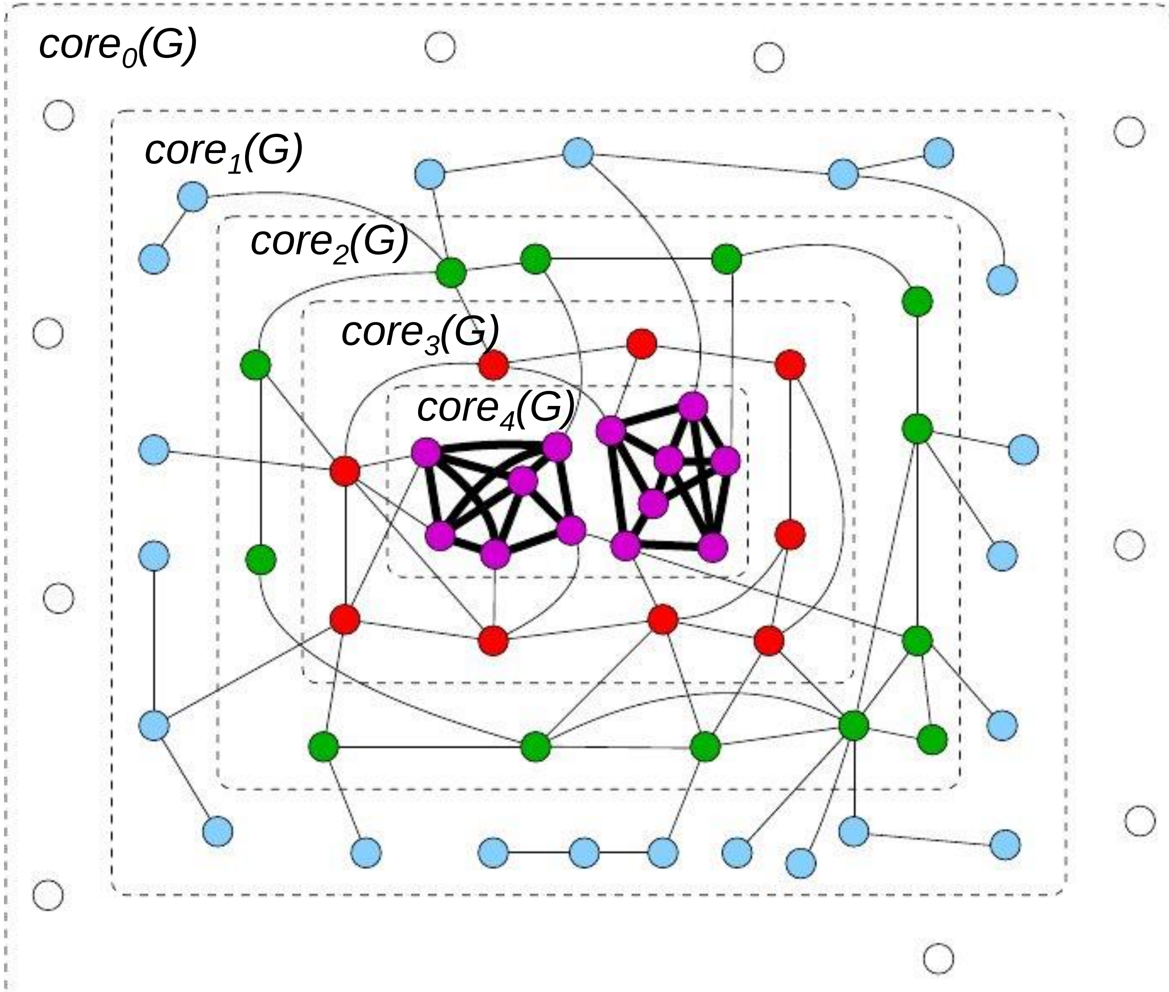
G_1 : 2-core of G

G_2 : 3-core of G

$$G_0 \supseteq G_1 \supseteq G_2 \supseteq G_3$$

- The degeneracy and the size of the maximum rank core provide a good indication of the cohesiveness of the graph G.

Another example



K-core

- The algorithm for computing the *k-th core of a graph:*

  Procedure *Trim$_k$(G, k)*

  Input: An undirected graph G and positive integer k

  *Output: k-**core**(G)*

  1. let F := G.

  *2. **while** there is a node x in F such that* ***deg$_F$***$(x)<k$

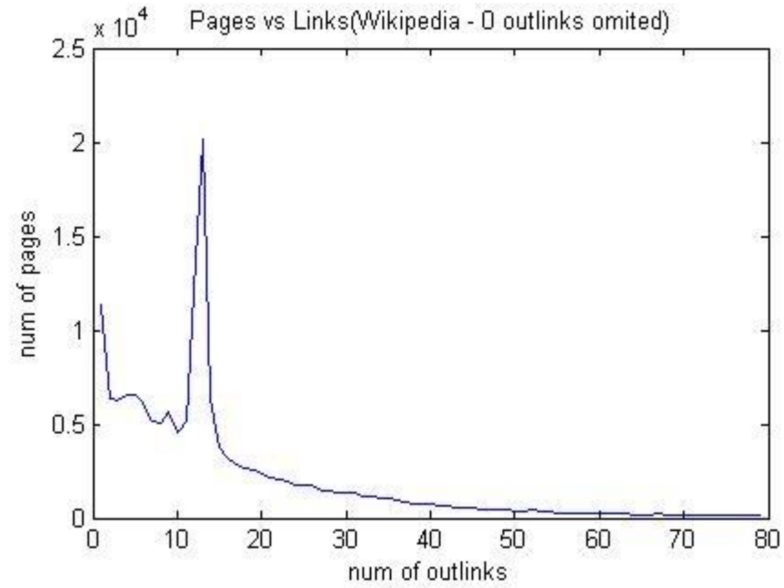
  *   **delete** node x from F.*

  *3. **return** F.*

- *Time complexity: O(n.k) (n= |G|)*
- Fast! especially in real word data where G is usually sparse.
  - requires the entire graph in memory

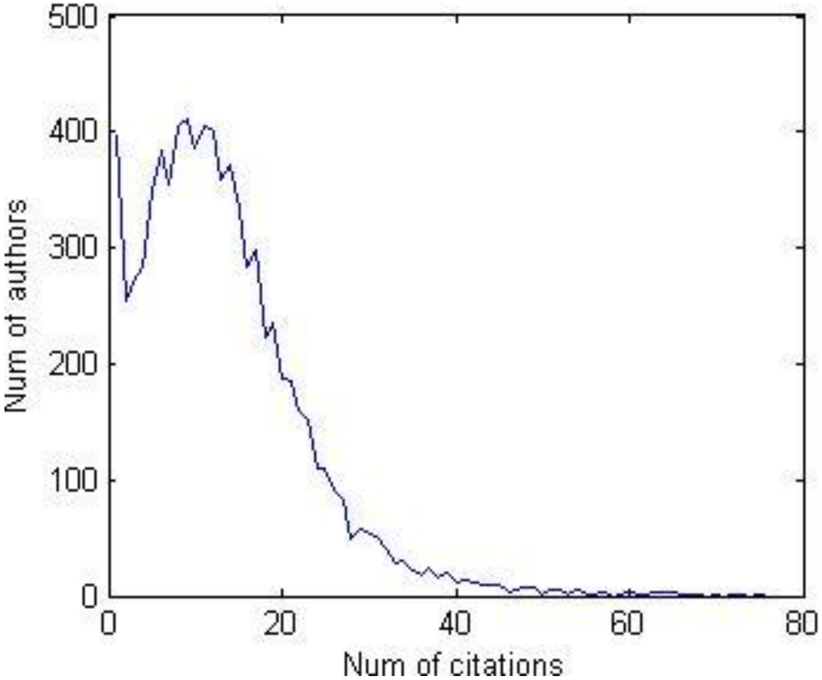
Wikipedia

- We consider only links among article-pages *within* Wikipedia
- A snapshot from January 2004 taken from the Wikipedia dump (freely available for download)
- # nodes: 1.2 M(unique pages)
- # links: 3.662 M



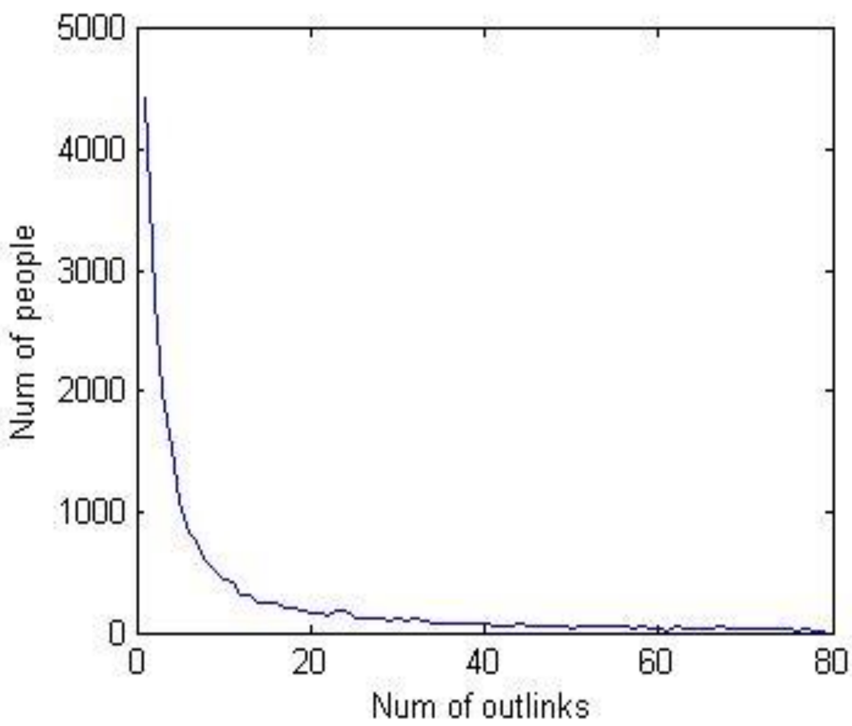
DBLP

- Taken from the DBLP data set
- Paper authored by *x,y,z* cites paper authored by *a,b,c*: creates directed citation-edges (x,a), (x,b), (x,c), (y,a)…
- 825 K author-nodes - 315K edges
- A very large part of authors has no in/out links (about 800 K) leaving the rest 25K to be examined

