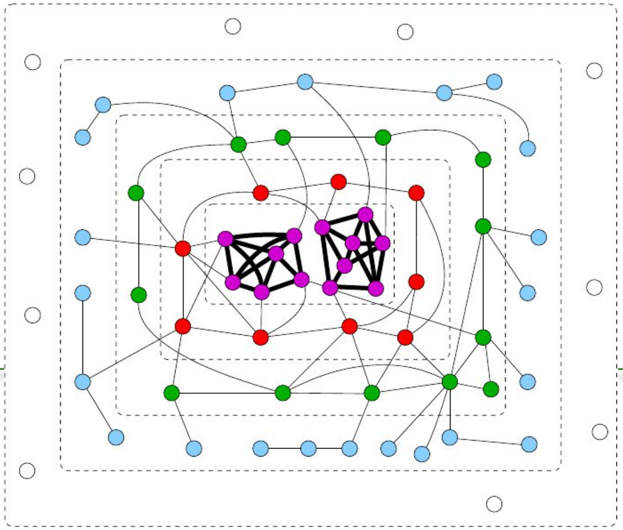
Graph Mining - graph generators & community detection



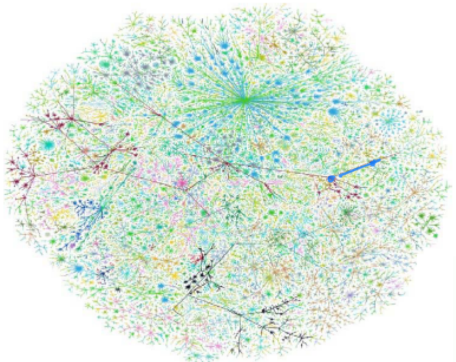
Michalis Vazirgiannis
LIX @ Ecole Polytechnique
Data Science and Mining Team (DASCIM),
LIX ÉcolePolytechnique http://www.lix.polytechnique.fr/dascim
Google Scholar: https://bit.ly/2rwmvQU
Twitter: @mvazirg

November 2020

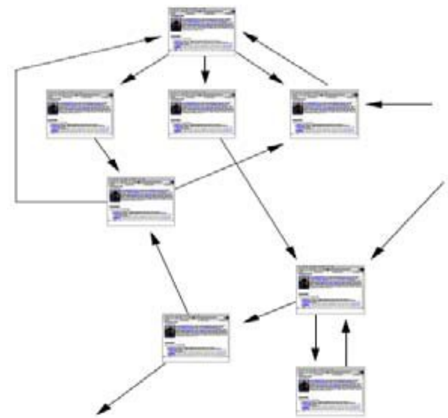
Learning for Graphs

1. **Introduction & Motivation**
2. Graph Generators
3. Unsupervised learning
   1. Community detection

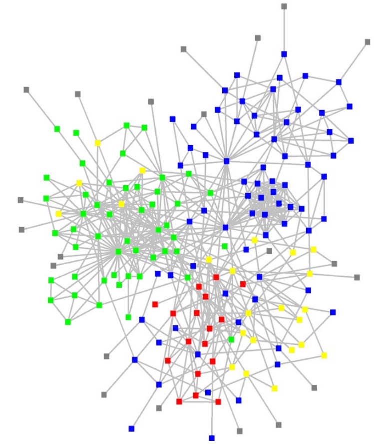
Networks are Everywhere



Internet



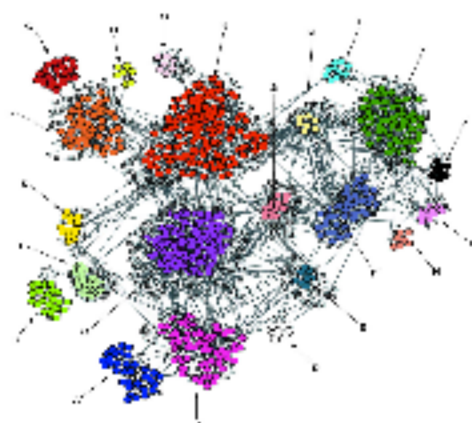
World Wide Web



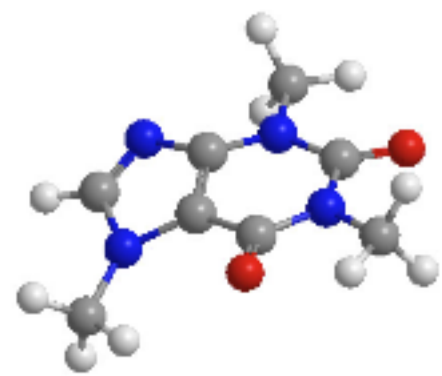
Email network

Social network

Magwene et al. Genome Biology 2004 5:R100



Co-expression network



Chemical network

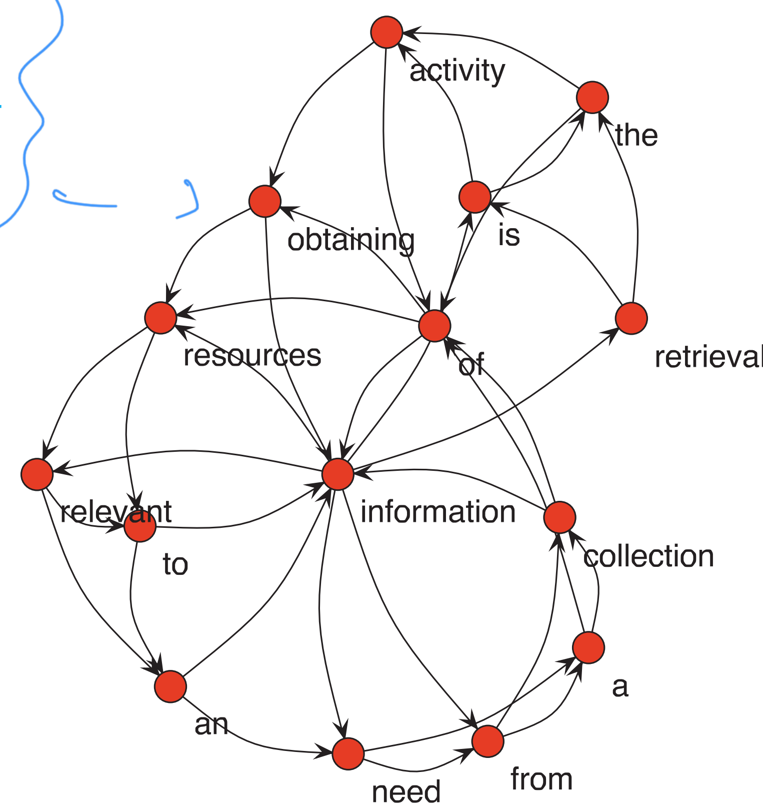
Social Networking Data

- Online social networks and social media
- Easily accessible network data at **large scale**
- Opportunity to scale up observations
- Large amounts of data raise new questions

4

Even representing text - Graph-of-word

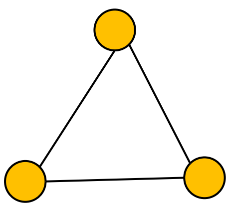
information retrieval is the activity of obtaining
information resources relevant to an information need
from a collection of information resources



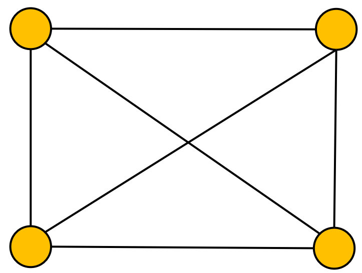
"Graph of word approach for ad-hoc information retrieval", F. Rousseau, M. Vazirgiannis,
Best paper mention award ACM CIKM 2013

Complete Graph

- **Definition:** A graph **G=(V, E)** is called complete $\mathbf{K_n}$ if every pair of nodes is connected by an edge



Complete graph with 3 nodes: triangle



Complete graph with 4 nodes

- What is the number of edges of a complete graph with n nodes?

