Graph Mining II
Unsupervised learning
Clustering
Graph degeneracy

M. Vazirgiannis

Dec 2017

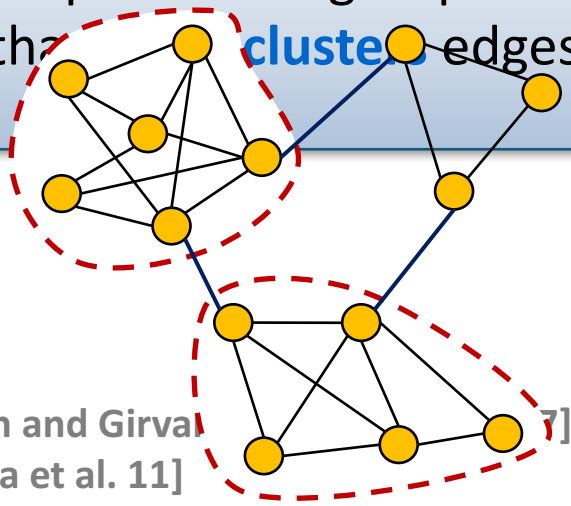
Outline

1. Introduction & Motivation
2. Graph Generators
3. Supervised Learning for graphs
    1. Graph Similarity
    2. Graph Kernels
    3. Graph classification
4. **Unsupervised learning**
    1. Community detection
    2. Applications

Community evaluation measures

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

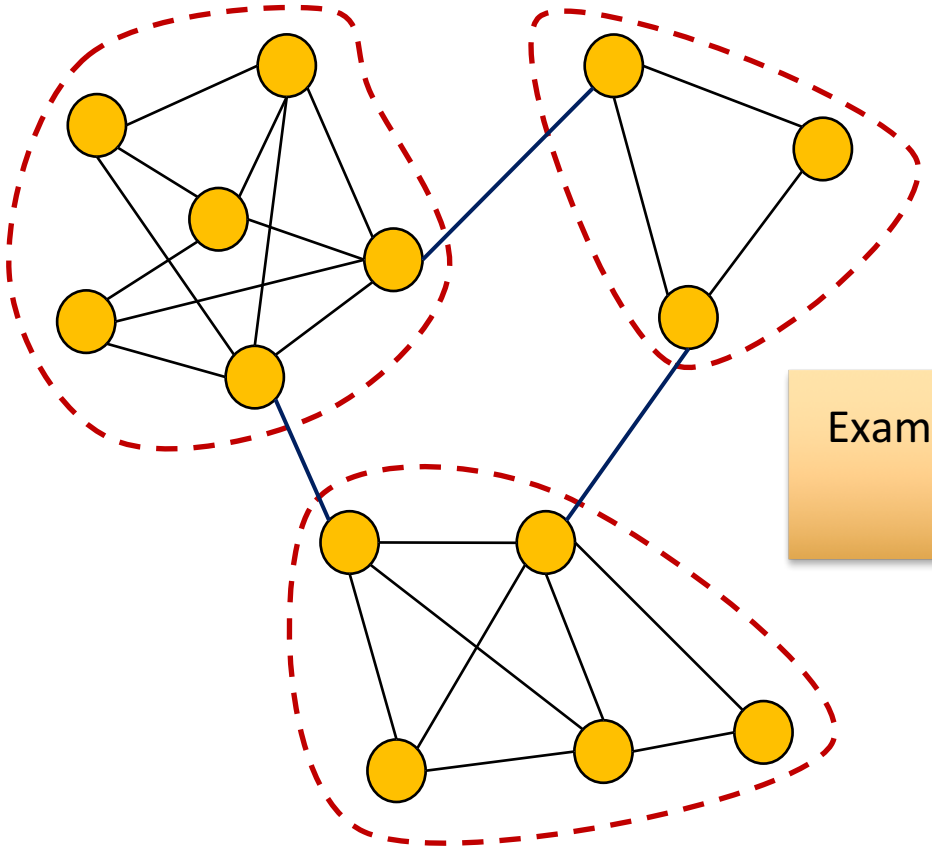
A community corresponds to a group of nodes with more **intra-cluster** edges than **cluster** edges



Example graph with three communities

[Newman '03], [Newman and Girva[illegible]7], [Fortunato '10], [Danon et al. '05], [Coscia et al. 11]

Schematic representation of communities



Example graph with three communities

Community detection in graphs

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

Community evaluation measures

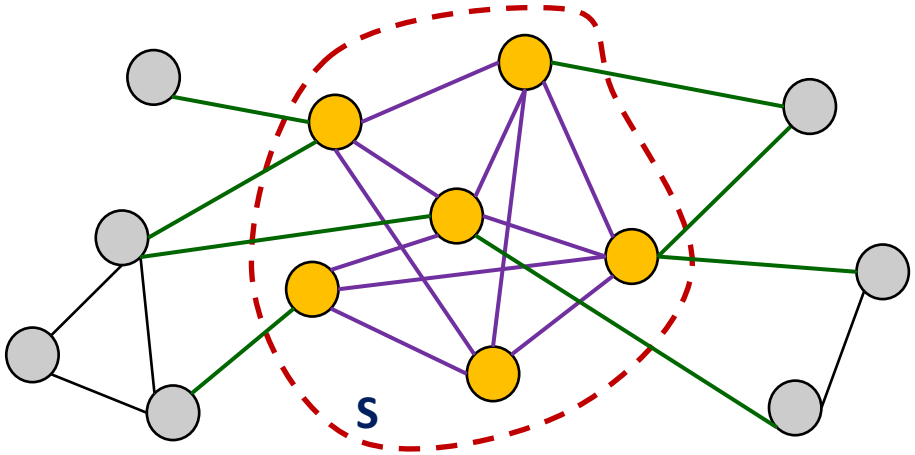
- Focus on
  - Intra-cluster edge density (# of edges within community),
  - Inter-cluster edge density (# of edges across communities)
  - Both two criteria

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

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

Notation

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



Nodes in **S** (n_S)

Edges in **S** (m_S)

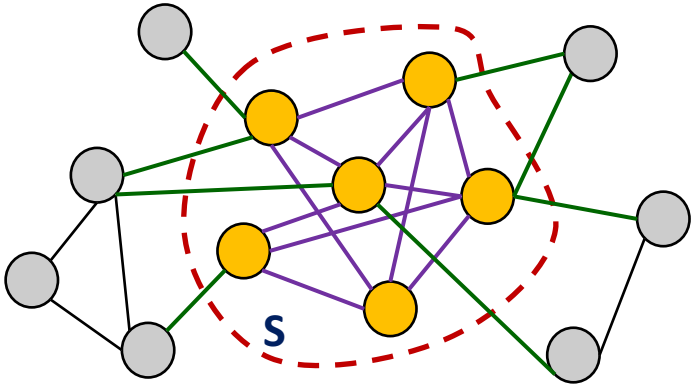
Edges in boundary of **S** (c_S)

Evaluation based on 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]

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

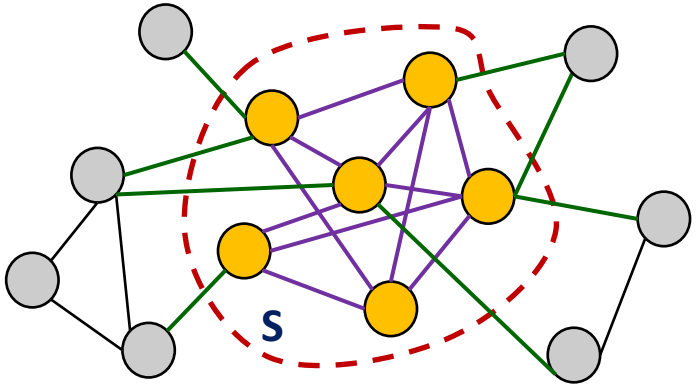
Fraction of existing edges – out of all possible edges – that leaving **S**

Evaluation based on internal 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]

$$f(S) = m_s$$

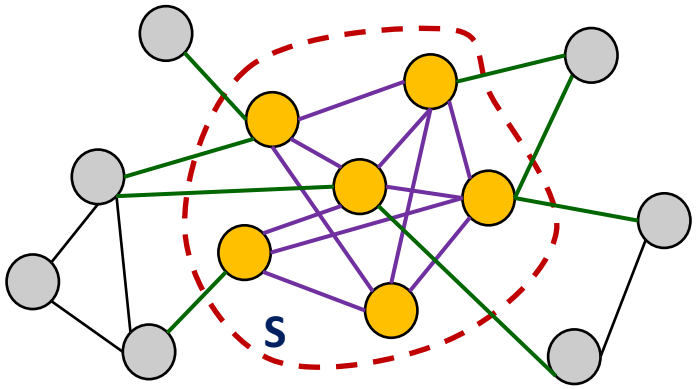
Number of edges between the nodes of **S**

Evaluation based on internal and external connectivity (2)

Conductance [Chung '97]

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

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



Normalized cut [Shi and Malic '00]

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

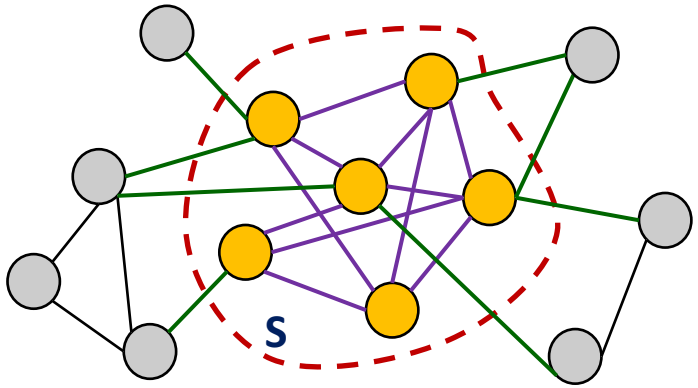
Measures the fraction of total edge volume that points outside **S** normalized by the size of **S**