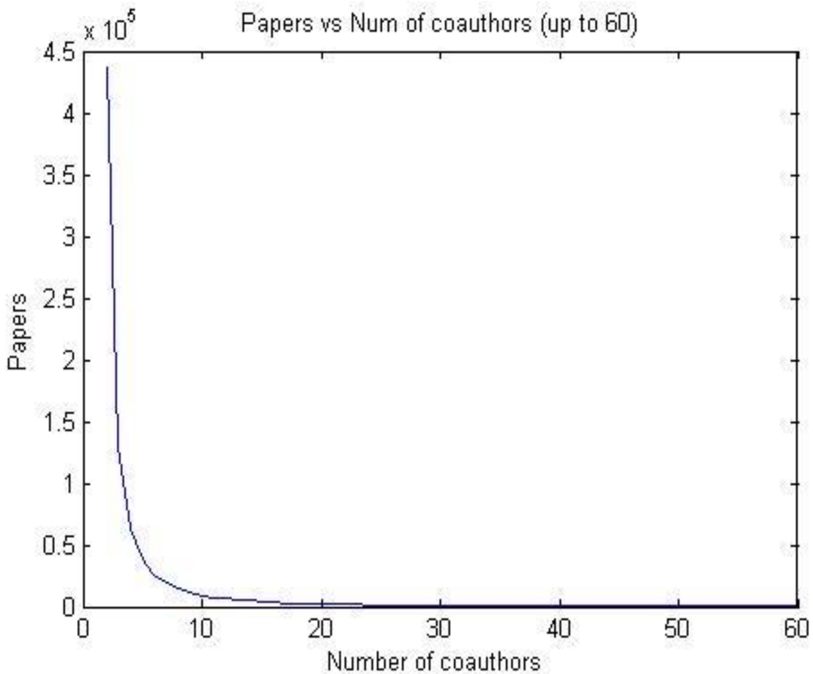


Epinions

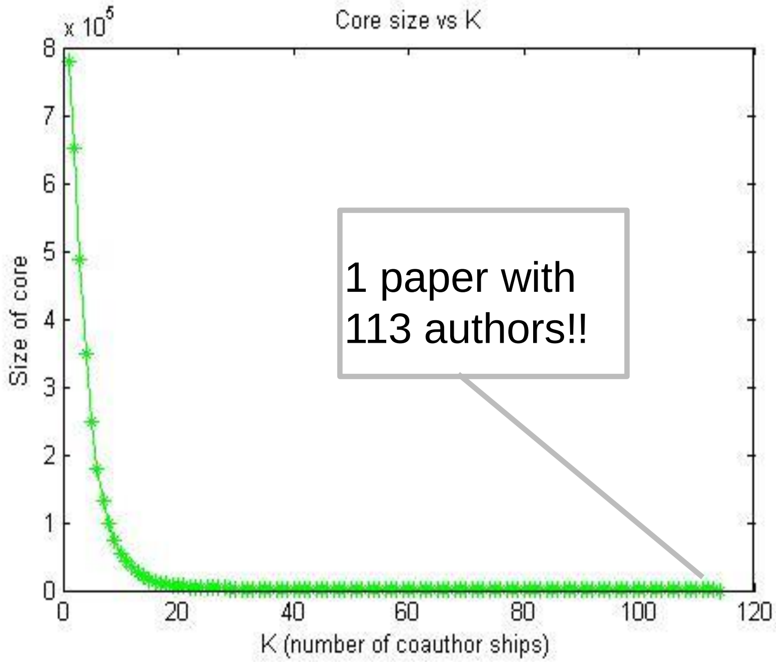
- A "who-trusts-whom" online social network of a general consumer review site Epinions.com (provided from Jure Leskovec among other graph data sets)
- #nodes: 75 K (users)
- #edges: 508 K (trust relations)



k-cores for the DBLP co-authorship graph



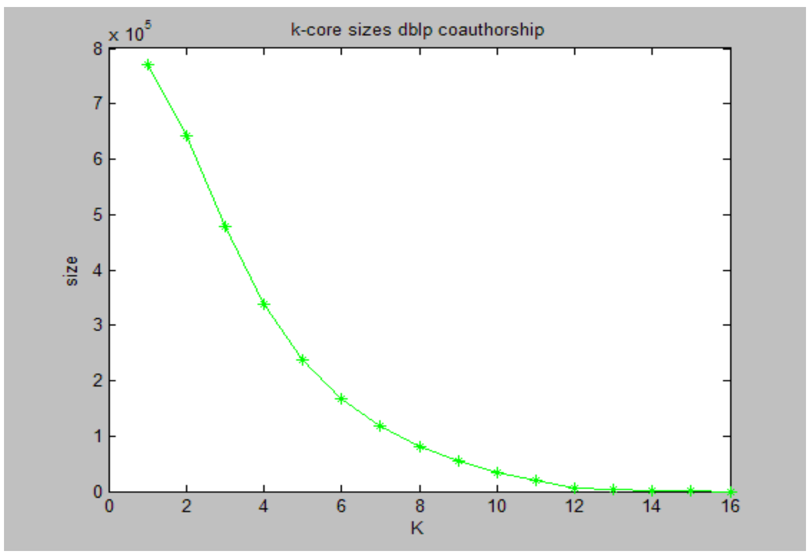
Distribution of the number of coauthors/paper k-core sizes in
In the unfiltered DBLP coauthorship graph



Distribution of the k-core sizes in
the unfiltered DBLP coauthorship graph

DBLP co-authorship – k-core on filtered graph

- Filtered out 1% of the papers
- max 15 authors/paper



Kurt Mehlhorn	Joseph S. B. Mitchell	Marc J. van Kreveld
Micha Sharir	David Eppstein	Martin L. Demaine
Pankaj K. Agarwal	Erik D. Demaine	Ferran Hurtado
Mark de Berg	Olivier Devillers	Timothy M. Chan
Rolf Klein	Sándor P. Fekete	Oswin Aichholzer
Mark H. Overmars	Henk Meijer	Bettina Speckmann
Herbert Edelsbrunner	Sariel Har-Peled	Jeff Erickson
Stefanie Wuhrer	John Hershberger	Therese C. Biedl
Jack Snoeyink	Alon Efrat	Greg Aloupis
Joseph O'Rourke	Stefan Langerman	David Bremner
Subhash Suri	Bernard Chazelle	Anna Lubiw
Otfried Cheong	Joachim	Esther M. Arkin
Hazel Everett	Gudmundsson	Boris Aronov
Sylvain Lazard	Giuseppe Liotta	Vida Dujmovic
Helmut Alt	Sue Whitesides	Suneeta Ramaswami
Emo Welzl	Christian Knauer	Thomas C. Shermer
Günter Rote	Raimund Seidel	David R. Wood
Leonidas J. Guibas	Michiel H. M. Smid	Perouz Taslakian
Chee-Keng Yap	Tetsuo Asano	John Iacono
Danny Krizanc	David Rappaport	Sergio Cabello
Pat Morin	Vera Sacristan	Sébastien Collette
Jorge Urrutia	Hee-Kap Ahn	Belén Palop
Diane L. Souvaine	Prosenjit Bose	Mirela Damian
Ileana Streinu	Michael A. Soss	Jirí Matousek
Dan Halperin	Godfried T. Toussaint	Otfried Schwarzkopf
Hervé Brönnimann		Richard Pollack

DBLP K-cores

- Extreme k-core: k=15 (DBLP), 76 authors
- Author ranking metric: max(k)-core that an author belongs to
  - e.g. Paul Erdos : 14
- On the max(k)-core we can identify the "closest" collaborators: **Hop-1 community**
  - Erdos hop-1 :

    Boris Aronov, Daniel J. Kleitman, János Pach, Leonard J. Schulman, Nathan Linial, Béla Bollobás, Miklós Ajtai, Endre Szemerédi, Joel Spencer,Fan R. K. Chung, Ronald L. Graham, David Avis, Noga Alon, László Lovász, Shlomo Moran, Richard Pollack, Michael E. Saks, Shmuel Zaks, Peter Winkler, Prasad Tetali, László Babai

K-core - issues

- Co-authorship graph: Authors participating in papers with many coauthors get biased credit
- i.e., in the unfiltered case:
  - 1 paper with 113 authors creates the most dense co-authroship collaboration structure
  - for most of the authors was the only paper
- Each author of a paper should get a just credit (i.e., 1/# authors)

Fractional k-cores

Co-authorship edge weight:

- For every edge *e = {x, x'}* we set
- *The weighted co-authorship affinity among x and x': collaboration !*