$|V| = n$

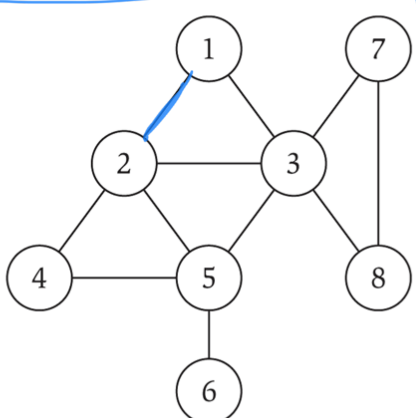
$|E|_{complete} = \frac{n(n-1)}{2} \sim O(n^2)$

- Note that, the notion of complete graphs is of particular importance for the problem of community detection
  - **Communities correspond to well-connected subgraphs**

Graph Representation: Adjacency Matrix

- A graph can be represented by the adjacency matrix **W**
  - Matrix of size **n x n**, where **n** is the number of nodes
  - $W_{ij} > 0$, if **i** and **j** are connected
  - $W_{ij} = 0$, if **i** and **j** are not connected
  - In case of unweighted graphs, $W_{ij} = 1$, if **(i, j)** is an edge of the graph
  - Space proportional to n^2

G



Undirected graph

A(i)

permutation invariant

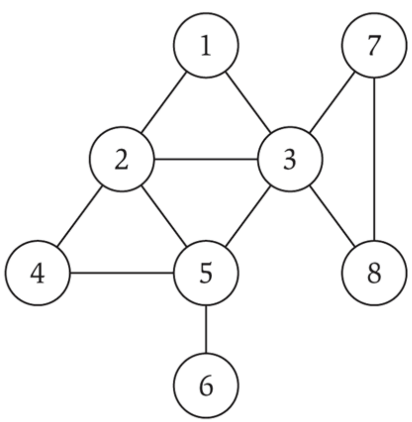
0	1	1	0	0	0	0	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	1	1
0	1	0	0	1	0	0	0
0	1	1	1	0	1	0	0
0	0	0	0	1	0	0	0
0	0	1	0	0	0	0	1
0	0	1	0	0	0	1	0

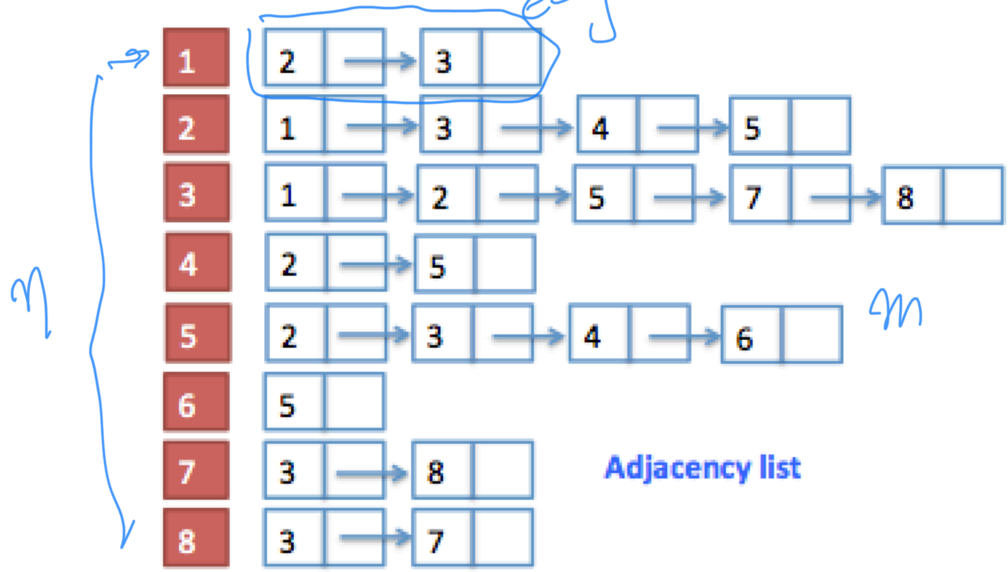
Adjacency matrix

Graph Representation: Adjacency Lists

- Adjacency lists
  - Representation of a graph with **n** nodes using an array of **n** lists of nodes
  - List **i** contains node **j** if there is an edge **(i, j)**
  - A weighted graph can be represented with a list of node/weight pairs
  - Space proportional to **$\Theta(m+n)$**
  - Checking if **(i, j)** is an edge takes **$O(d_i)$** time

edges adj's



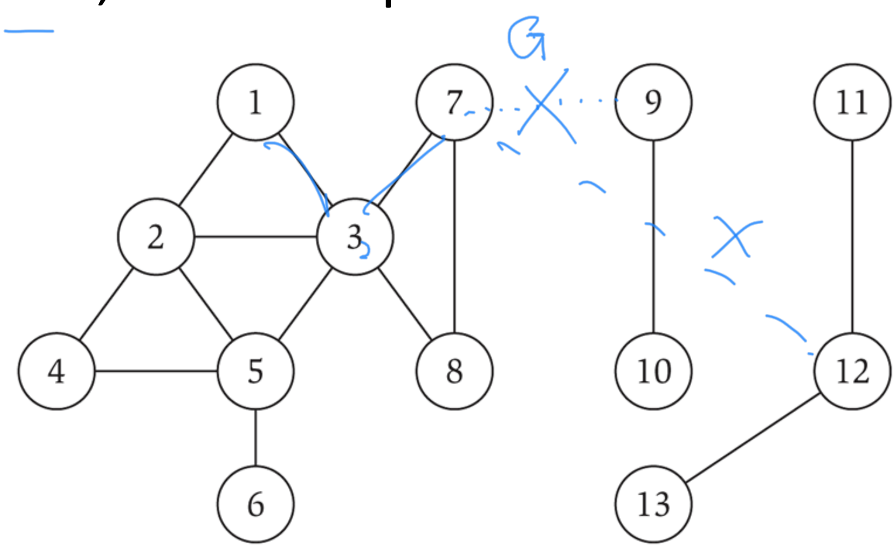


Paths and Connectivity in Graphs

- **Definition:** A path in an undirected graph **G=(V,E)** is a sequence of nodes $v_1, v_2, \ldots, v_k$ with the property that each consecutive pair v_{i-1}, v_i is joined by an edge in **E**

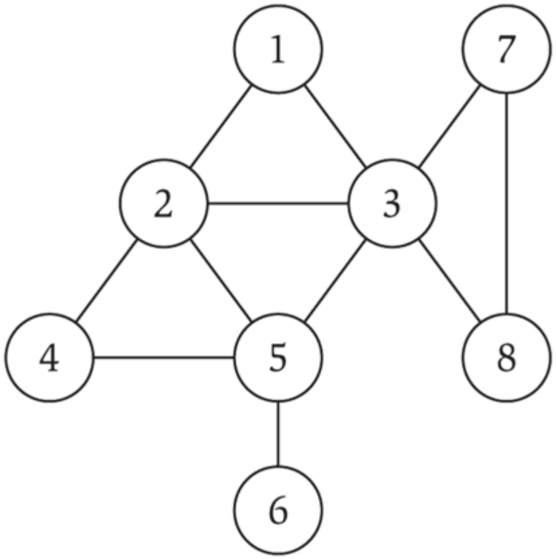
- **Definition:** An undirected graph is connected if for every pair of nodes **u** and **v**, there is a path between **u** and **v**