Evaluation based on internal connectivity (3)

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

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

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



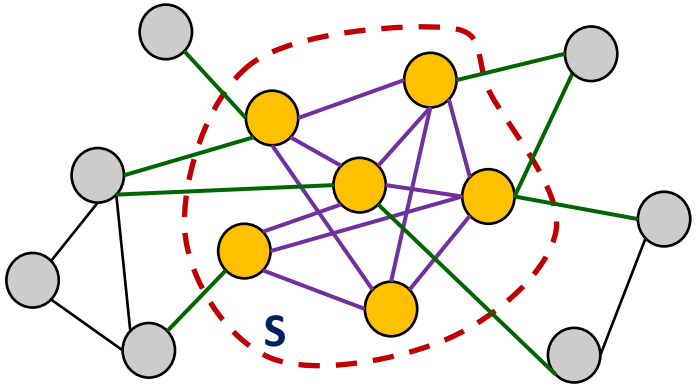
Evaluation based on network model

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

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

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

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



Graph Clustering Algorithms

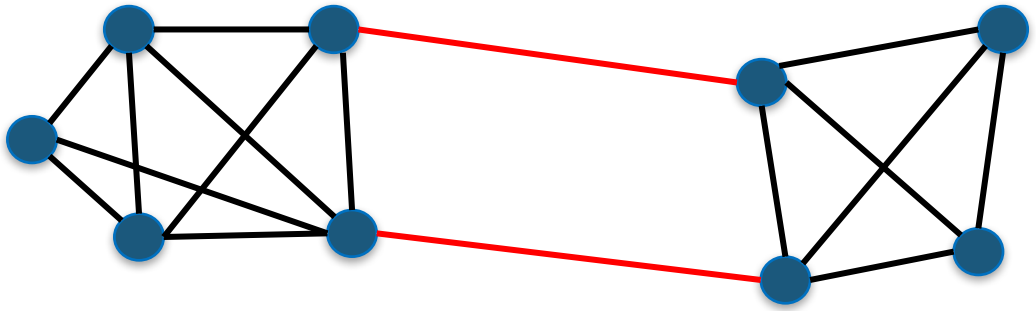
- Taxonomy
- Hierarchical methods
- **Spectral Clustering**
- Modularity Based Methods

Notations

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

Graph-Cut

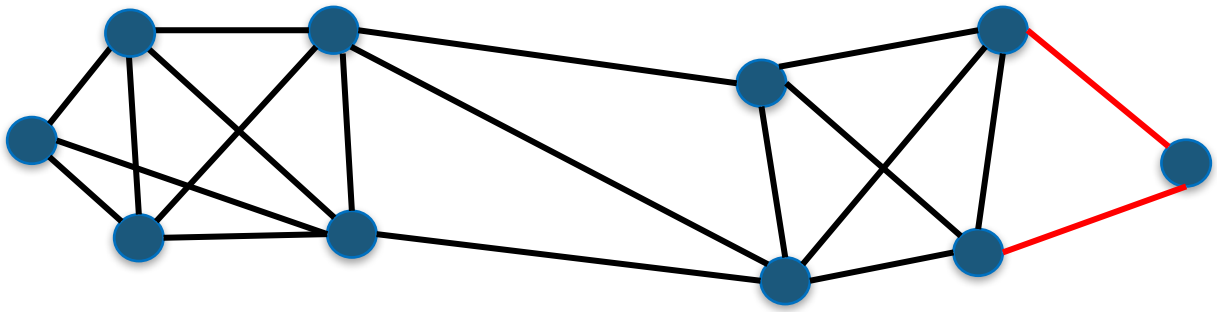
- For k clusters:
  - $cut(A_1, .., Ak) = 1/2 \sum_{i=1}^{k} w(Ai, \overline{A_i})$
    - undirected graph:1/2 we count twice each edge



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

Min-Cut

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



Normalized Graph Cuts

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

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

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

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

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

From Graph Cuts to Spectral Clustering

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

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

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

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

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

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

Graph Laplacian

- How is the previous useful in Spectral clustering?

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

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

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

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

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

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

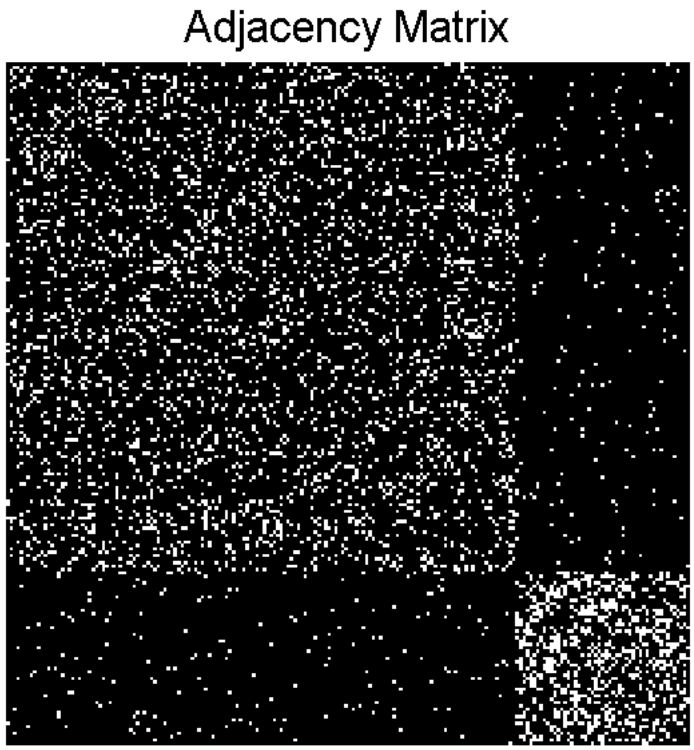
Properties of L

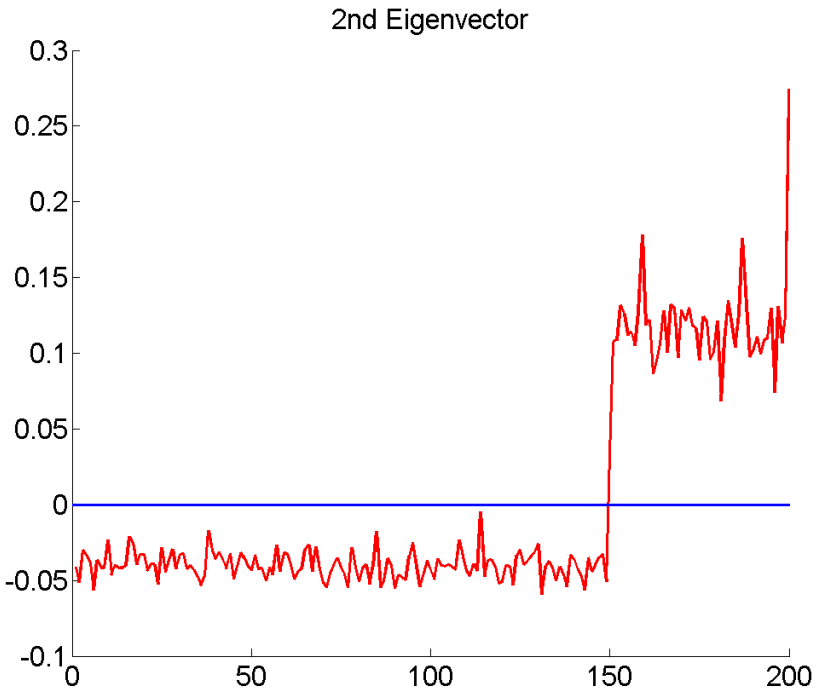
- L is
  - Symmetric
  - Positive
  - Semi-definite
- The smallest eigenvalue of L is 0
  - The corresponding eigenvector is $\mathbb{1}$
- L has n non-negative, real valued eigenvalues
  - $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

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 TRIVIA as it offers no information
- We use the second eigenvector as an approximation
  - $f_i > 0$ the vertex belongs to one cluster , $f_i < 0$ to the other

Example


Adjacency Matrix


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

Ratio Cut

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

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

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

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

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 $f: V \to \mathbb{R}$ for Graph G :