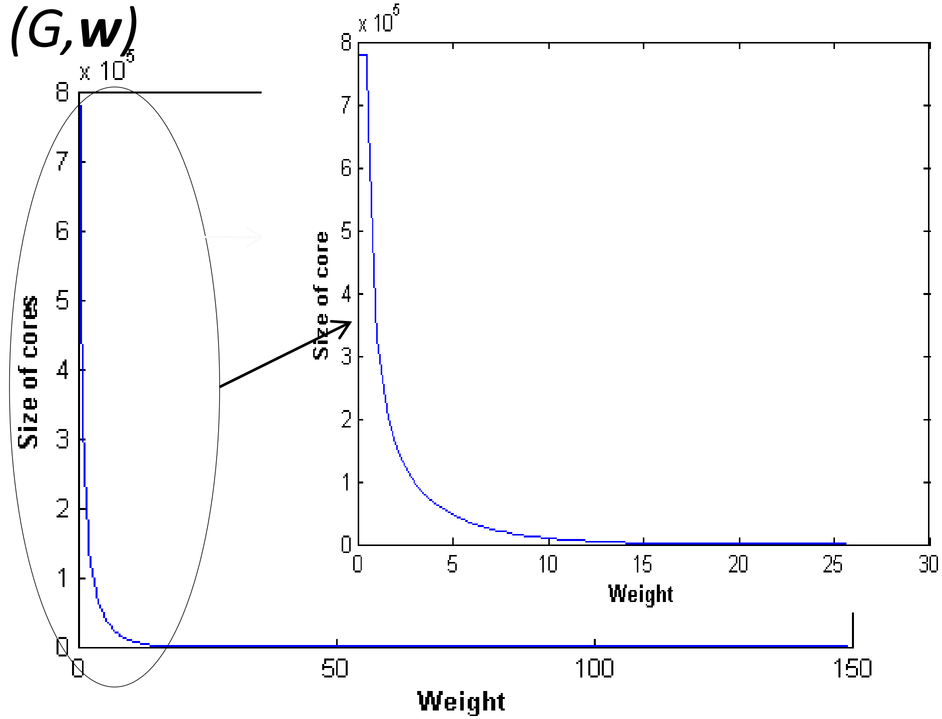
$$w(e) = \sum_{y \in N(x) \cap N(x')} \frac{1}{|N(y)|}$$

Vertex fractional degree. x in *(G,**w**)*

$$deg_{G,w}(x) = \sum_{e \in E(x)} w(e)$$

- the total co-authorship value of an author

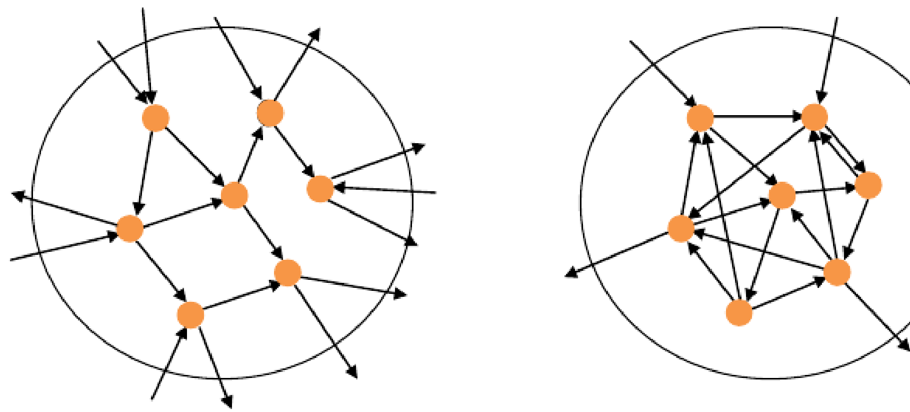
Distribution of the fractional k-core sizes in the DBLP coauthorship graph



Fractional cores and hop-1 list for selected authors.

Author	Core	Size	Hop-1 list
C.H. Papadimitriou	20.80	417	Mihalis Yannakakis 19.62 Erik D. Demaine 0.14 Georg Gottlob 1.0 Moshe Y. Vardi0.25
G.Weikum	16.30	1506	Hans-Jörg Schek 7.43 Surajit Chaudhuri 5.05 Raghu Ramakrishnan 0.41 Gustavo Alonso 0.43 Divyakant Agrawal 0.29 Yuri Breitbart 1.49 Amr El Abbadi 0.29 Catriel Beeri 0.33 Rakesh Agrawal 0.48 Abraham Silberschatz 0.17 Gautam Das 0.7 S. Sudarshan 0.2 Michael Backes 0.33 Jennifer Widom 0.19 David J. DeWitt 0.19 Stefano Ceri 0.275 Serge Abiteboul 0.33 Umeshwar Dayal 0.17 Michael J. Carey 0.14 ...
Tanenbaum	13.0	4016	Maarten van Steen 4.68 Frances M. T. Brazier 0.98 Howard Jay Siegel 0.13 M. Frans Kaashoek 7 Anne-Marie Kermarre 0.25 Robbert van Renesse 5.4 Michael S. Lew 0.02

Degeneracy on directed graphs



- Directed graphs:
  - WIKI - graph
  - DBLP – 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.
- This provides a two dimensional range where cores degenerate.
- Trade off between in/out edges can give us a more specific view of the cohesiveness and the “social" behavior

D-core matrix Wikipedia

Given a directed graph D. We define:

$$\delta^{in}(D) = \min\{x \mid \deg_D^{in}(x) \mid x \in V(D)\}$$

and

$$\delta^{out}(D) = \min\{x \mid \deg_D^{out}(x) \mid x \in V(D)\}$$

A (k,l)-D-*core* of D is a maximal sub-digraph F of

$$D: \delta^{\mathrm{out}}(\mathrm{F}) > \mathrm{k} \qquad \text{and} \qquad \delta^{in}(F) > l$$

Degeneracy on directed graphs

(k,l)-D-*core* (G): the (k,l) D-core of graph G

for each k,l : $dc_{k,l} = |(k,l)\text{-D-}core\ (G)|$

D-core matrix: $D(k,l)=dc_{k,l}$, k,l integers – each cell stores the size of the respective D-core

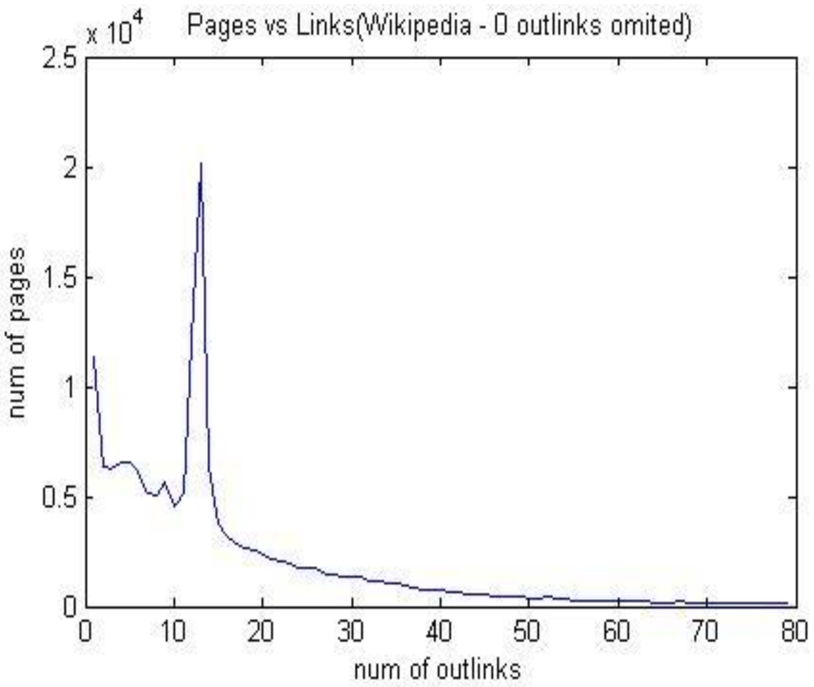
Frontier: $F(D) = \{(k;l): dc_{k,l} > 0\ \&\ dc_{k+1,l+1} = 0\}$: the extreme (k,l)-D-cores