1 3 7 8



Cycles in Graphs

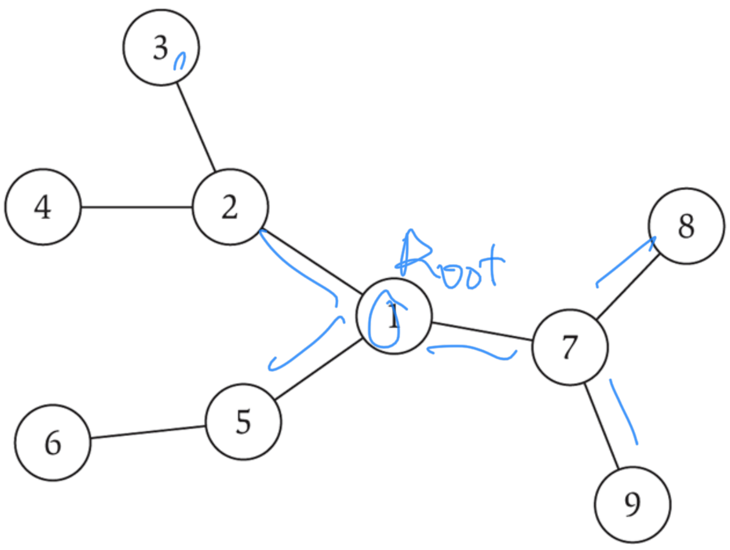
- **Definition:** A cycle is a path $v_1, v_2, \ldots, v_k$ in which $v_1 = v_k$, $k > 2$ and the first **k-1** nodes are all distinct



Cycle C = 1 – 2 – 4 – 5 – 3 – 1

Trees

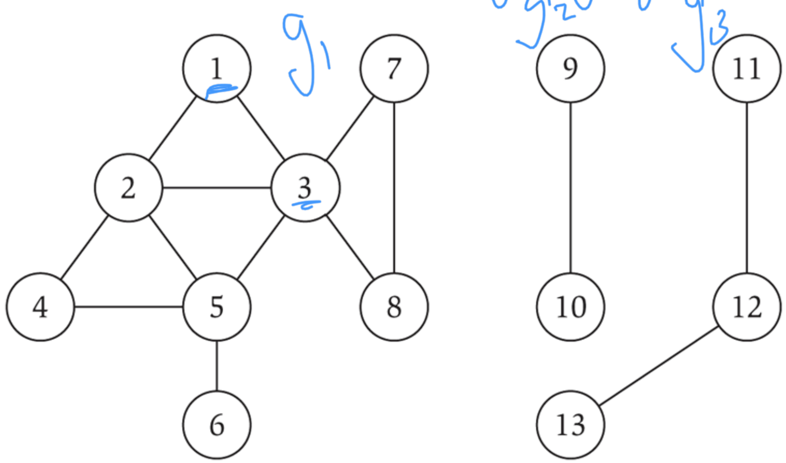
- **Definition:** An undirected graph is a tree if it is connected and does not contain a cycle
- **Theorem:** Let **G** be an undirected graph with **n** nodes. Then, any two of the following statements imply the third:
  - **G** is connected
  - **G** does not contain a cycle
  - **G** has **n-1** edges



Connected Components

- A **connected component** is a maximal connected subgraph of a graph **G** (there is a path between any pair of nodes)

$G = \{g_1, g_2, g_3\}$



Connected component containing node 1:
{1, 2, 3, 4, 5, 6, 7, 8}

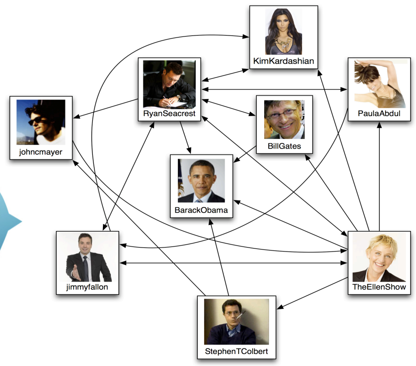
Graph with 3 connected components

Question: How can we compute the connected components of a graph?

A: Apply BFS breadth first search

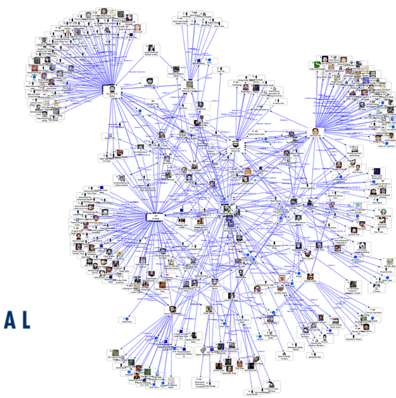
Connectivity in Directed Graphs (1/2)

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

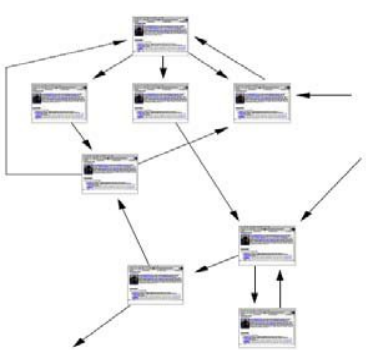


Twitter

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

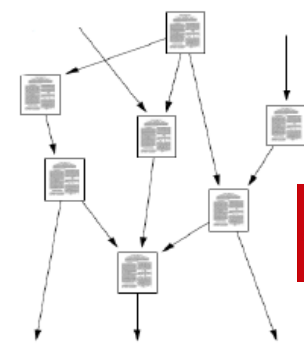


Online Social Networks



Web Graph

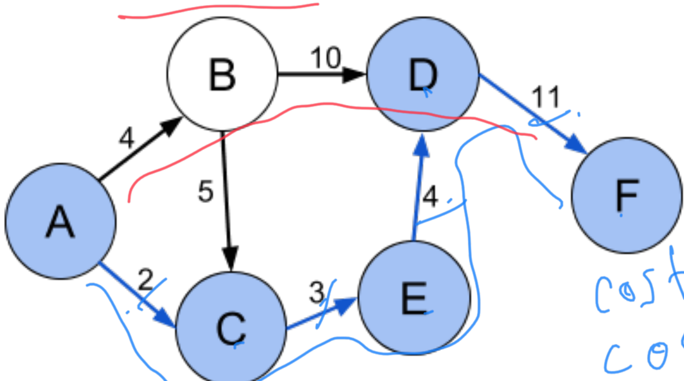
Wikipedia



Citation Graph

Shortest Paths

- **Definition:** find a path between two nodes in a graph, in such a way that the sum of the weights of its constituent edges is minimized
  - Many applications (e.g., road networks)
  - **Single-source** shortest path problem
  - **Single-destination** shortest path problem
  - **All-pairs** shortest path problem



Shortest path (A, C, E, D, F) between vertices A and F in the weighted directed graph