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

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

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

Normalized Cut

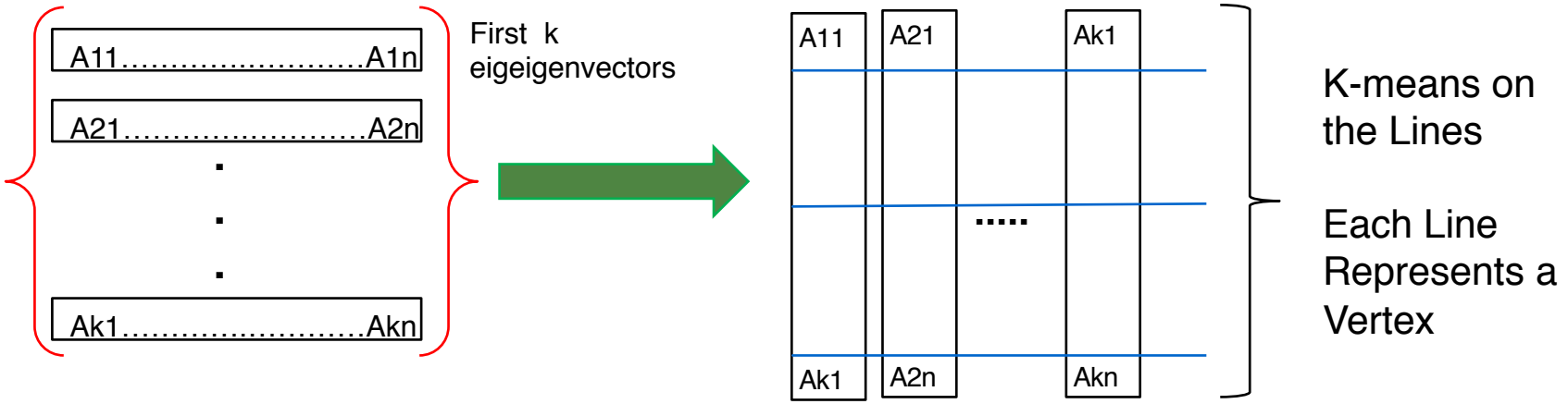
- Similarly we come to : $min_f f^T L f$
  subject to $f^T D 1 = 0,\ fTDf = v(V)$
- Assume $h = D^{1/2} f$
  - $min_h h^T D^{-1/2} \mathrm{L} D^{-1/2} \mathrm{h}$ subject to
    $h^T D^{1/2} 1 = 0, \qquad h^T h = v(V)$
  - The answer is in the eigenvector of the second smallest eigenvalue of $L_{sym} = D^{-1/2} \mathrm{L} D^{-1/2}$
    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:
  - $f_i{}^T L f_i = cut(Ai, \overline{A_i})/|Ai|$
  - $\sum_{i=1}^{k} f_i{}^T L f_i = \sum_{i=1}^{k} (F^T LF)_{ii} = Tr(F^T LF)$
    - Where Tr is the Trace of a Matrix
- So now the RatioCut becomes:
  $min(F^T LF)\ subject\ to\ FTF = I$

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 $u_1, \ldots, u_k$ 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 $y_i = 1, \ldots . n \in \mathbb{R}^k$ with the k-means algorithm into clusters C1, . . . ,Ck.

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

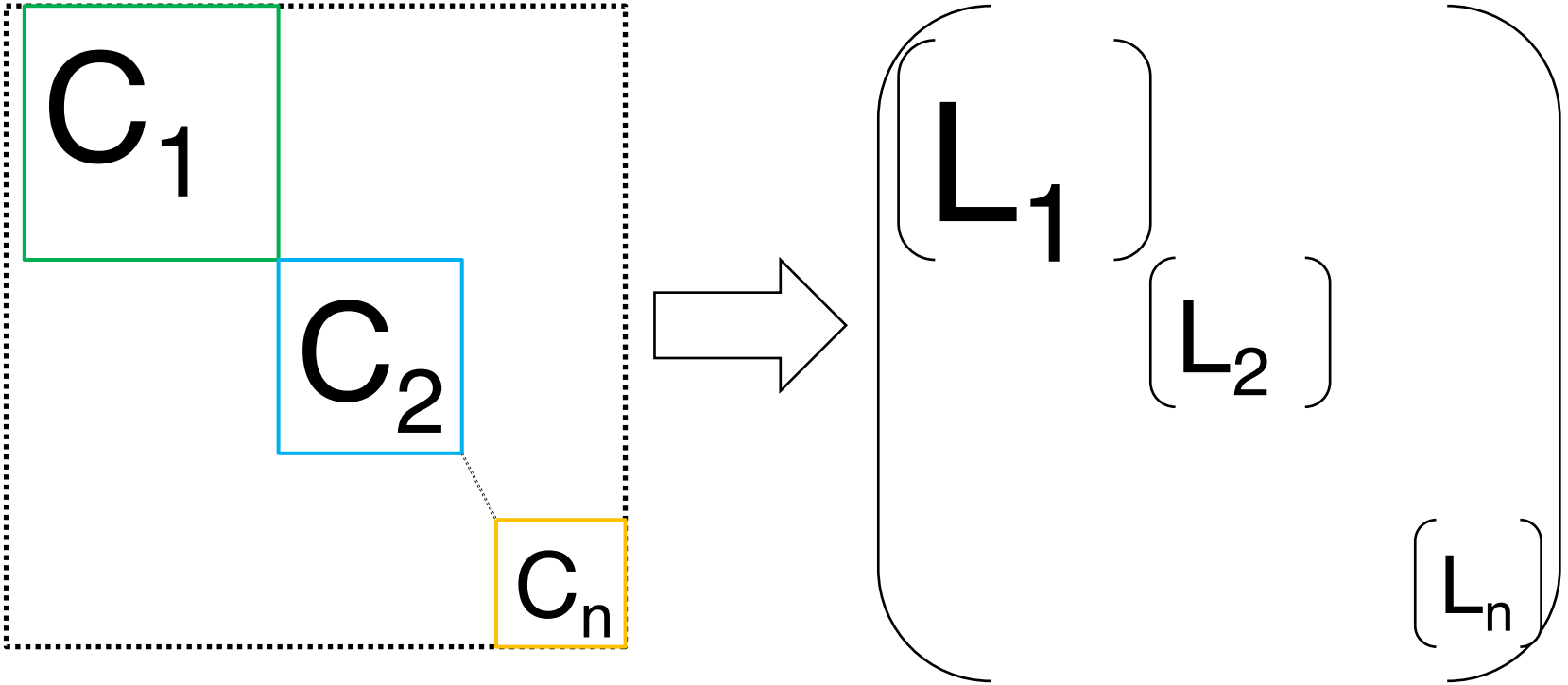
HOW DO WE CHOOSE k?

- 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

- UIrike von Luxburg, A Tutorial on Spectral Clustering, Statistics and Computing, 2007
- Davis, C., W. M. Kahan (March 1970). The rotation of eigenvectors by a perturbation. III. SIAM J. Numerical Analysis 7
- Shi, Jianbo, and Jitendra Malik. "Normalized cuts and image segmentation."*Pattern Analysis and Machine Intelligence, IEEE Transactions on* (2000).
- Mechthild Stoer and Frank Wagner. 1997. A simple min-cut algorithm. *J. ACM*
- Ng, Jordan & Weiss, K-means algorithm on the 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

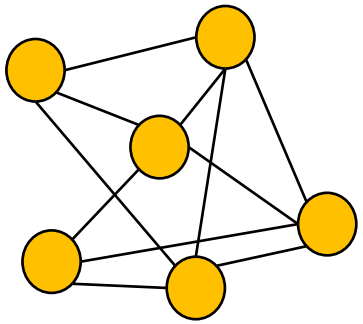
Graph Clustering Algorithms

- **Modularity Based Methods**

Main idea

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

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



Formal definition of modularity

Modularity Q

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

where

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

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

Properties of modularity

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

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

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

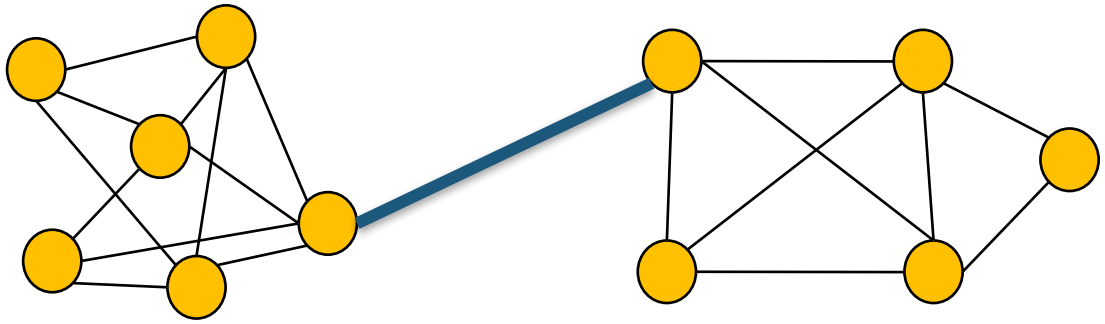
Applications of modularity

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

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

Modularity-based community detection

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



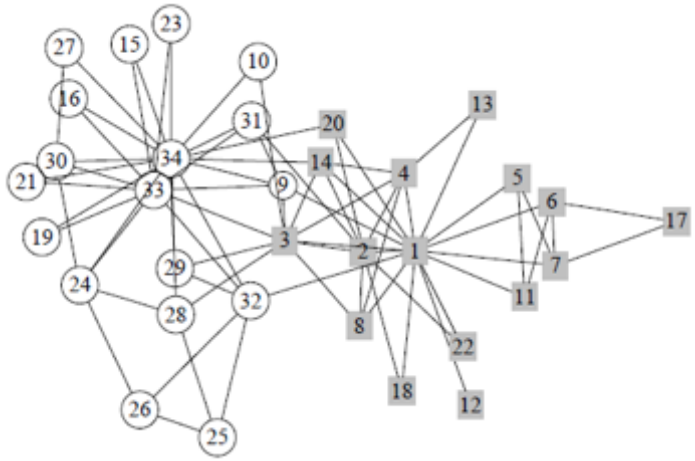
Edge betweenness is higher for edges that connect different communities

Newman-Girvan algorithm (1)