Collaboration indices

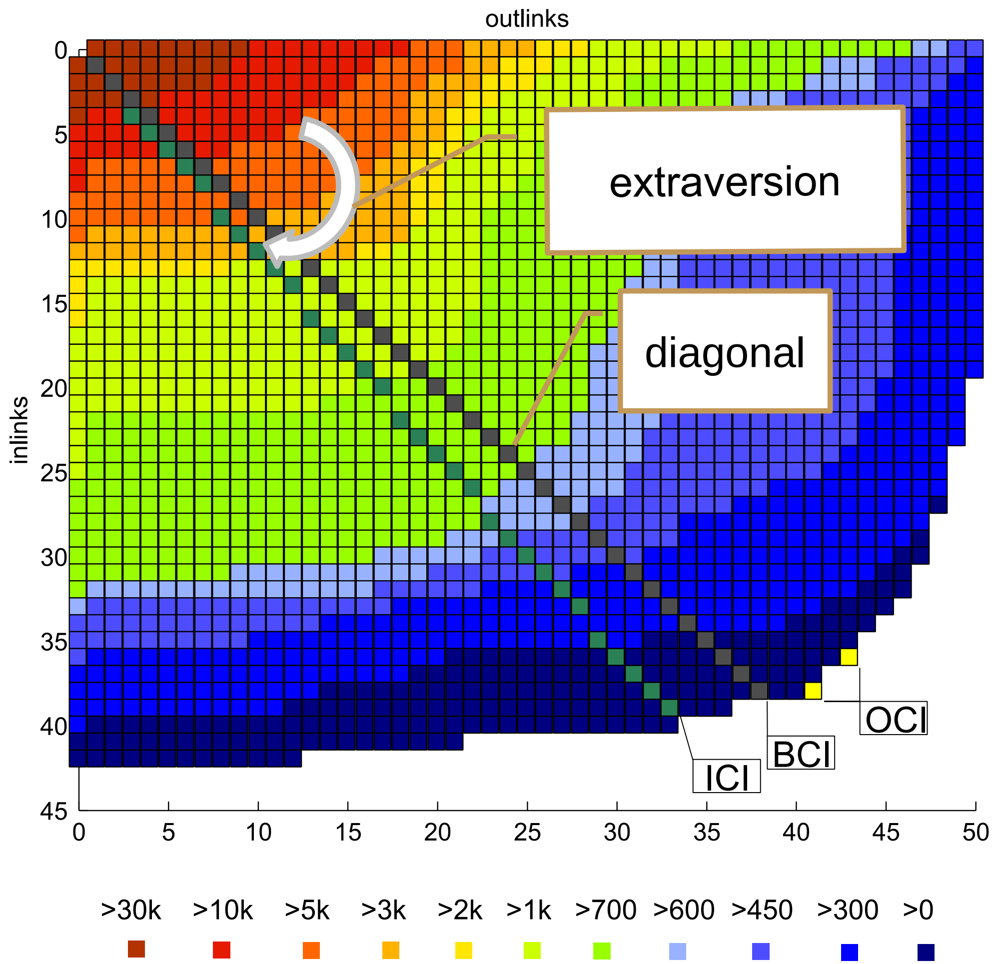
- Balanced collaboration index (**BCI**) : Intersection of diagonal $D(k,k)$ with frontier
- Optimal collaboration index (**OCI**) : $DC(k,l)$ where $\max((k+l)/2)$ distance from $D(0,0)$
- Inherent collaboration index (**ICI**): All cores on the angle defined by the average inlinks/outlinks ratio

Wikipedia

- Wikipedia Terms pages
- We consider only links among article-pages *within* Wikipedia
- January 2004 snapshot
- # nodes: 1.2M(unique pages)
- # links: 3.662 M

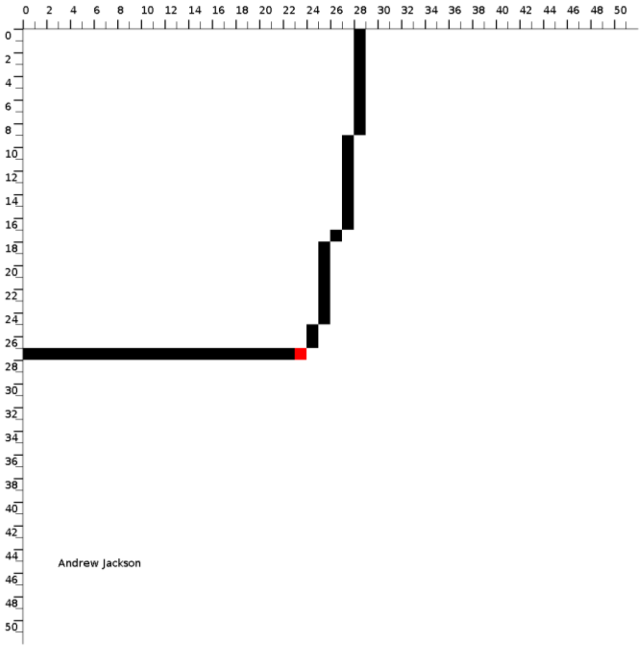


D-core matrix Wikipedia

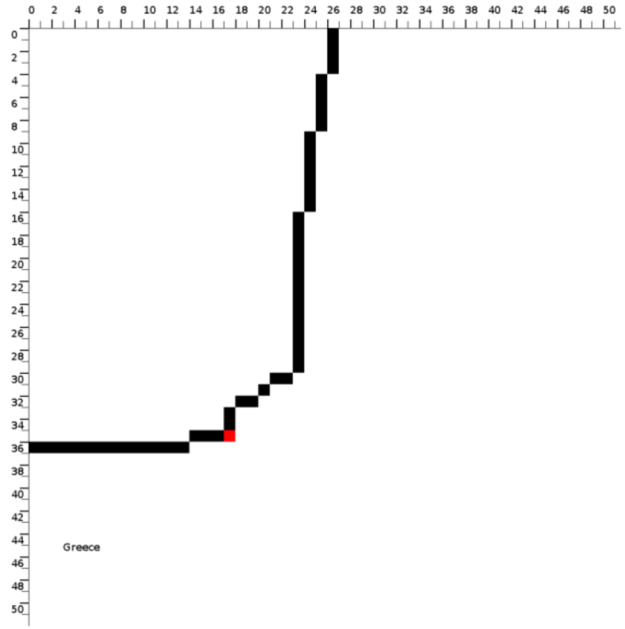


The extreme D-core(38,41) contains 237 pages

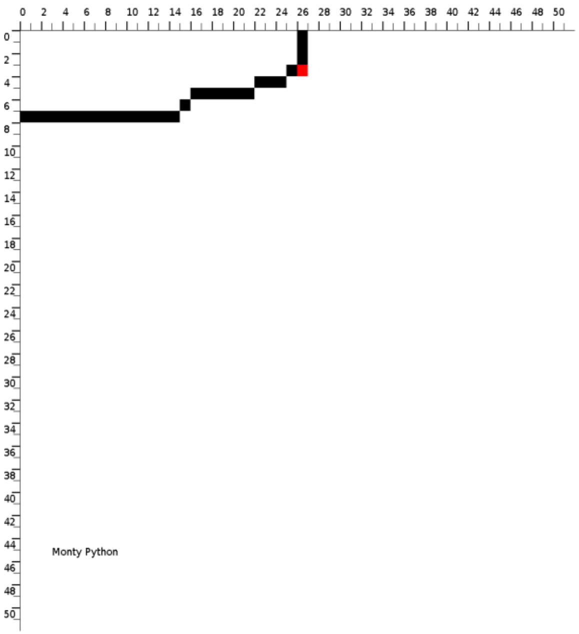
Thematic D-core frontiers - Wikipedia



“Andrew Jacson”

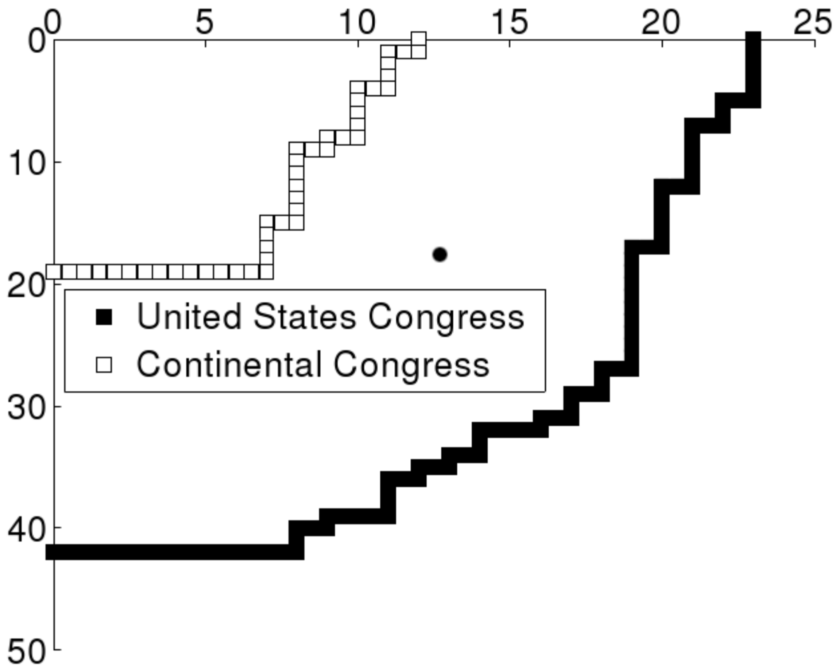


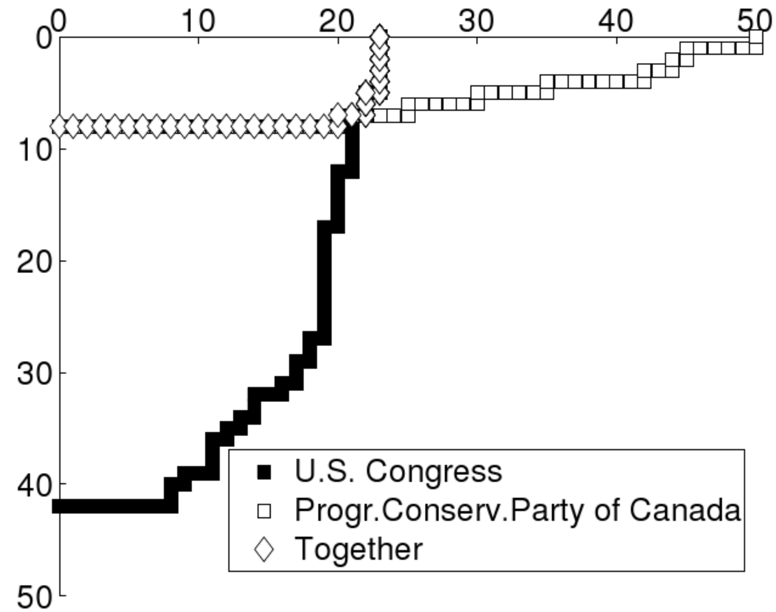
“Greece”