A → F

cost ~ = 20

cost ~ = 25

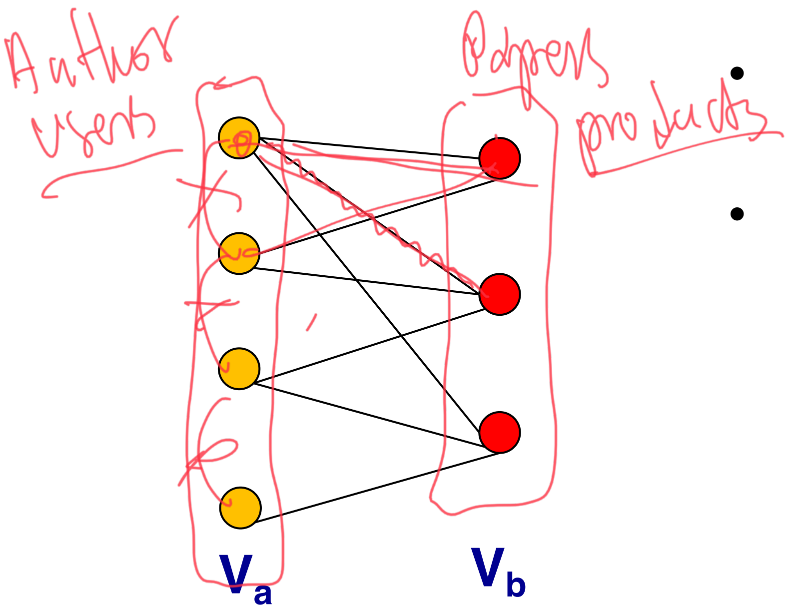
Many algorithms:

- Dijkstra
- Bellman-Ford

14

Bipartite Graphs

- **Definition:** A graph **G=(V,E)** is called **bipartite** if the node set **V** can be partitioned into two disjoint sets $\mathbf{V_a}$, $\mathbf{V_b}$ and every edge **(u,v)** connects a node of $\mathbf{V_a}$ to a node of $\mathbf{V_b}$



- Strong modeling capabilities and many real-world applications
- E.g., **Collaborative filtering** in recommender systems
  - Model the customer-product space using a bipartite graph (who-purchased-what)
  - If a user A has purchased the same product with a user B, then it is more likely to purchase another product as B did, than of a person selected randomly

15

Outline

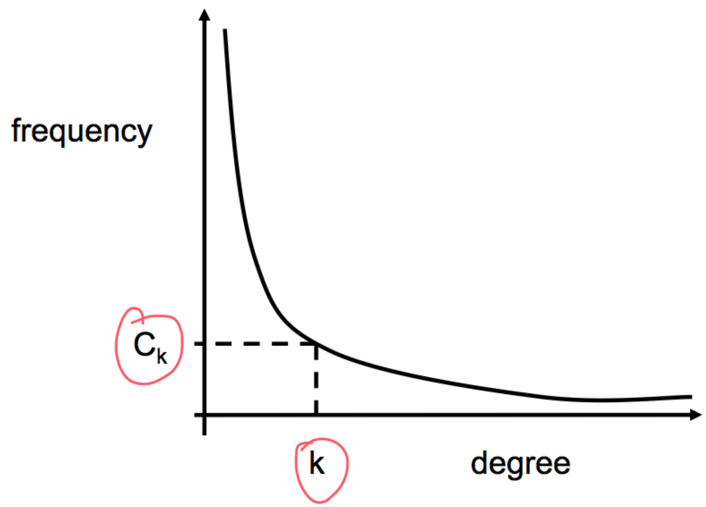
1. Introduction & Motivation
2. **Properties of real graphs**
3. Graph Generators
4. Unsupervised learning
    1. Community detection

Properties of Real-World Graph

- Networks arising from **real-world** applications obey fascinating properties
- **Static networks**
  - Heavy-tailed degree distribution
  - Small diameter
  - Giant connected component (GCC)
  - Triangle Power Law
  - Community structure
  - …
- **Dynamic networks**
  - Densification
  - Small and shrinking diameter

Degree Powel Law Distributions

- The **probability distribution** of the degrees over the network

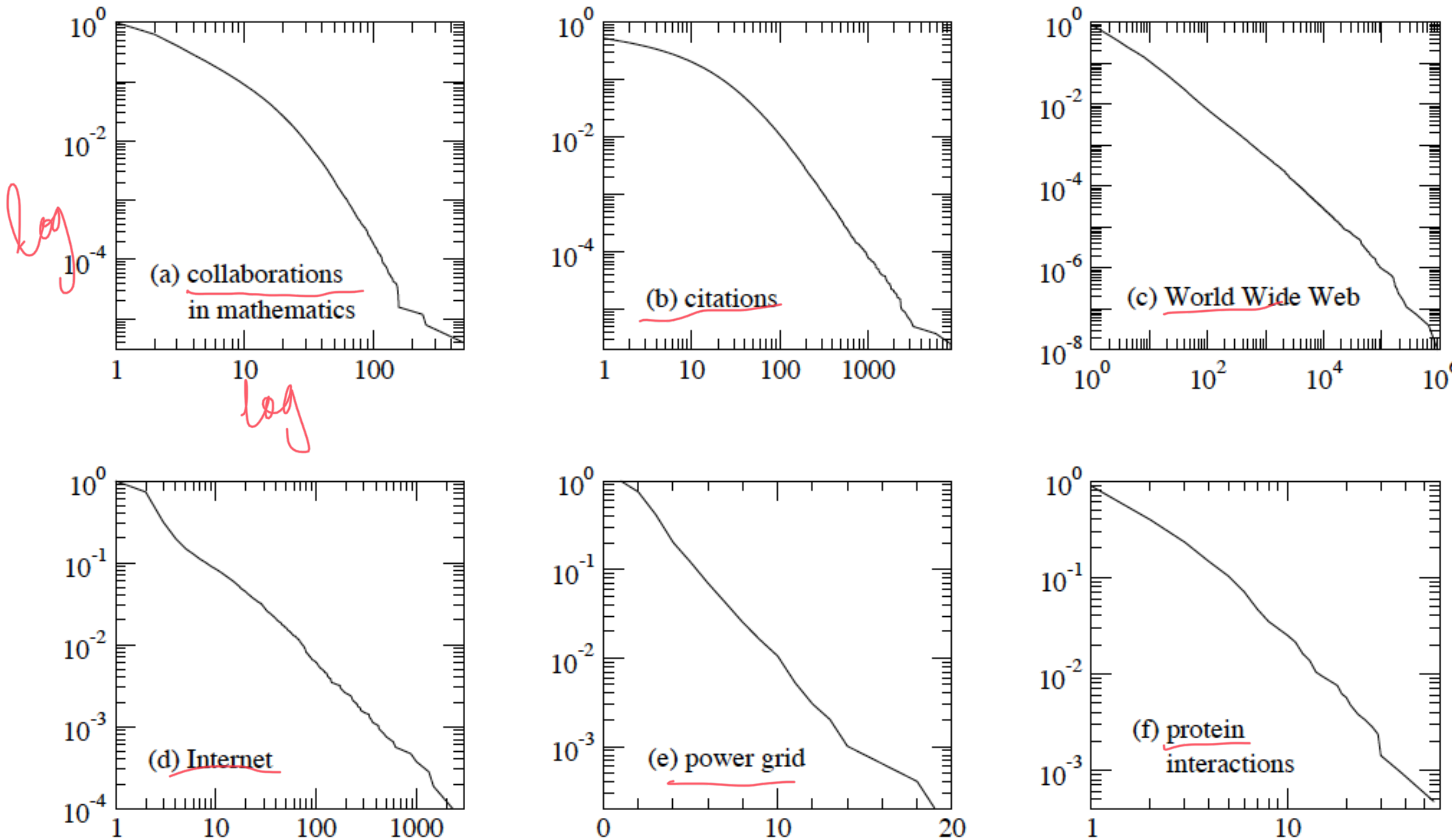


- **Problem:** find the probability distribution the **fits** best the **observed data**
- C_k = #of nodes with degree k
- $C_k = c\,k^{-\gamma}$ - $\gamma > 1$ and c a constant
- How to recognize a power-law distribution?