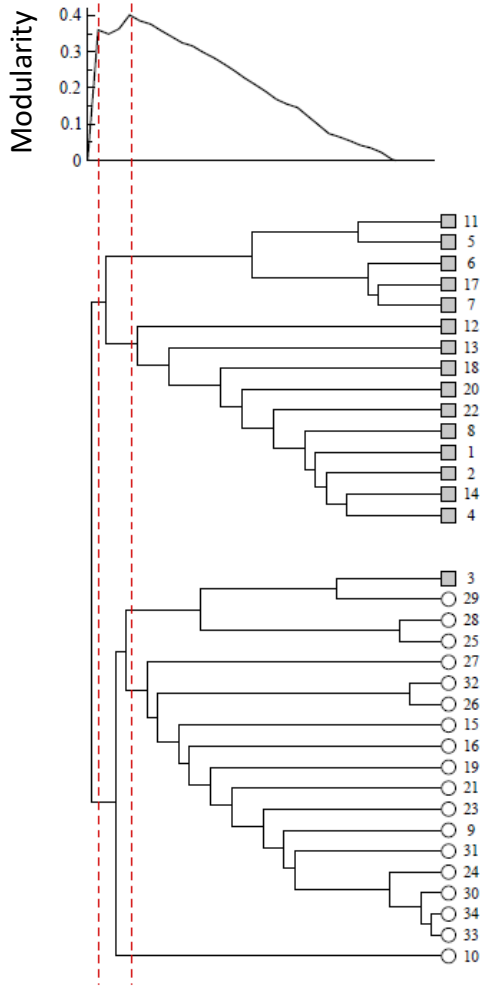
- **Basic steps:**
  1. Compute betweenness centrality for all edges in the graph
  2. Find and remove the edge with the highest score
  3. Recalculate betweenness centrality score for the remaining edges
  4. Go to step 2
- How do we know if the produced communities are **good ones** and stop the algorithm?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

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

Newman-Girvan algorithm (2)



Zachary's karate club



Community structure

[Newman and Girvan '04]

Modularity optimization

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

[Fortunato '10]

Spectral optimization (1)

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

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

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

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

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

[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_i of the modularity matrix **B**

where $a_i = u_i^T s$

$$s = \sum_i a_i u_i$$

- Modularity can now expressed as

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

Where β_i is the eigenvalue of **B** corresponding to eigenvector u_i

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

Spectral optimization (3)

- Spectral modularity optimization algorithm

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

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

- More than two partitions?

1. **Iteratively**, divide the produced partitions into two parts
2. If at any step the split does not contribute to the modularity, leave the corresponding subgraph as is
3. End when the entire graph has been splinted into no further divisible subgraphs

- Complexity: **$O(n^2 \log n)$** for sparse graphs

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

References – Graph clustering

- 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 embedded eigen-space, NIPS 2001
- Hagen, L. Kahng, , "New spectral methods for ratio cut partitioning and clustering," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , 1992

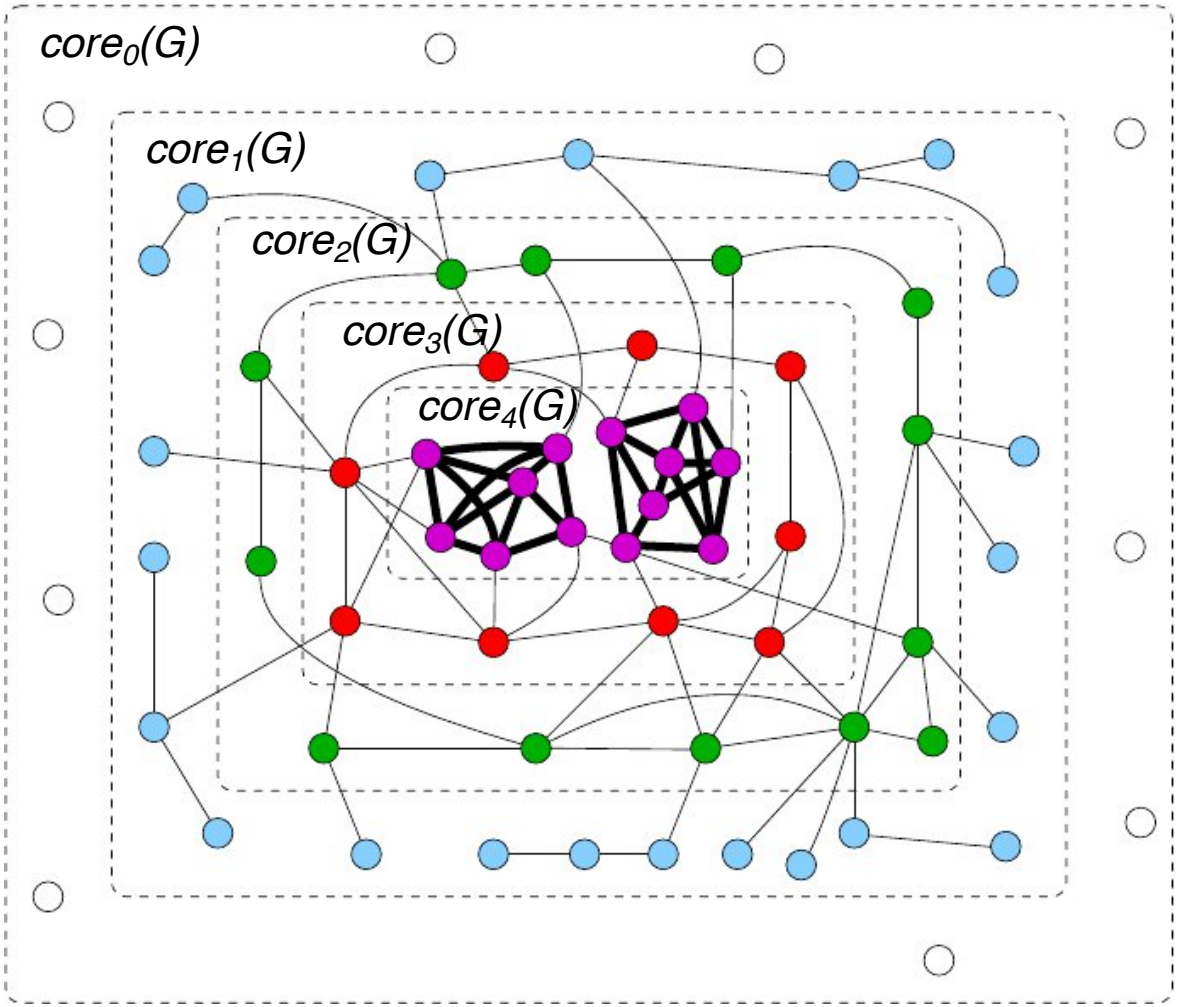
Extensions of modularity

- 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]**

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

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.

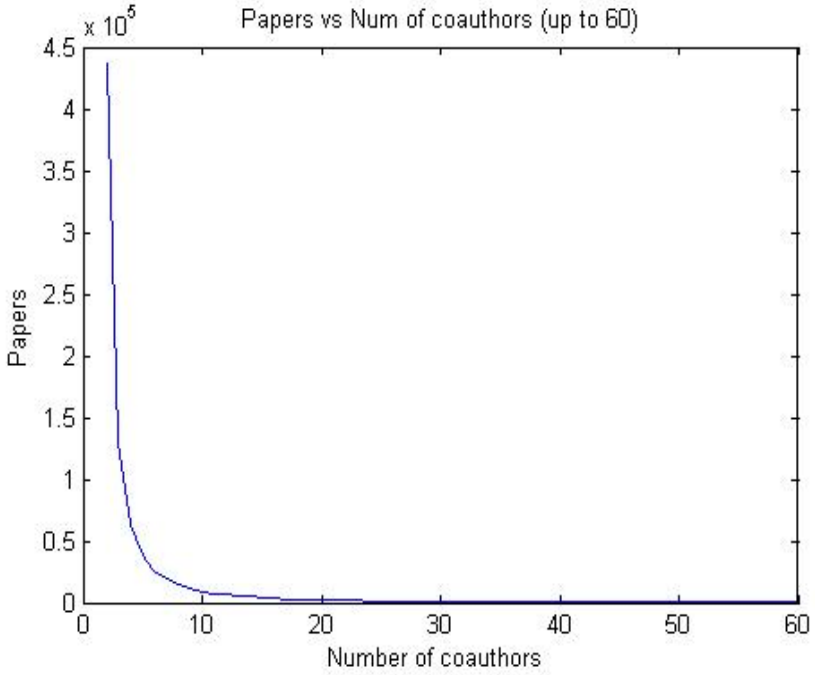
  *2. **while** there is a node x in F such that **deg**$_F$(x)<k*