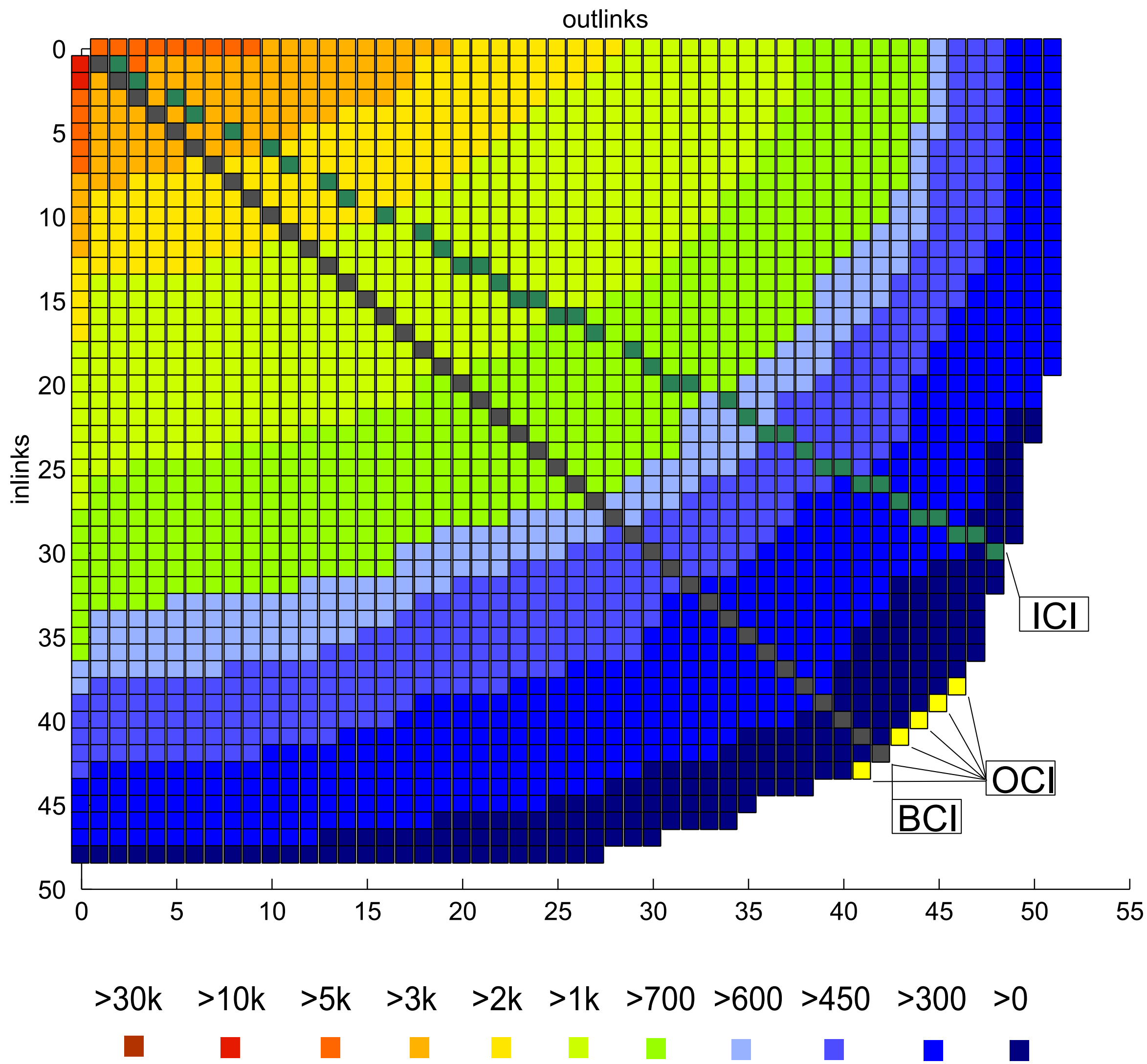
“Monty Pythons”

Thematic D-core frontiers - Wikipedia





D-core matrix for DBLP



The Extreme DBLP D-core authors

- JosÃ© 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 HÃ¤rder
- FranÃ § ois 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 SaccÃ
- 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

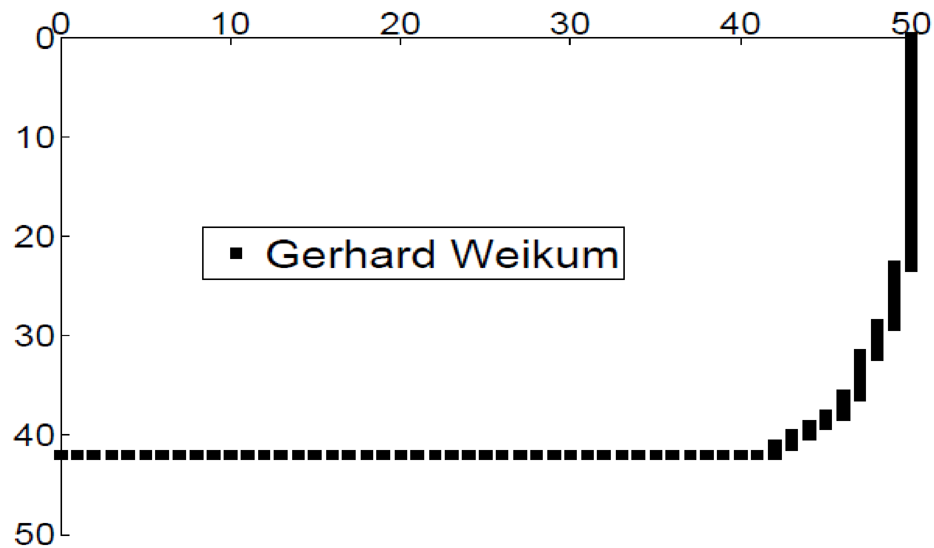
- Michel E. Adiba
- Kyuseok Shim
- Goetz Graefe
- Jiawei Han
- Edward Sciore
- Rakesh Agrawal
- Carlo Zaniolo
- V. S. Subrahmanian
- Claude Delobel
- Christophe LÃ©cluse
- Michel Scholl
- Peter C. Lockemann
- Peter M. Schwarz
- Laura M. Haas
- Arnon Rosenthal
- Erich J. Neuhold
- Hans-JÃ¶rg Schek
- Dirk 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
- JÃ¼rgen Koch
- Joachim W. Schmidt
- Guy M. Lohman
- Bruce G. Lindsay
- Paul F. Wilms
- Z. Meral Ã–zsoyoglu
- Gultekin Ã–zsoyoglu
- Kyu-Young Whang
- Shahram Ghandeharizadeh
- Tova Milo
- Alon Y. Levy
- Georg Gottlob
- Johann Christoph Freytag
- Klaus KÃ¼spert
- Louiqa Raschid
- John Mylopoulos
- Alexander Borgida
- Anand Rajaraman
- Joseph M. Hellerstein
- Masaru Kitsuregawa
- Sumit Ganguly
- Rudolf Bayer
- Raymond T. Ng
- Daniela Florescu
- Per-Ã…ke 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

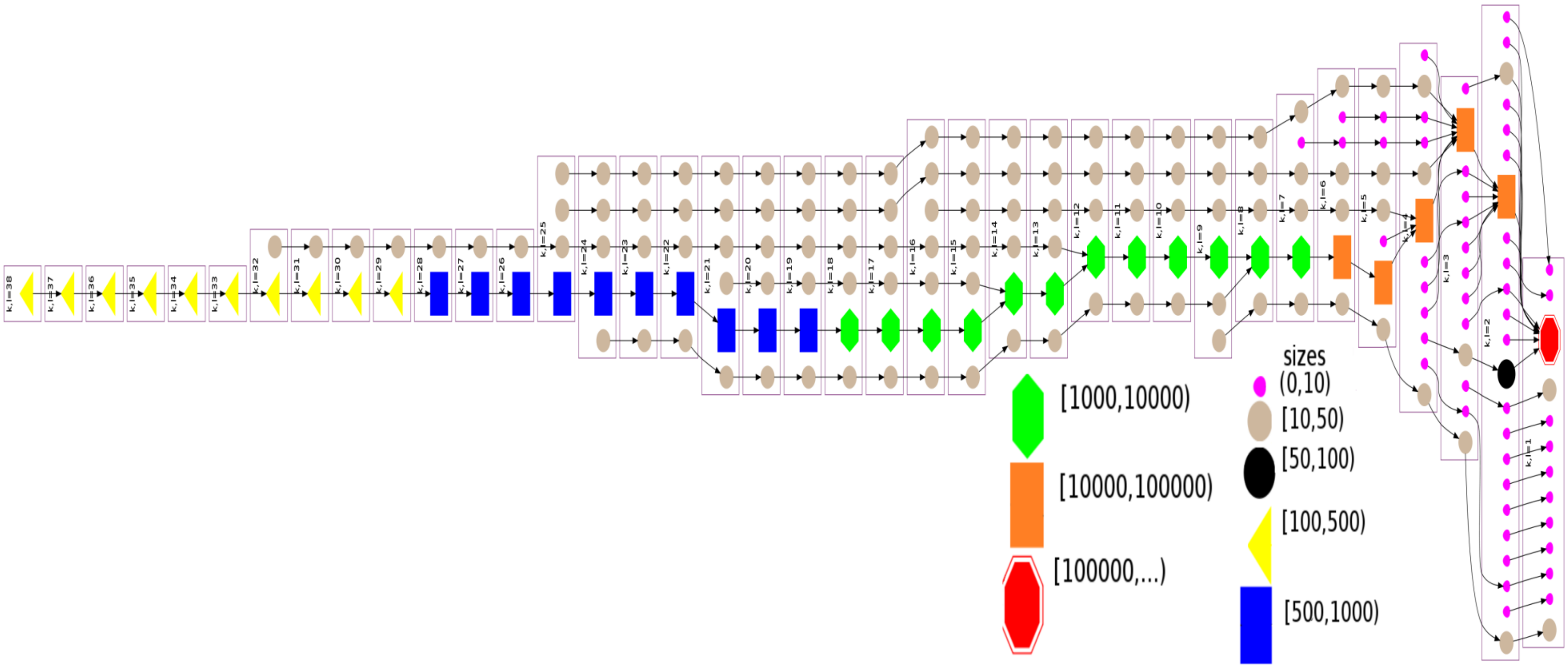
- The frontier of an individual: defined by the outmost d-cores that the individual belongs to.
- We can evaluate the citation based robustness of an individual within the community by her frontier.