$$\ln C_k = \ln c - \gamma \ln k$$

  – Plotting $\ln C_k$ versus $\ln k$ gives a straight line with slope $-\gamma \ln k$

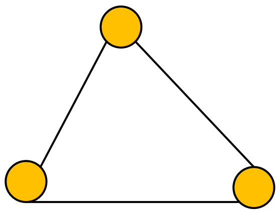
Power-law Degree Distribution in Real-Networks (2/2)



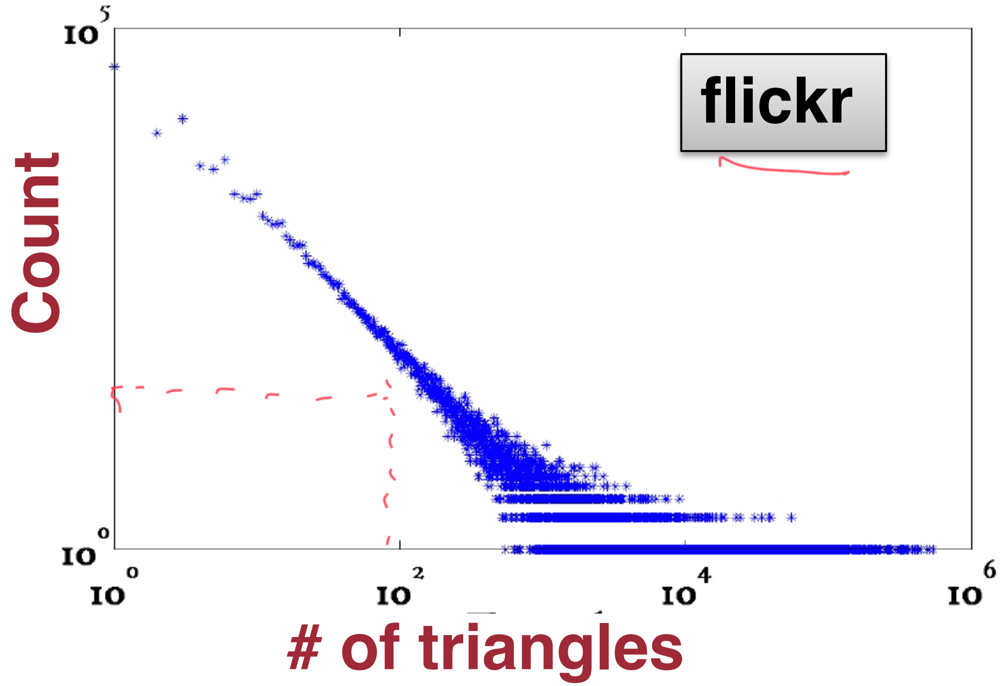
Physics

Cumulative degree distribution for six different networks [Newman 2003]

Triangle Participation Distribution



Complete graph with 3 nodes: triangle



- Number of nodes that participate in **k** triangles vs. **k** in log-log scale
- **Heavy-tailed** distribution

[Tsourakakis, 2008]

Clustering Coefficient

- Captures the tendency of the nodes of a graph to cluster together

  T(G) = 3 x # of triangles in G / # of connected triplets

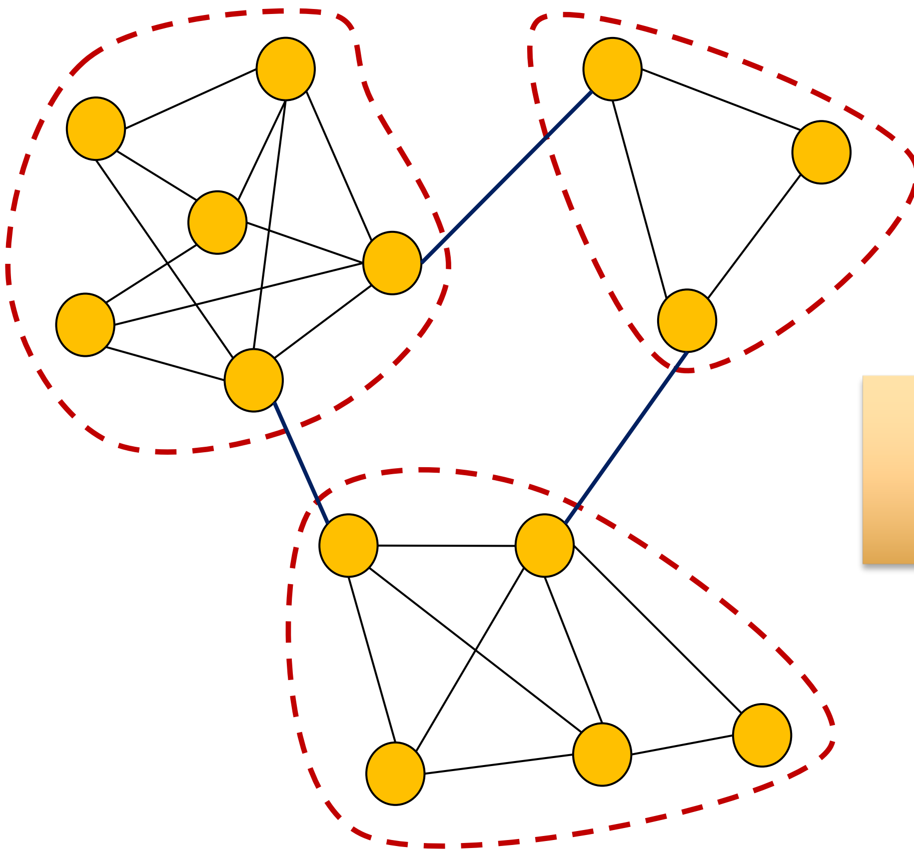
- Captures the transitivity of clustering
  - If **u** is connected to **v** and **v** is connected to **w** …
  - ... it is likely that **u** is also connected to **w**

- Real-world networks tend to have high clustering coefficient
  - Connections to the existence of clustering and community structure property

[Frieze, Gionis, Tsourakakis, 2013]