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

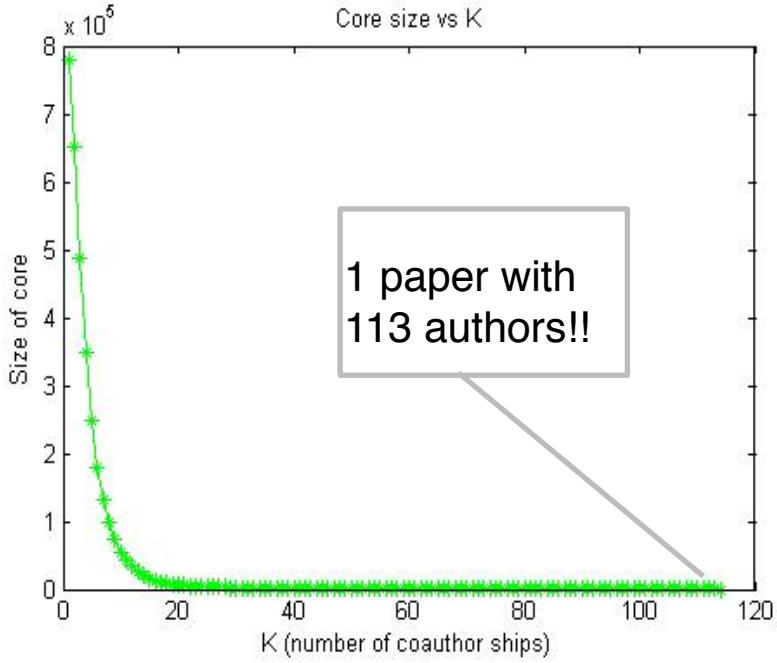
  *3. **return** F.*

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

k-cores for the DBLP coauthorship graph



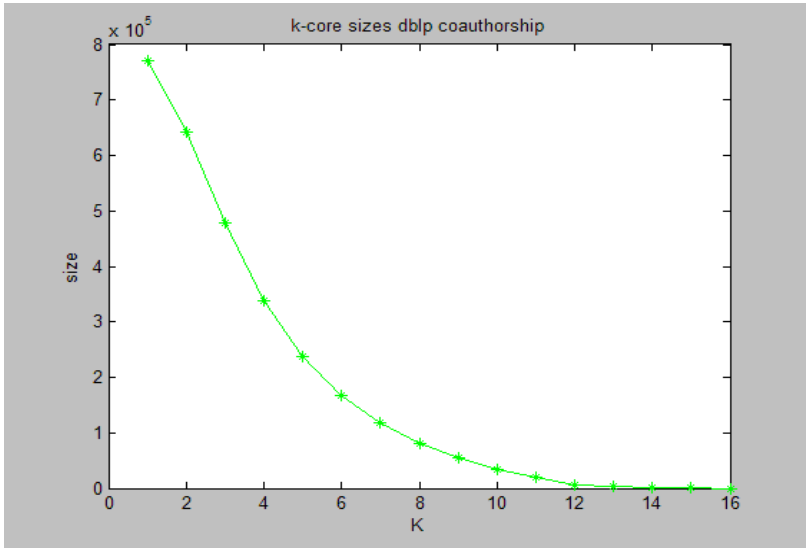
Distribution of the number of coauthors/paper k-core sizes in In the unfiltered DBLP coauthorship graph



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 !*

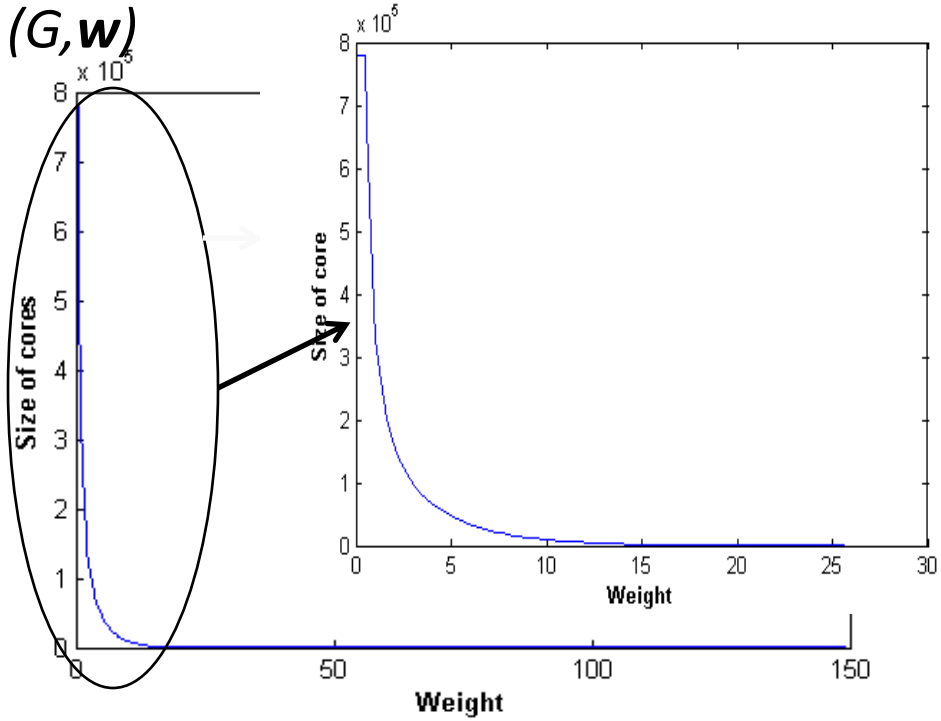
$$w(e) = \sum_{y \in N(x) \cap N(x')} \frac{1}{|N(y)|}$$

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

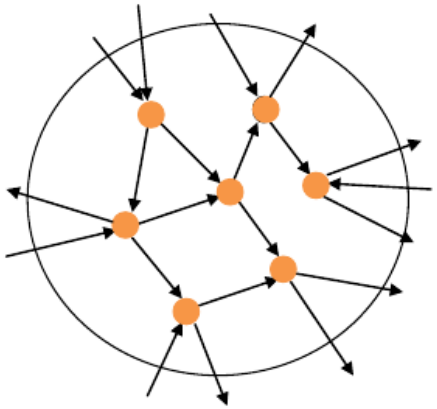
$$deg_{G,w}(x) = \sum_{e \in E(x)} w(e)$$

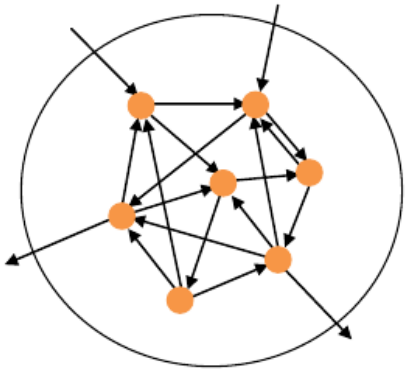
- the total co-authorship value of an author

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



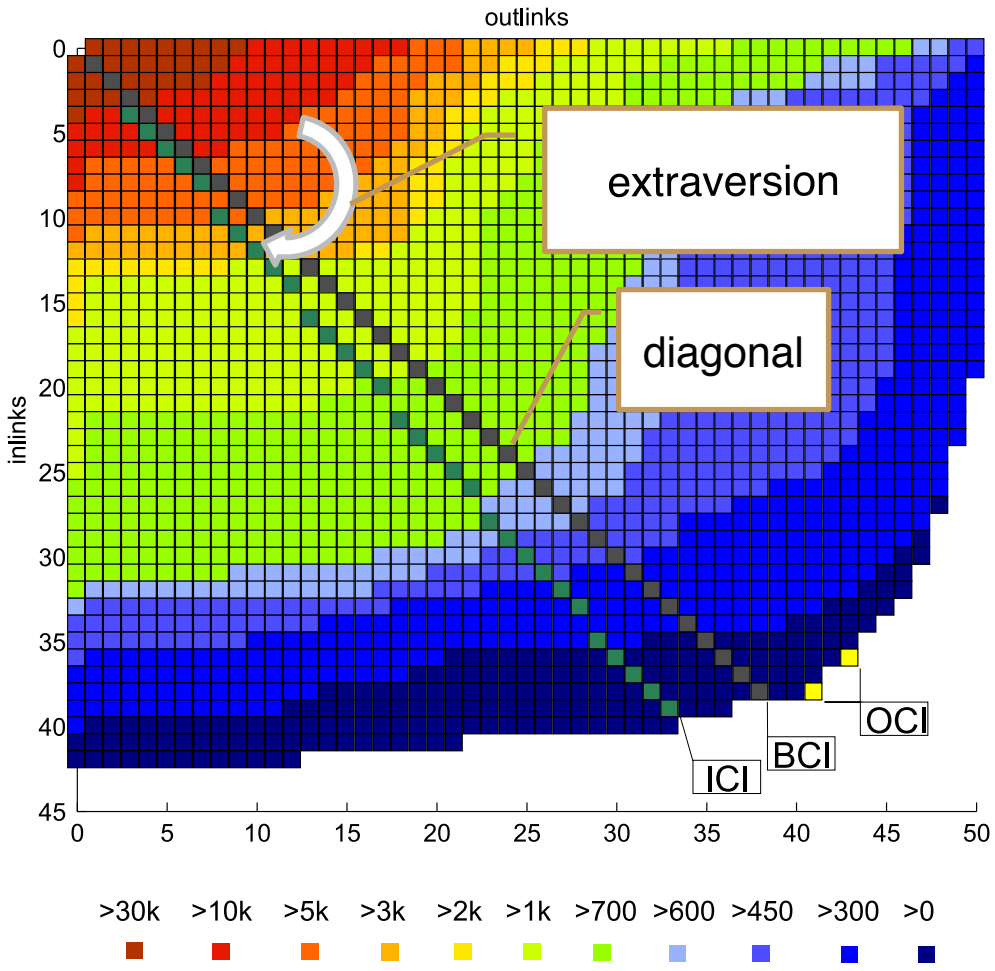
Degeneracy on directed graphs





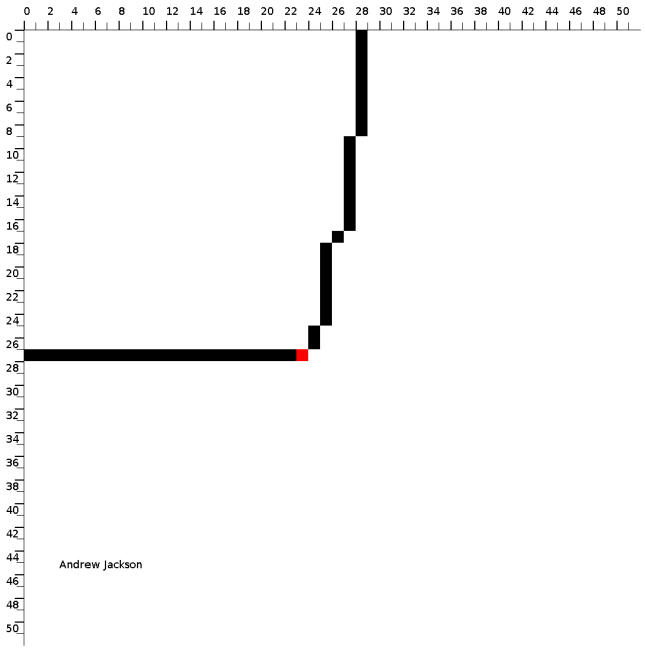
- Directed graphs:
  - WIKI - graph
  - DBLP – Citation graph
- Is there a degeneracy notion for directed graphs?
- We extend the k-core concept in directed graphs by applying a limit on in/out edges respectively.
- This provides a two dimensional range where cores degenerate.
- Trade off between in/out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