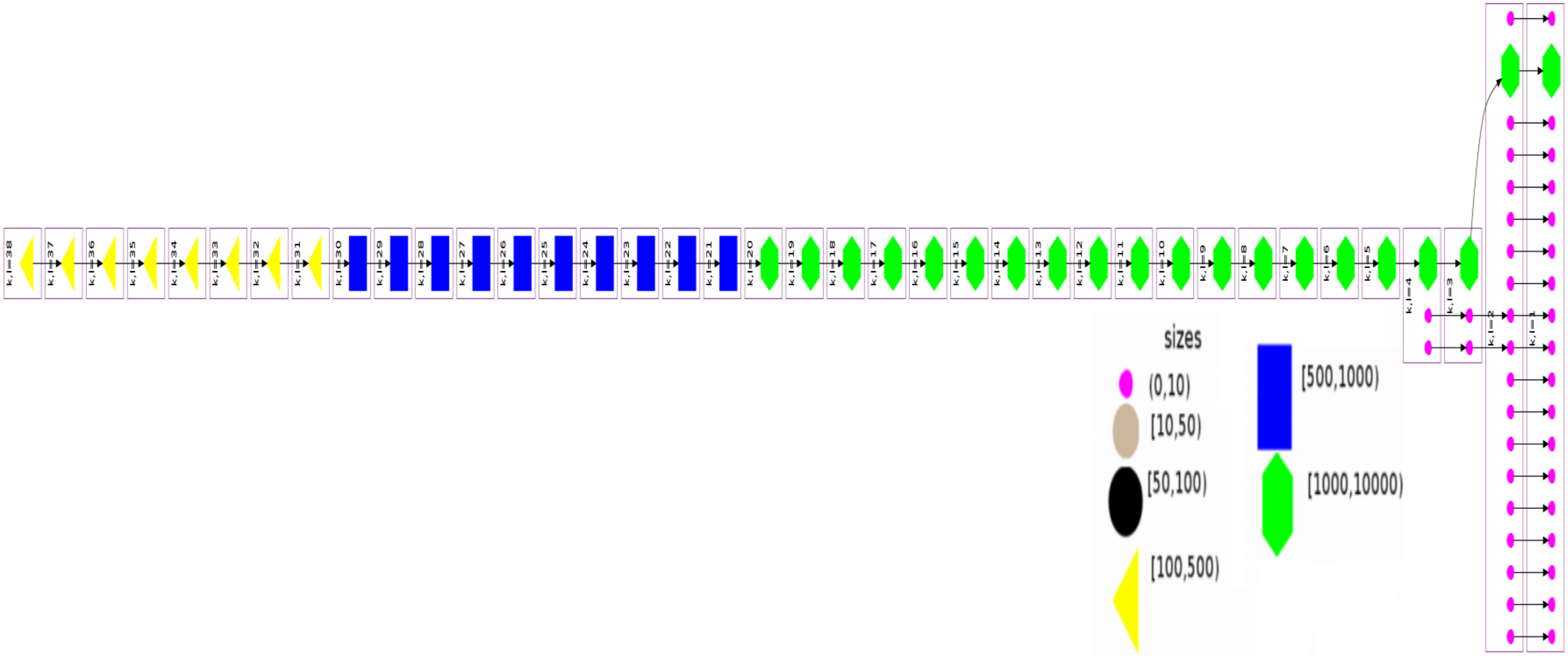
Strongly Connected Components (SCC)

- The SCC’s are not usually used for community detection but in this case serve well for indicating how the communities would survive throughout the cores.
- We trace SCC’s through the D-Cores(k,k) – i.e., on the “diagonal” direction.

Wikipedia SCCs



DBLP SCCs



Degeneracy in Signed graphs

- Signed (directed) graphs can depict a wide variety of concepts. We define degeneracy upon a “trust“ network.
- A member of a directed signed graph **G** can either trust or distrust an other but not the both simultaneously.
- Obviously self links are of no interest.
- Each vertex **v** has both positive & negative in-degree and both positive & negative out-degree

S-cores

- Degeneracy in directed graphs has a simple 2-dimensional visualization in 2 axes.
- We consider each case of trust interaction (positive/negative, in/out) as a separate case
  - high complexity in comprehension of the results
  - Some combinations could be explored only with d-cores (i.e. in/out degree for the same sign)
  - The purpose of the extensions is to examine/evaluate the underlying community under the scope of TRUST/DISTRUST
- Solution: Consider as one dimension the in/out degrees

S-cores

- We compute the trust network degeneracy degeneracy along the 4 combinations of direction and sign (in,out):
  - **(+,+)**: Mutual Trust
  - **(+,-)**: Trust under distrust (i.e. trust those who do not trust me)
  - **(-,-)**: Mutual distrust
  - **(-,+)**: Distrust under trust

Definitions

- Given a pair $(s, t) \in \{+,-\}^2$, we define the (s, t)-degeneracy of G:
  $\delta^{s,t}(G) = \max\{(k+l)/2 \mid G \text{ contains a non-empty } (k^s, l^t)\text{-d-core}\}$
- 4 quadrants -> we define frontiers much like the d-cores :
  - $R_G = \{(i, j) \in F^2_G \;\; a_{i,j} > 0 \text{ and } a_{i+1,j+1} = 0\}$ (the extreme non-empty S-cores)

Data description

- Explicit Data: Epinions, Slashdot
- Implicit Data: Wikipedia Topics
- Explicit: data that describe an existing trust network
- Implicit: Inferred data, extracted from user interactions (edit, delete, revert actions in articles)

Data Statistics

Explicit

Network	Nodes	Edges	Negative
Epinions	119,217	841,200	15.0%
Slashdot	82,144	549,202	22.6%

Implicit (Wikipedia)

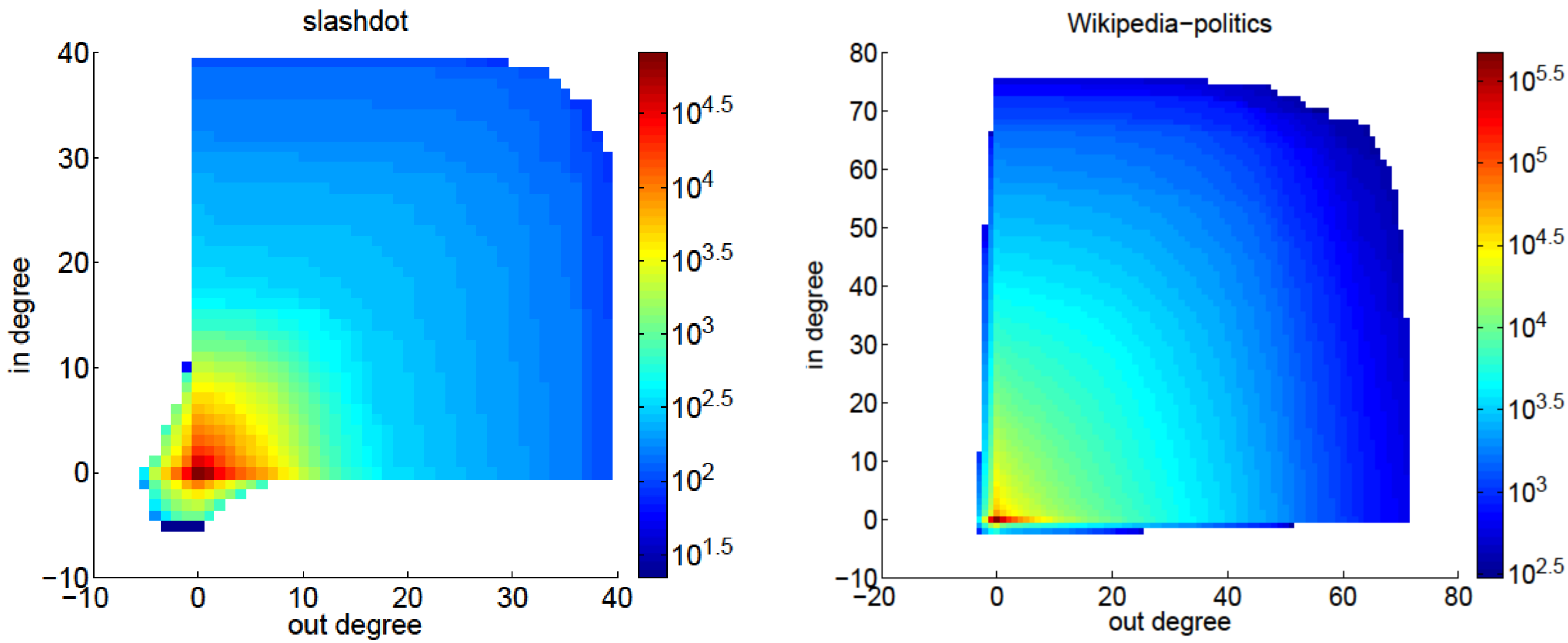
Domain	Articles	Nodes	Edges	Positive	Negative
History	3,331	141,983	534,693	439,193	95,500
Politics	12,921	453,116	2,428,945	2,099,410	329,535
Religion	6,459	277,482	1,423,279	1,244,166	179,113
Mathematics	9,610	158,671	651,450	548,073	103,377

Wikipedia graph inference

- Types of interactions extracted:
  1. number of words inserted by the author of the current revision in the vicinity of the text belonging to other authors
  2. number of words deleted or replaced between the current revision and the previous
  3. if the current revision is a reversion (restoration) reversions can be by one author upon many.
- Additionally, between the authors of the above revisions:
  1. votes in administrator elections
  2. barnstars, i.e., prizes acknowledging important contributions, which can be put on a user's profile page by other contributors.