Community Structure

clustering

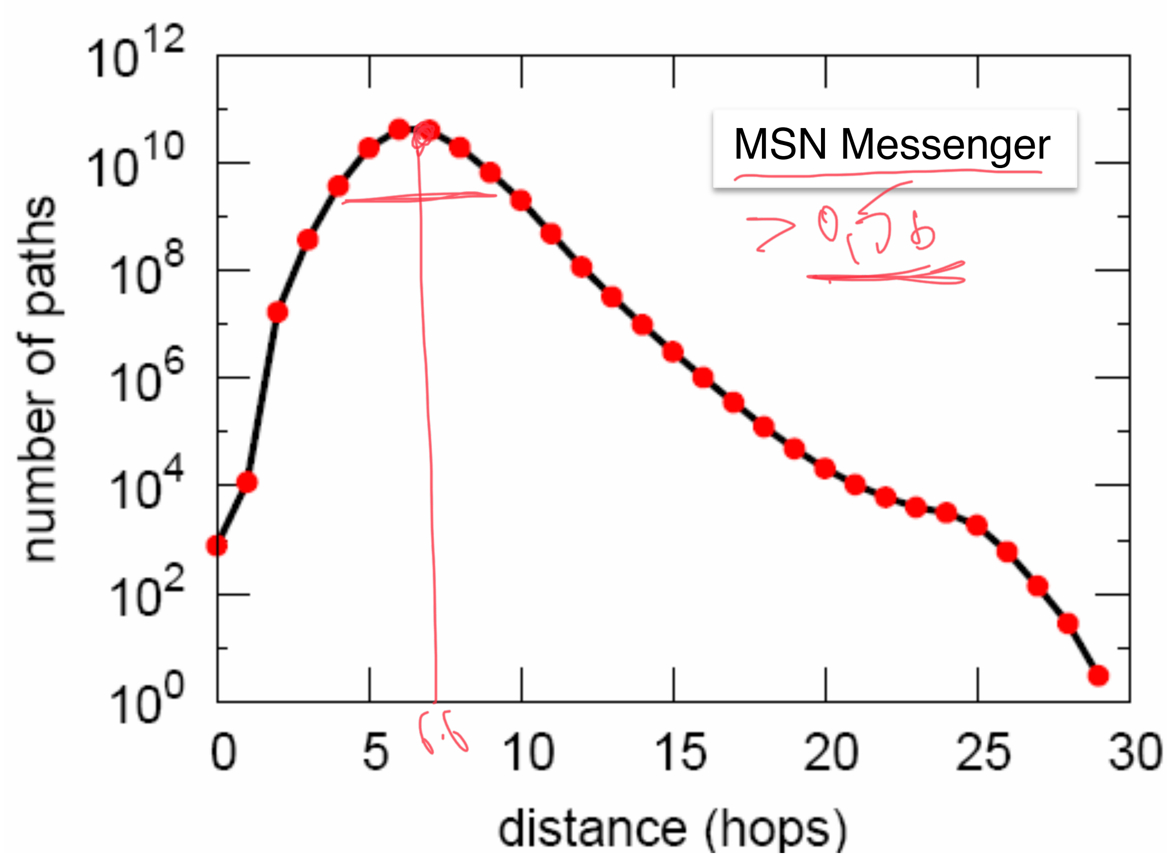


Example graph with three communities

- Will be covered later on in detail

Small-world Phenomenon (1/2)

- The small-world phenomenon appears in various network settings



- **Average path length** is **6.6**
- 90% of the nodes are reachable in less than 8 steps

- **Facebook** network:
  - Average distance is **4.7**
  - [Ugander et al., 2011]

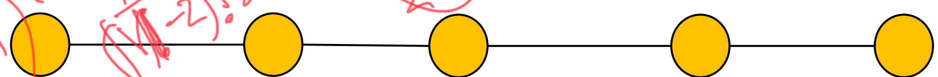
[Leskovec, 2009]

Small Diameter

- **Diameter** is the largest shortest path in the graph
  - Diameter is often sensitive to **chains** of nodes

$G(V,E)$

#sh paths $\binom{|V|}{2} \rightsquigarrow \frac{|V|!}{(|V|-2)!\,2!} \rightsquigarrow \frac{(n-1)n}{2}$



$|V|=n$

- In practice, we use the **effective diameter**
  - Upper bound of the shortest path over 90% of the pairs of nodes

- As an effect of the small-world phenomenon, real networks have **small diameter**

[Leskovec, 2009]

Outline

1. Introduction & Motivation
2. **Graph Generators**
3. Unsupervised learning
    1. Community detection

Graph generating models

Goal: Characterize, model and understand the structure of real networks

- How do real-world networks look like?
  1. **Empirical: statistical properties of networks** (e.g., degree distribution, diameter)
  2. **Generative models of network structure**
     - Mechanisms that reproduce the underlying generative processes

Graph generating models

- Creating models for real-world graphs is important for several reasons
  - Help us to **understand** and **reason** about the observed properties
  - Create **artificial data** for simulation purposes
  - **Predict** the evolution of networks
  - **Privacy preservation:** release the parameters of the generative model, instead of the network itself

What is a Network Model?

- Informally, it is a process (randomized or deterministic) for generating a graph
- Models of **static** graphs
  - **Input:** a set of parameter **Π** and the size of the graph **n**
  - **Output:** a graph **G(Π,n)**
- Models of **evolving** graphs
  - **Input:** a set of parameter **Π** and an initial graph $\mathbf{G_0}$
  - **Output:** a graph $\mathbf{G_t}$ for each time step **t**

Erdős–Rényi Random Graph Model

- Suppose that we want to generate a network with **n** nodes
- The $G_{n,\,p}$ model:
  - Graph with **n** nodes and edge probability **p**
  - For each pair of nodes **(u, v)**, add the edge **(u, v)** **independently** with probability **p**
  - Family of graphs, in which a graph with **m** edges appears with probability

$$p^m (1-p)^{\binom{n}{2}-m}$$

- The $G_{n,\,m}$ model:
  - Select **m** edges uniformly at random

Degree Distribution of the ER Model (1/2)

- **Q:** Do Erdős–Rényi graphs look **realistic**?
- The degree distribution is **Binomial**
  - Let $\mathbf{C_k}$ denote the number of nodes with degree **k**

$$C_k = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

- What if **n → infinity** and we fix the expected degree = **c**?

$$\text{If } n \to \infty \text{ and } np \to c \ (\text{with } c > 0) \text{ then}$$

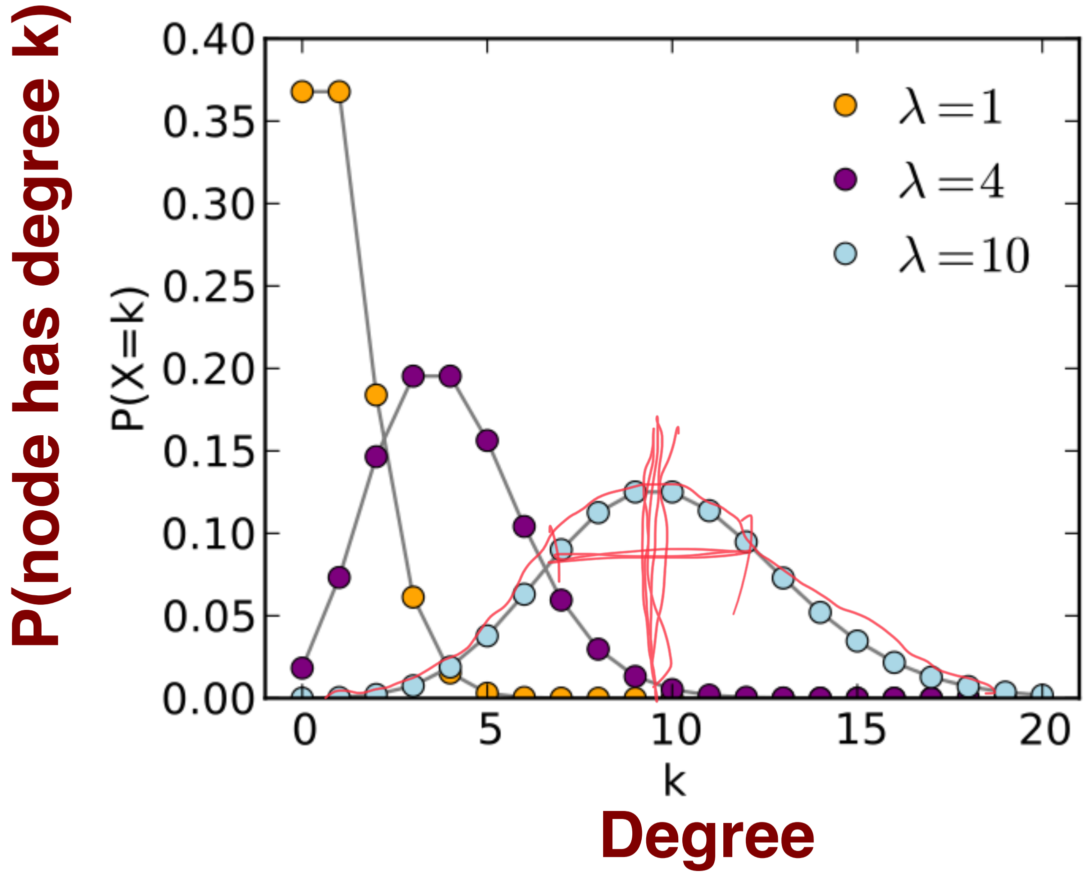
$$\frac{n!}{(n-k)!k!} p^k (1-p)^{n-k} \to \boxed{e^{-c} \frac{c^k}{k!}}$$