D-core matrix Wikipedia

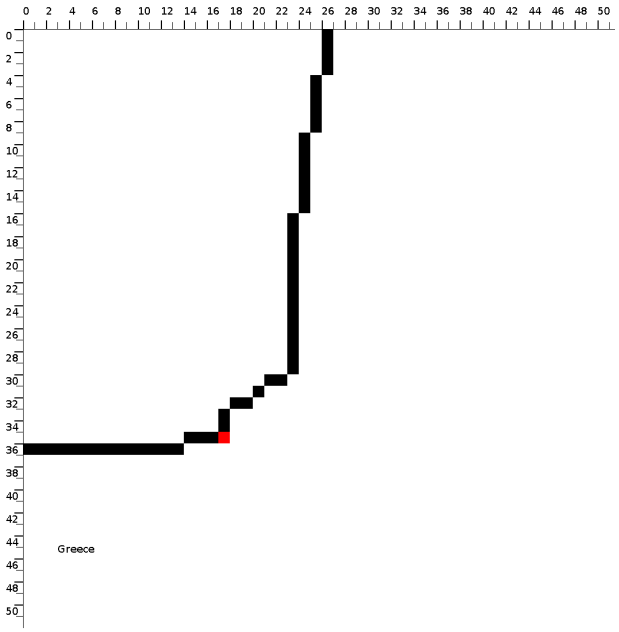


The extreme Dcore(38,41) contains 237 pages

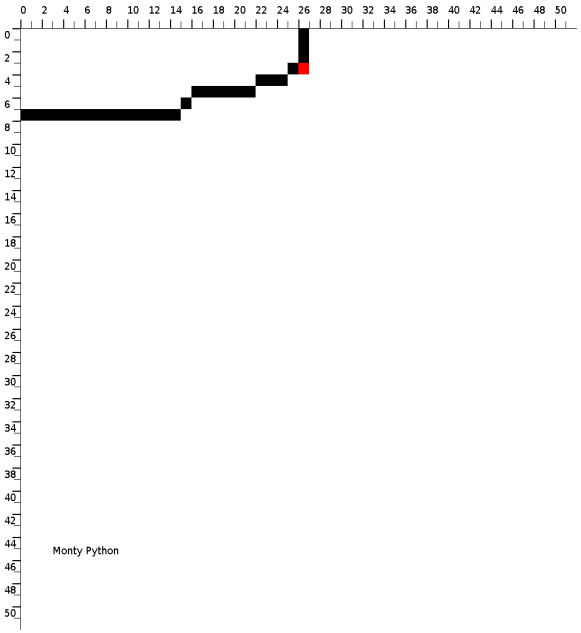
Thematic D-core frontiers - Wikipedia



“Andrew Jacson”

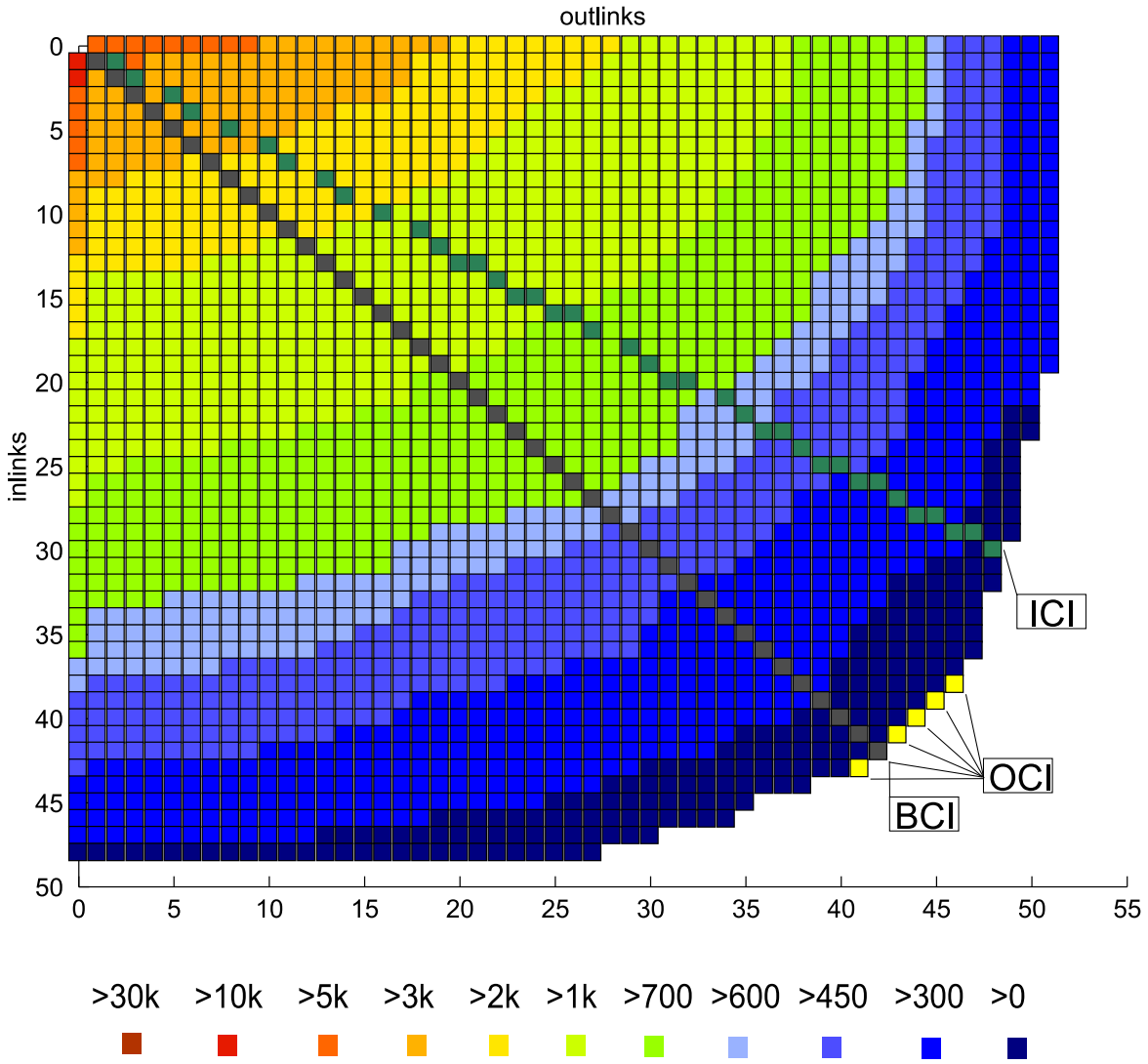


“Greece”



“Monty Pythons”