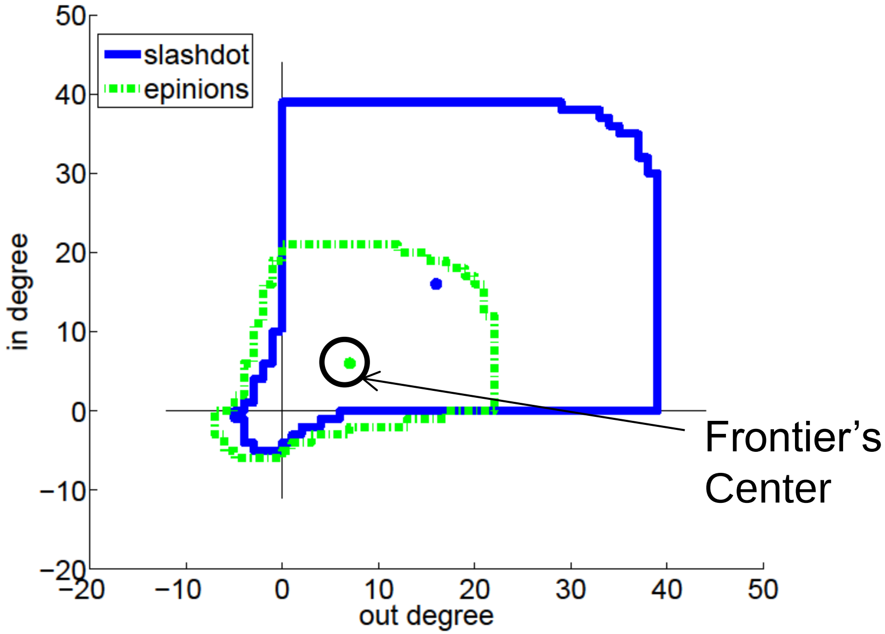
Examples

- S-Cores sizes on real world data

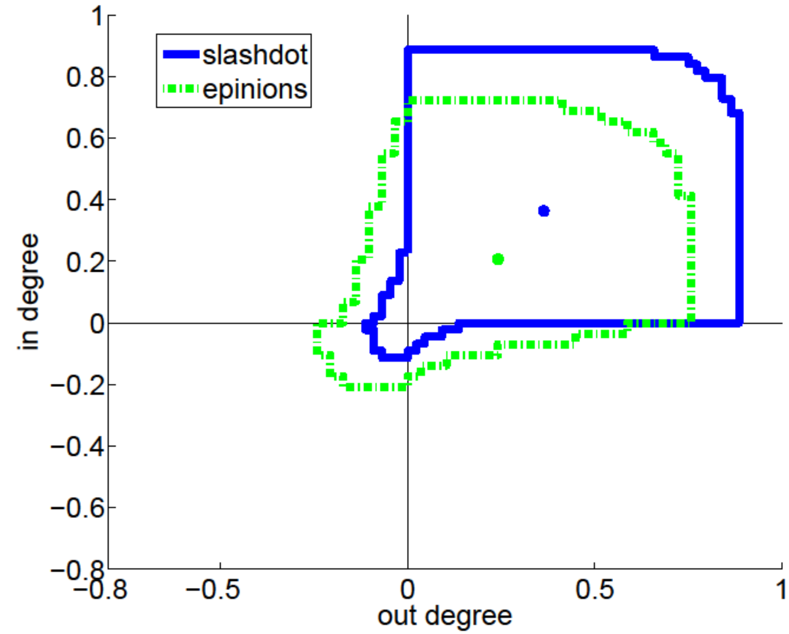


- In both cases positive trust dominates
- In slashdot there is proportionaly much more mutual distrust than in the wikipedia-politics case

Frontiers (explicit graphs)



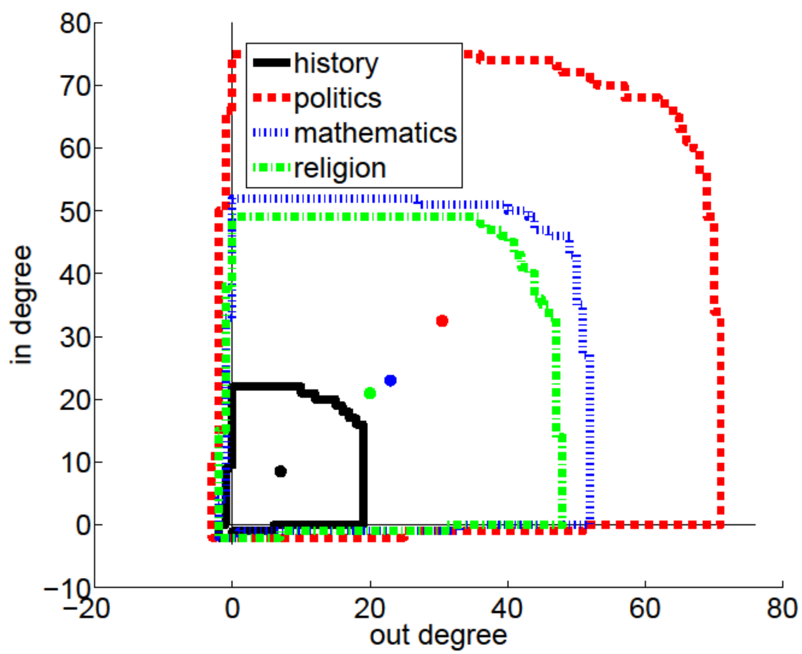
Original frontiers



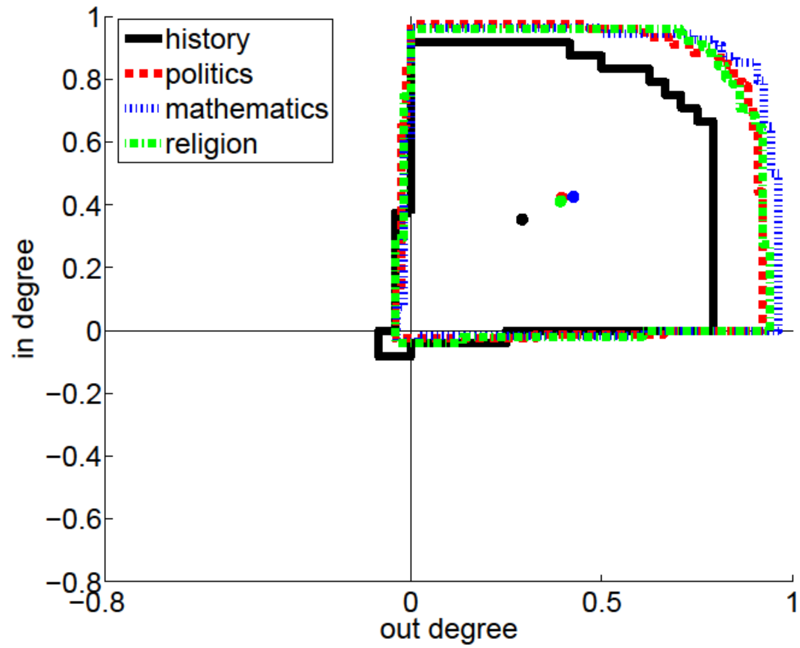
Normalized

- Slashdot is more robust under degeneracy than the epinions trust graph: i.e., there is denser mutual trust in S than in E.
- In epinions the mutual negative distrust is much more important as well as the imbalanced trust graphs (+/-, -/+)

Wikipedia Topics



Original frontiers



Normalized

- Wikipedia *politics* is the most robust trust network, *history* is the least one
- In the normalized case: history is the one with the largest mutually negative trust constituent

Users and Articles Frontiers

- We utilize the s-core structure to evaluate the trustworthiness of a user or an article
- Users