Poisson distribution

Degree Distribution of the ER Model (2/2)

Poisson distribution

$$\frac{\lambda^k e^{-\lambda}}{k!}$$



The degree distribution of ER random graph model is **not realistic** for real-world graphs

Source: Wikipedia

Preferential Attachment Model – General Idea

- Real-world networks tend to have **power-law** (or in general heavy-tailed) degree distribution
- 1999 **Barabasi-Albert** (BA) model
  - Based on the idea of preferential attachment
- Intuition
  - Design a graph generator producing a small number of high degree nodes (hubs) and …
  - … also captures the long-tail (nodes with small degree)

Idea: Consider nodes that are more likely to connect to high-degree nodes

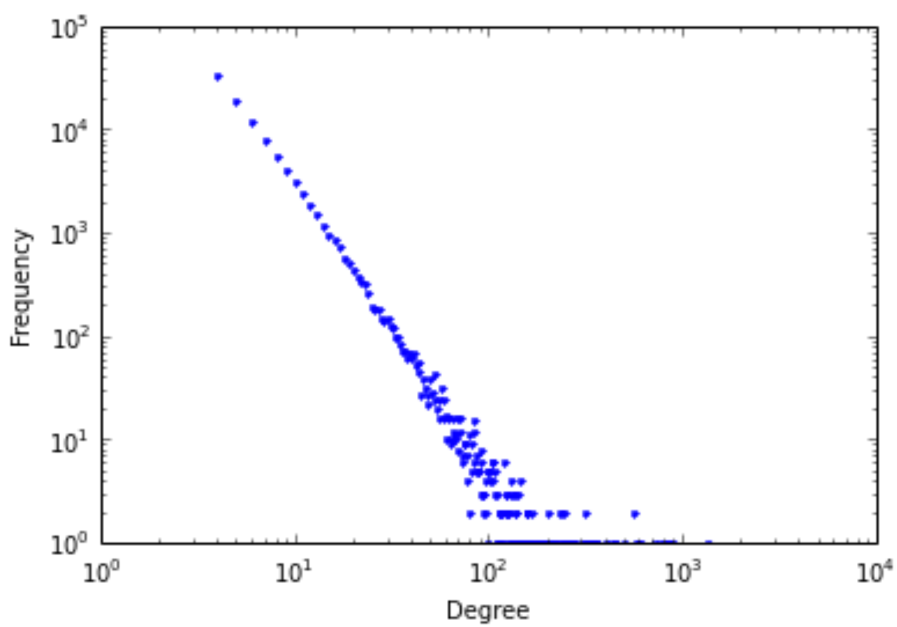
Barabasi-Albert Model (1/2)

- The **Barabasi-Albert** model:
  - **Input:** subgraph G_0 , **m**: degre / new node
  - The process:
    - The nodes are created - one at the time
    - Each new node connects to **m** existing nodes selected with probability proportional to their degree
    - Let **[d1, d2, …, dt]** : degree sequence at time **t**.
      - node at **t+1** will be connected to node **i** with probability

$$p_i = \frac{d_i}{\sum_i d_i}$$

Barabasi-Albert Model (2/2)

- known as the **rich get richer** effect
  - E.g., a web page that already has many incoming hyperlinks is likely to get more in the future
- The BA model produces graphs with **power-law** degree distribution $C_k = k^{-\gamma}$, where $\gamma = 3$



- Barabasi-Albert graph
- **n = 100,000** nodes
- **m = 4**

The BA model holds for several real-world networks (flickr, Delicious, LinkedIn) [Leskovec et al., 2008]

Network Models and Temporal Evolution

- Most of the existing models (e.g., BA) consider that
  - The **number of edges** grows **linearly** with respect to the number of nodes
  - The **diameter increases** based on a factor of **log n** or **log log n**
- In real networks we have observed
  - **Densification power law**
  - **Shrinking diameter**

Kronecker Model of Graphs (1/4)

- Reminder: **Kronecker product** of matrices
  - **A = [a_{ij}]** an **n x m** matrix
  - **B = [b_{ij}]** an **p x q** matrix
  - Then **C = A $\boxtimes$ B** is defined as the **np x mq** matrix

$$\mathbf{C} = \overset{n \times m}{\mathbf{A}} \otimes \overset{p \times q}{\mathbf{B}} \begin{pmatrix} a_{1,1}\mathbf{B} & a_{1,2}\mathbf{B} & \cdots & a_{1,m}\mathbf{B} \\ a_{2,1}\mathbf{B} & a_{2,2}\mathbf{B} & \cdots & a_{2,m}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1}\mathbf{B} & a_{n,2}\mathbf{B} & \cdots & a_{n,m}\mathbf{B} \end{pmatrix}_{np \times mq}$$

- **Intuition:** repeat the Kronecker product between the adjacency matrix of an initial graph to get the final graph

[Leskovec et al., 2010]

Kronecker Model of Graphs (2/4)

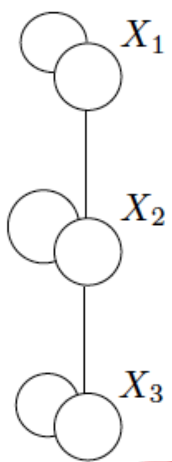
- **Kronecker** model:
  - Start by an initiator adjacency matrix **A_1** of size **p x p**
  - The Kronecker product of two graphs is defined as the Kronecker product of their adjacency matrices
  - The Kronecker graph after **k** iterations is defined as the graph with the following adjacency matrix

$$\mathbf{A}_k = \underbrace{\mathbf{A}_1 \otimes \mathbf{A}_1 \otimes \cdots \otimes \mathbf{A}_1}_{k \text{ iterations}} = \mathbf{A}_{k-1} \otimes \mathbf{A}_1$$

  - Each Kronecker multiplication exponentially increases the size of the graph

[Leskovec et al., 2010]

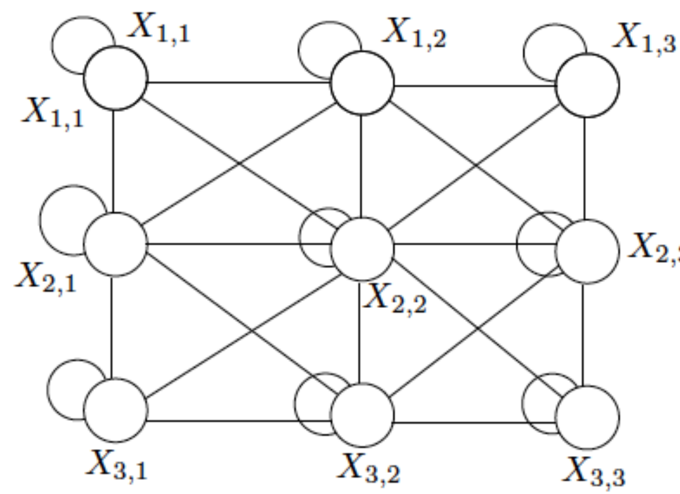
Kronecker Model of Graphs (3/4)