D-core matrix for DBLP

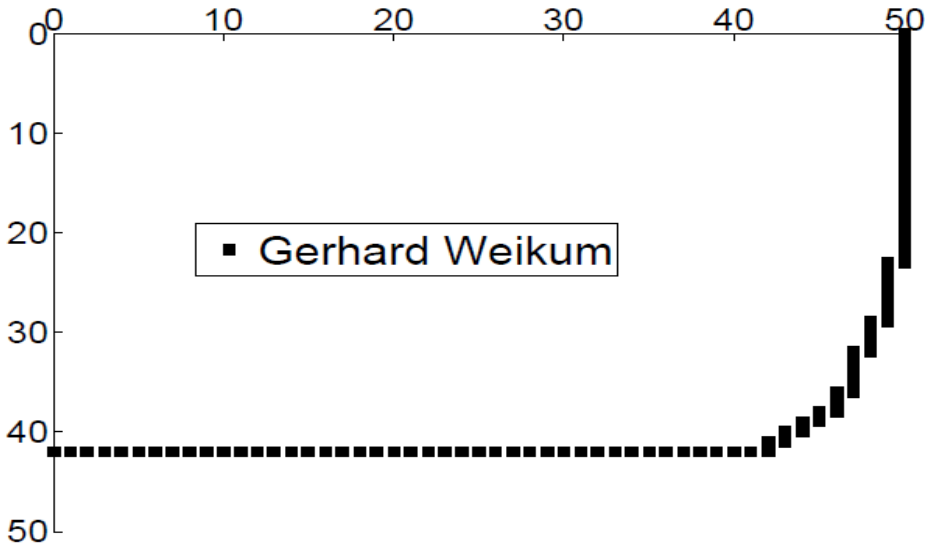


The Extreme DBLP D-core authors

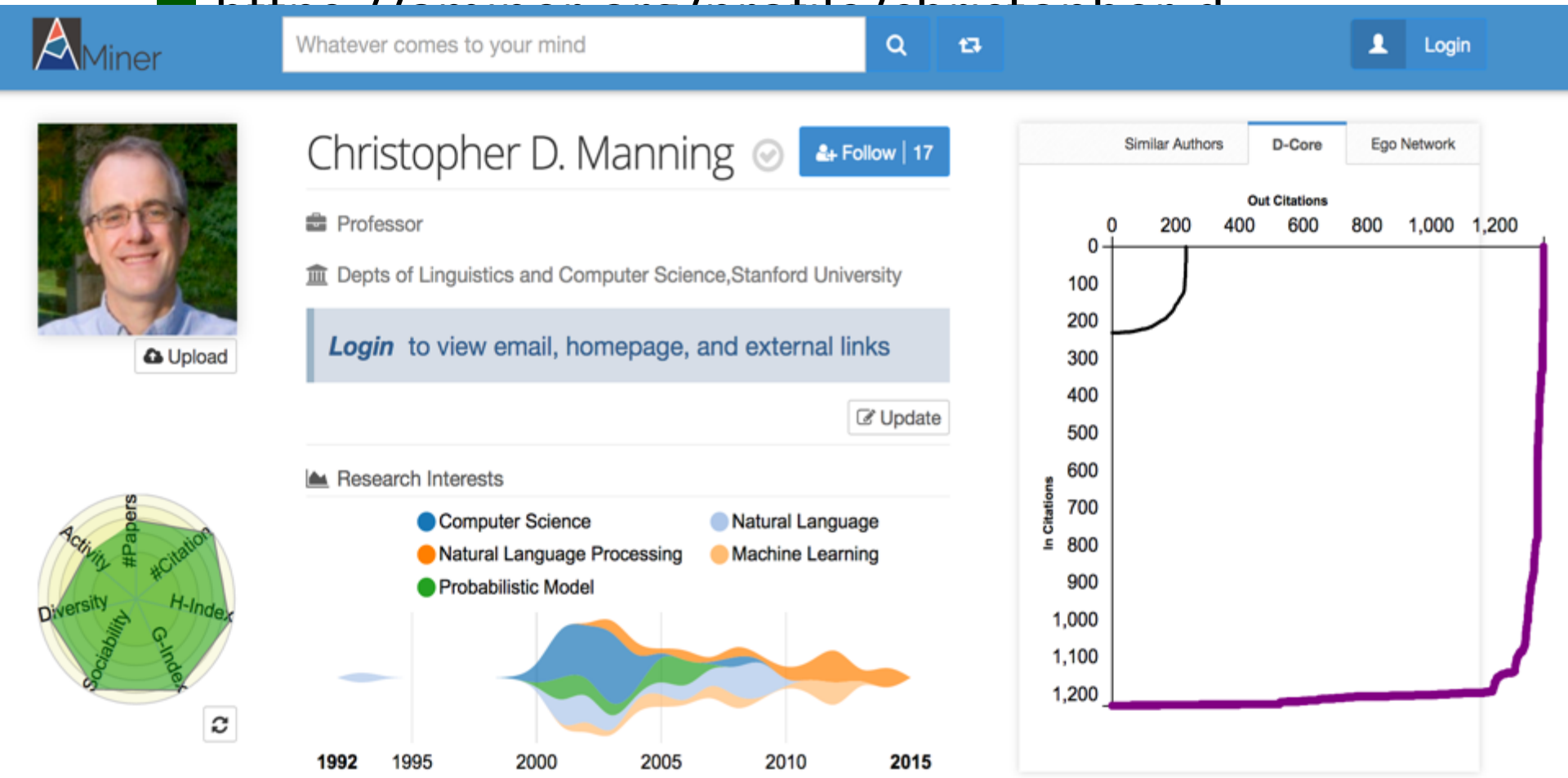
- JosÃ© A. Blakeley
- Hector Garcia-Molina
- Abraham Silberschatz
- Umeshwar Dayal
- Eric N. Hanson
- Jennifer Widom
- Klaus R. Dittrich
- Nathan Goodman
- Won Kim
- Alfons Kemper
- Guido Moerkotte
- Clement T. Yu
- M. Tamer Ã–zsu
- Amit P. Sheth
- Ming-Chien Shan
- Richard T. Snodgrass
- David Maier
- 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.



D-core adopted by aminer.org



https://aminer.org/profile/christopher-d-manning/

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 both simultaneously.
- Obviously self links are of no interest.
- Each vertex **v** has both positive & negative in-degree and both positive & negative out-degree

Quantifying trust dynamics in signed graphs, the S-Cores approachC Giatsidis, B Cautis, S Maniu, DM Thilikos, M Vazirgiannis, SIAM/ SDM, 668-674

S-cores

- Degeneracy in directed graphs has a simple 2-dimensional visualization in 2 axes.
- We could consider each case of degree (positive/negative, in/out) as a separate case and have all the combinations:
  - 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 and out degrees

S-cores

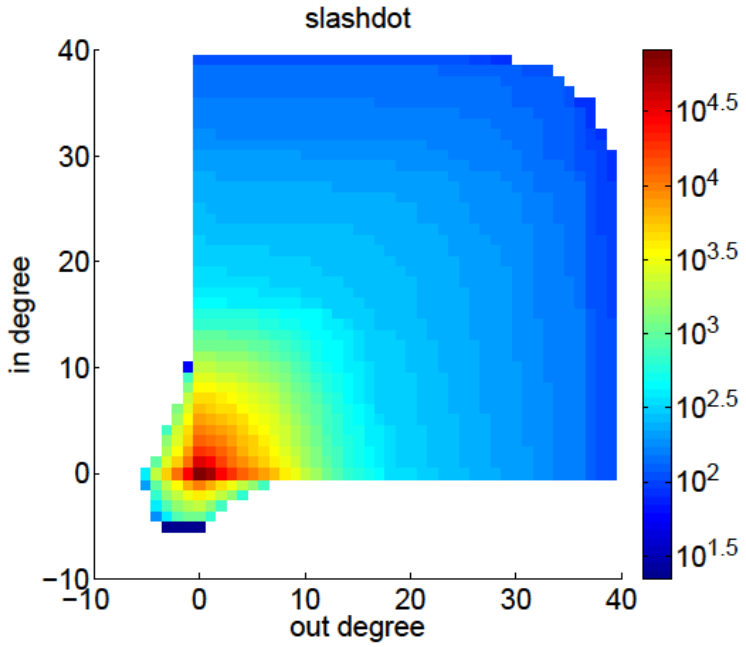
- Now we have 4 thresholds in the following combinations (in,out):
  - (+,+) Mutual Trust
  - (+,-) Trust under distrust (i.e. trust those who do not trust me)
  - (-,-) Mutual distrust
  - (-,+)Distrust under trust

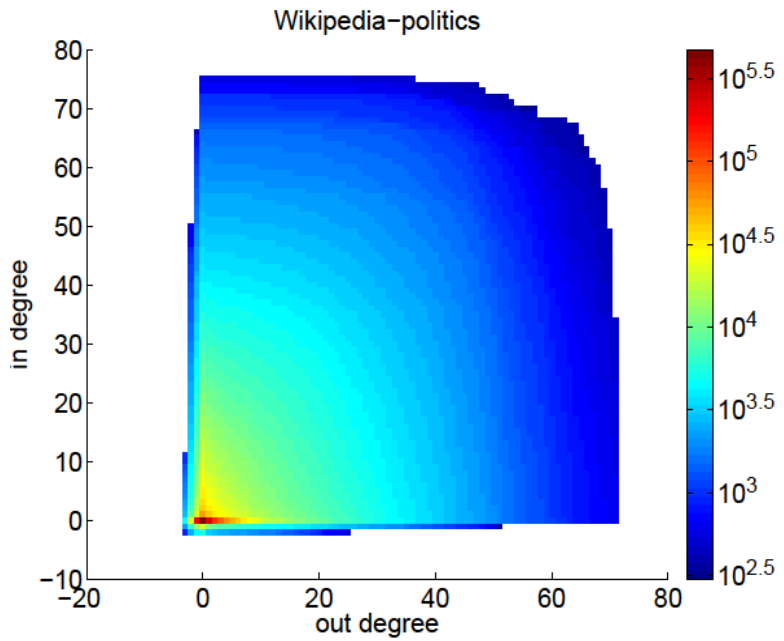
Definitions

- 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)

Examples

- S-Cores sizes on real world data:





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) and using as models existing signed trust networks.