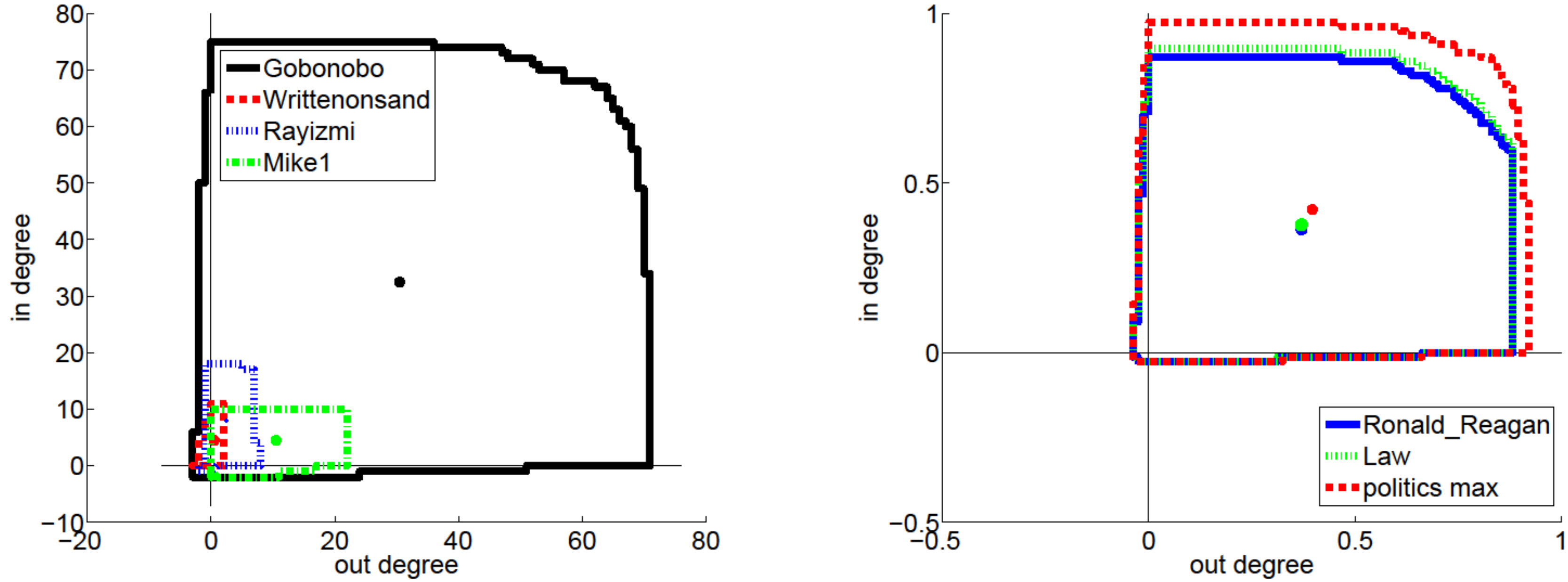
  the user frontier is defined by intersection of the s-cores she participates

- Article frontier:

  Multiple users contribute to an Article

  The intersection of their contributing editors individual frontiers is the article frontier.

Users & Articles



- **Editors**

  Gobonobo is by far the most trusting and trusted one – i.e. a very senior editor

- **Article frontier**

  "Reagan" article is almost as trusted as the "Politics" topic.

References (community evaluation measures)

- G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. Nature 435(7043), 2005.
- I. Farkas, D. Ábel, G. Palla, and T. Vicsek. Weighted network modules. New J. Phys. 9(180), 2007.
- S. Lehmann, M. Schwartz, and L.K. Hansen. Biclique communities. Phys. Rev. E 78(1), 2008.
- J.M. Kumpula, M. Kivelä, K. Kaski, and J. Saramäki. Sequential algorithm for fast clique percolation. Phys. Rev. E 78, 2008.
- P. Pollner, G. Palla, and T. Vicsek. Parallel clustering with CFinder. Parallel Processing Letters 22, 2012.
- R. Andersen and K.J. Lang. Communities from Seed Sets. In: WWW, 2006.
- R. Andersen, F. Chung, and K.J. Lang. Local Graph Partitioning using PageRank Vectors. In: FOCS, 2006.
- R. Andersen and Y. Peres. Finding Sparse Cuts Locally Using Evolving Sets. In: STOC, 2009.
- D. Gleich and C. Seshadhri. Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In: KDD, 2012.
- A.S. Maiya and T.Y. Berger-Wolf. Sampling Community Structure. In: WWW, 2010.

References (community evaluation measures)

- J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters. Internet Mathematics 6(1), 2009.
- S.L. Tauro, C. Palmer, G. Siganos, and M. Faloutsos. A simple conceptual model for the internet topology. In: GLOBECOM, 2001.
- D. Chakrabarti, Y. Zhan, D. Blandford, C. Faloutsos and G. Blelloch. NetMine: New Mining Tools for Large Graphs. In: SDM Workshop on Link Analysis, Counter-terrorism and Privacy, 2004.
- F.D. Malliaros, V. Megalooikonomou, and C. Faloutsos. Fast robustness estimation in large social graphs: communities and anomaly detection. In: SDM, 2012.
- J. Leskovec, K.J. Lang, and M.W. Mahoney. Empirical comparison of algorithms for network community detection. In: WWW, 2010.

References (degeneracy)

- C. Giatsidis, D. Thilikos, M. Vazirgiannis,"D-cores: Measuring Collaboration of Directed Graphs Based on Degeneracy", Knowledge and Information Systems Journal, Springer, 2012.
- Christos Giatsidis, Dimitrios M. Thilikos, Michalis Vazirgiannis: D-cores: Measuring Collaboration of Directed Graphs Based on Degeneracy. In: ICDM, 2011.
- Christos Giatsidis, Klaus Berberich, Dimitrios M. Thilikos, Michalis Vazirgiannis: Visual exploration of collaboration networks based ongraph degeneracy. In: KDD, 2012.
- Christos Giatsidis, Dimitrios M. Thilikos, Michalis Vazirgiannis: Evaluating Cooperation in Communities with the k-Core Structure. In: ASONAM, 2011.
- S.B. Seidman. Network Structure and Minimum Degree. Social Networks, 1983.

- An online demo at: **http://www.graphdegeneracy.org/**

Outline

1. Introduction & Motivation
2. Graph fundamentals
3. Community evaluation measures
4. Graph clustering algorithms
5. Clustering and community detection in directed graphs
6. Alternative Methods for Community Evaluation
7. **New directions for research in the area of graph mining**

Open Problems and Future Research Directions (1)

- ***Community detection in directed graphs***
  - A formal and precise definition of the clustering/community detection problem in directed networks (how clusters should look like)
  - In the existing methods on directed networks, there is no a clear way of how the edge directionality should be taken into account
  - Not straightforward generalizations of the methods for undirected graphs
  - **Note:** a single definition/notion of communities should possibly not fit to all needs – highly application-oriented task **[Schaeffer '07]**
- Extension of existing methods to cover the case of signed graphs

Open Problems and Future Research Directions (2)

- ***New Concepts & Applications of Degeneracy in Social Graphs***
  - Reciprocity in signed graphs
    - Reciprocity is defined at node level indicating the average return rate of individual actions
    - Evaluation of reciprocity in trust networks
  - User's engagement in social graphs
    - Important for prediction and monitoring of social nets evolution
    - Each user can either *stay* or *leave* from the graph
    - Model of user engagement based on his/her core number
    - Characterization of the engagement level of the graph based on the properties of the k-core decomposition

Open Problems and Future Research Directions (3)

- ***New Concepts & Applications of Degeneracy in Social Graphs*** [1]
  - Degeneracy in text indexing and retrieval
    - Consider documents as directed graphs
    - Preserve word order and co-occurrence (tw)
    - Replace *term frequency* (*tf*) with *term weighted degree* (tw) in ad hoc retrieval
    - Potential in summarization and n-grams indexing
- ***Methodological level***
  - Defining preferential attachment model for signed graphs

[1] Rousseau F, Vazirgiannis M., "Composition of TF normalizations: new insights on scoring functions for ad hoc IR", ***ACM SIGIR 2013***

Open Problems and Future Research Directions (4)

- ***Scalability***
  - Distributed spectral clustering
    - Compute Laplacian and eigenvector decomposition in a *distributed* manner
  - Degeneracy for large scale graph clustering
    - Degeneracy identifies the cores of the best clusters
    - The degenerated data are exponentially smaller than the original one so the scheme scales
  - k-core computation O(nm)
    - Can be costly for dense graphs
    - Optimize with divide and conquer + start from high degree nodes

Open Problems and Future Research Directions (5)

- ***Clustering Validity for graph clustering***
  - How to decide if the results of graph clustering are valid ?
  - Parameter values and algorithms choice …
  - Reliable benchmark graph dataset **[Lancichinetti and Fortunato ’09]**
  - Experimental and comparative studies should be performed
- ***Towards data-driven and application-driven approaches***
  - Study the structure and properties of the graph we are interested in
  - Take into account possible structural observations that may affect the community detection task

Potential applications of degeneracy results

Social Networks

- "Which is the set of core members of a community, based on their intensive mutual collaboration"?
- "Is the Epinions trust network mostly positively trustworthy?"

Scientific corpora

- "Which is the densest community of collaboration in the DBLP citation graph in *data mining*" ?
- "Which is the densest collaboration community of Dr. X in the Arxiv citation graph ?"
- "Which is the densest collaboration group in a co-authorship graph in which Dr. X belongs to?"

Telecoms

- "Which is the most connected component of users in a telecom network based on mutual calls?"

Biology

- "Which is the most important set of proteins in a protein interaction graph?"

Acknowledgments

- DIGITEO Chair Grant (France) – M. Vazirgiannis, C. Giatsidis
- Google PhD Fellowship – F.D. Malliaros

- Visit our prototype:

http://www.graphdegeneracy.org

Thank You!! - Questions?

- **Christos Giatsidis**

  École Polytechnique, France

  giatsidis@lix.polytechnique.fr

  http://www.lix.polytechnique.fr/~giatsidis

- **Fragkiskos D. Malliaros**

  École Polytechnique, France

  fmalliaros@lix.polytechnique.fr

  http://www.lix.polytechnique.fr/~fmalliaros

- **Prof. Michalis Vazirgiannis**

  AUEB, Greece & École Polytechnique, France

  mvazirg@aueb.gr