Data Statistics

Explicit

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

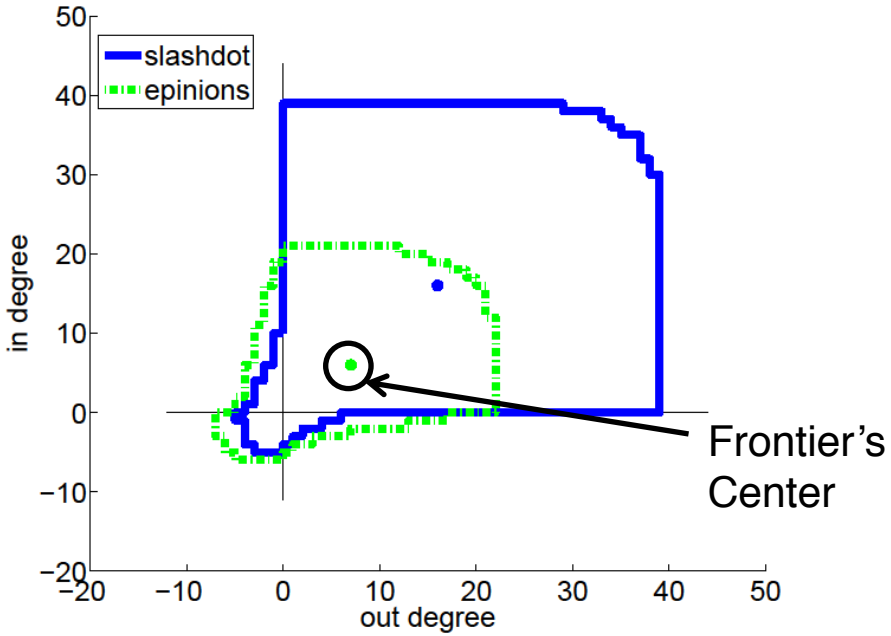
Implicit (Wikipedia)

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

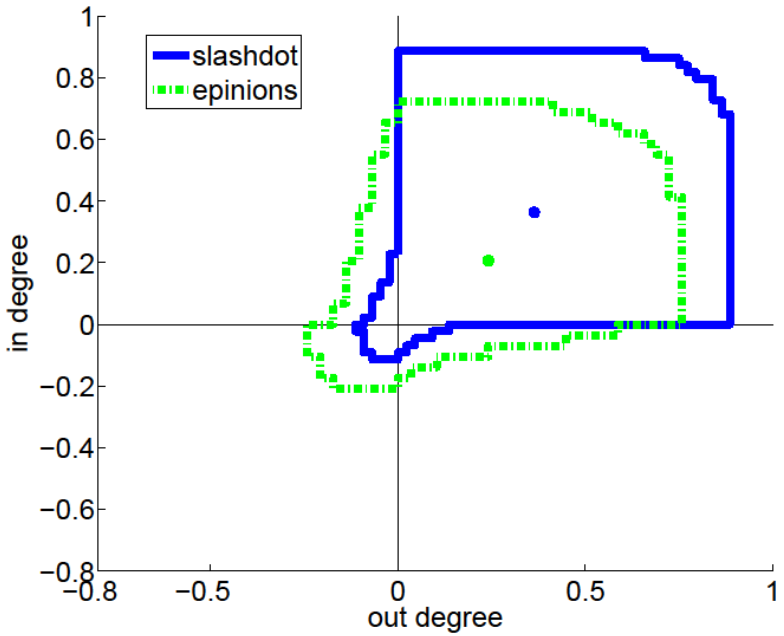
Wikipedia graph inference

- 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.

Frontiers (explicit graphs)

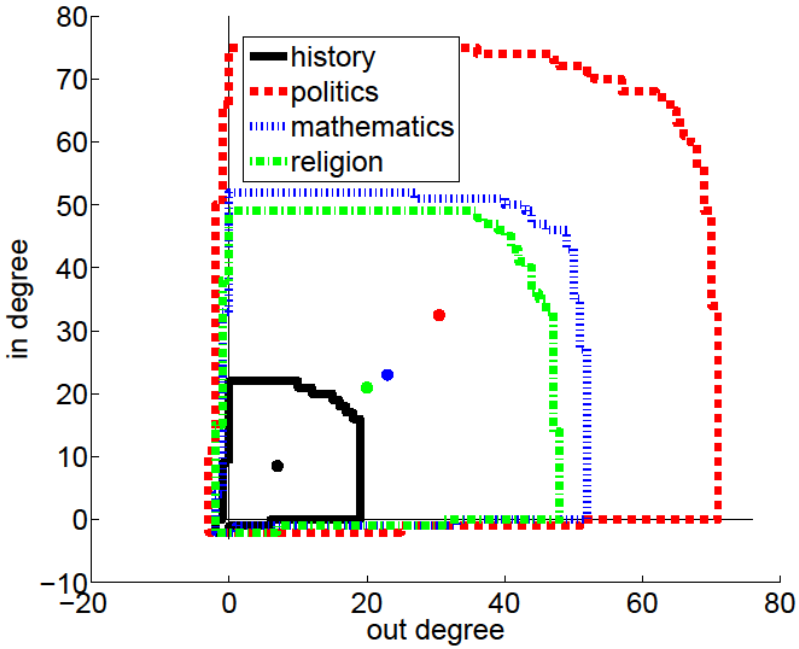


Original frontiers

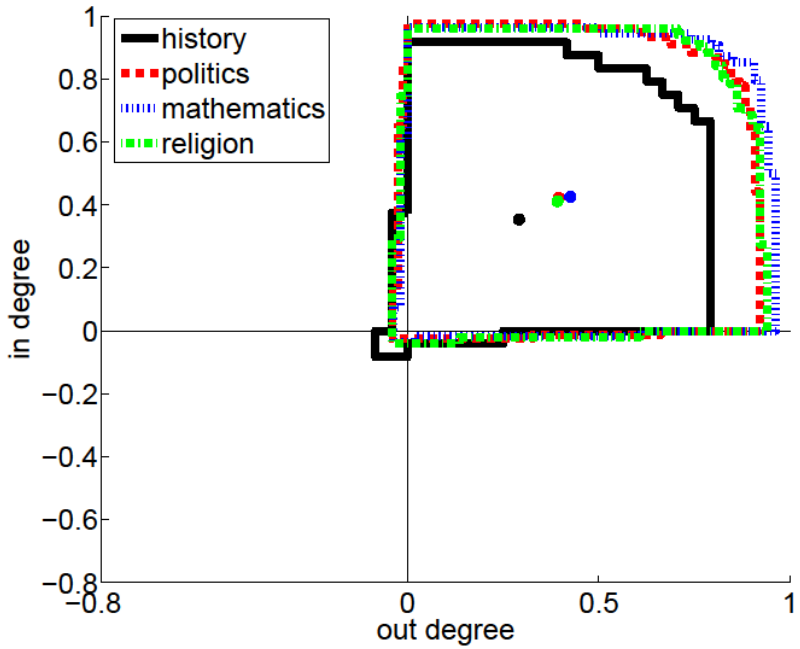


Normalized

Wikipedia Topics



Original frontiers

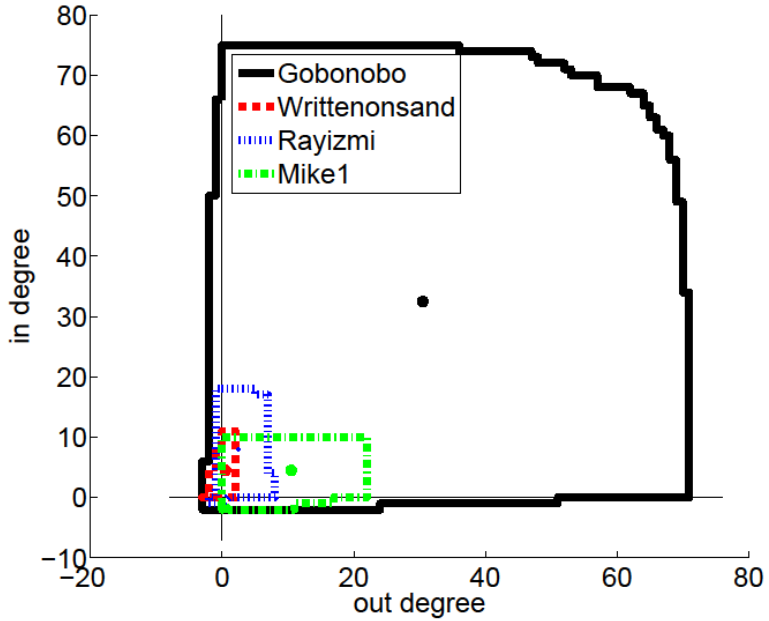


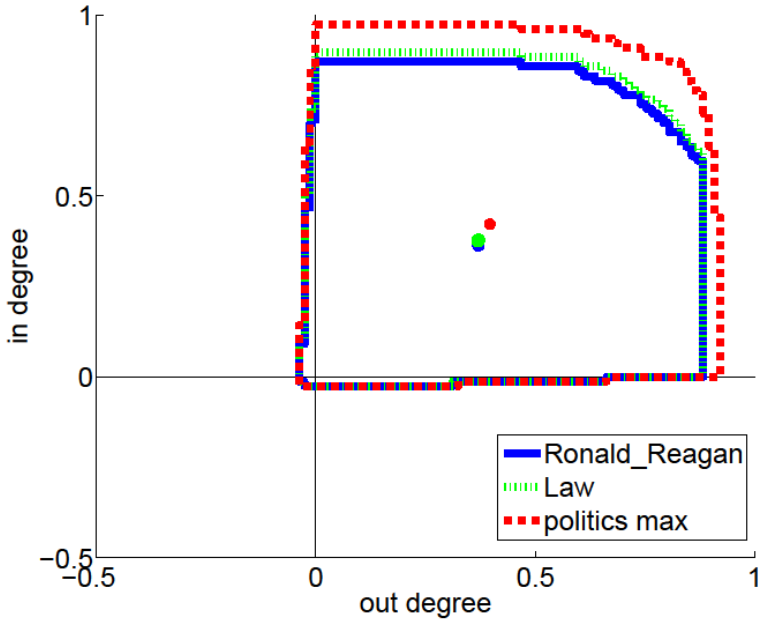
Normalized

Frontiers on Users and Articles

- We utilize the general s-core structure to extract an evaluation of the underlying elements of the graph.
- Users: A frontier can be easily defined by the membership of a user to different s-cores.
- Article-> Multiple users contribute to an Article we combine their individual frontiers to final one.

Users & Articles





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 (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 (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 (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/