Graph G_1

1	1	0
1	1	1
0	1	1

deform



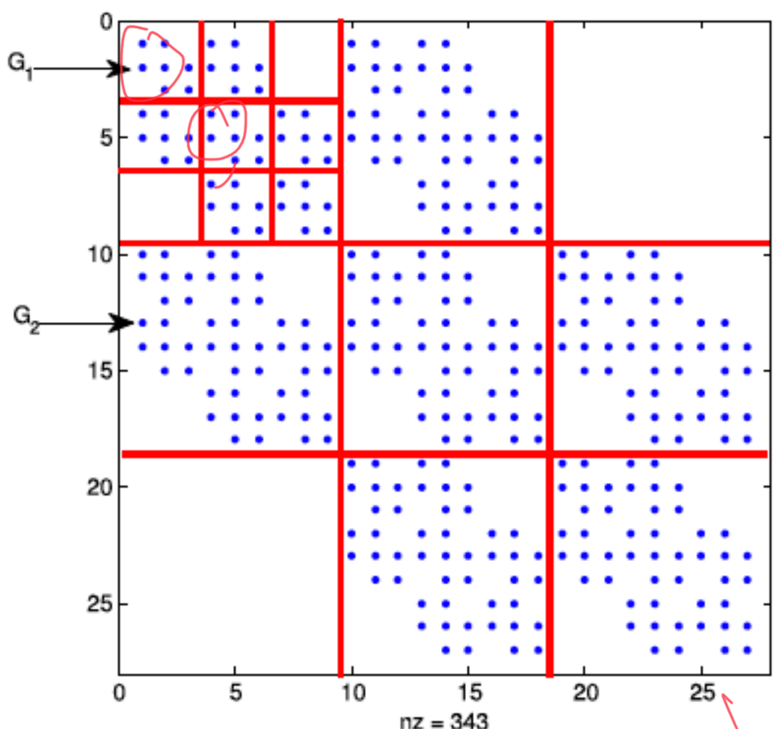
1

Graph $G_2 = G_1 \boxtimes G_1$

G_1	G_1	0
G_1	G_1	G_1
0	G_1	G_1

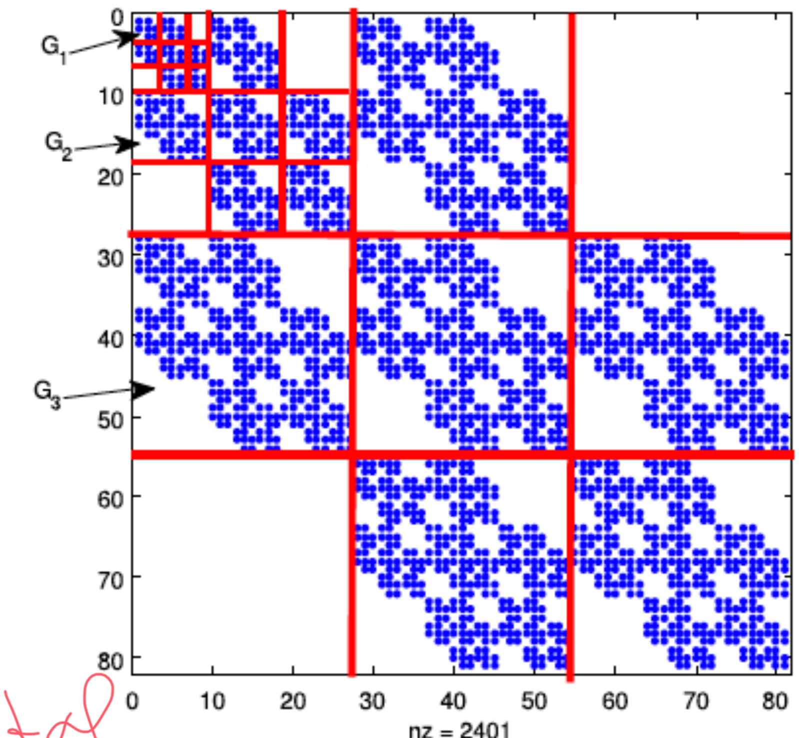
[Leskovec et al., 2010]

Kronecker Model of Graphs (4/4)



(α) $A(G_3) = A(G_2) \otimes A(G_1)$

fractal



(β) $A(G_4) = A(G_3) \otimes A(G_1)$

Intuition: Recursion and self-similarity

[Leskovec et al., 2010]

Stochastic Kronecker Model

- In practice, the **stochastic Kronecker graph** is used
  - Start by an initiator matrix $\boldsymbol{\theta}$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

p(1 -> 2

p(2 -> 1)

  - We obtain a graph with $\mathbf{n = 2^k}$ nodes by repeating $\mathbf{k}$ times the Kronecker product: $\mathbf{A_{k,\theta} = \theta \boxtimes \ldots \boxtimes \theta}$
  - Consider the value **(i, j)** of the matrix $\mathbf{A_{k,\theta}}$ as the probability of existence of the edge **(i, j)** (applying randomized rounding)
  - Typically, **2 x 2** initiator matrices produce good results

[Leskovec et al., 2010]

Generate Realistic Kronecker Graphs

- Given a network **G**, how can we find a "good" initiator matrix **θ**, such that $\mathbf{A_G \sim= \theta \boxtimes \ldots \boxtimes \theta}$?
  - Fit the parameters **θ** of the model
  - Idea: use **maximum-likelihood estimation**

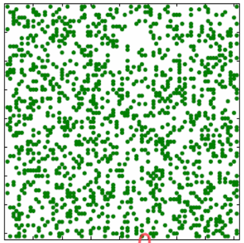
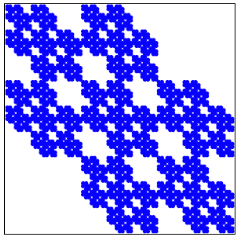
$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$ + K

EM

$$\arg\max_{\Theta} P(G|\Theta)$$

Θ(, K)

After Kronecker products

arg max$_{G_1}$ ( | )

real

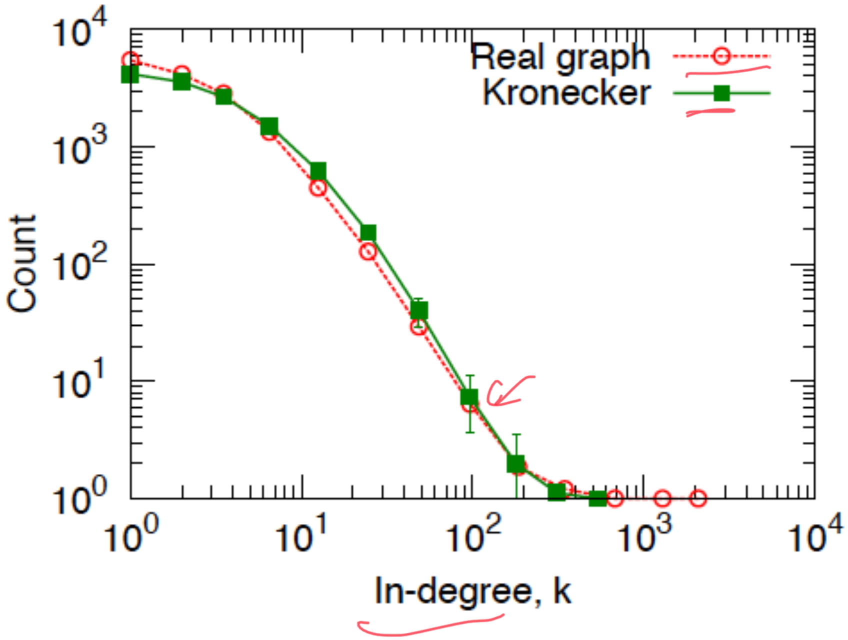
[Leskovec et al., 2010]

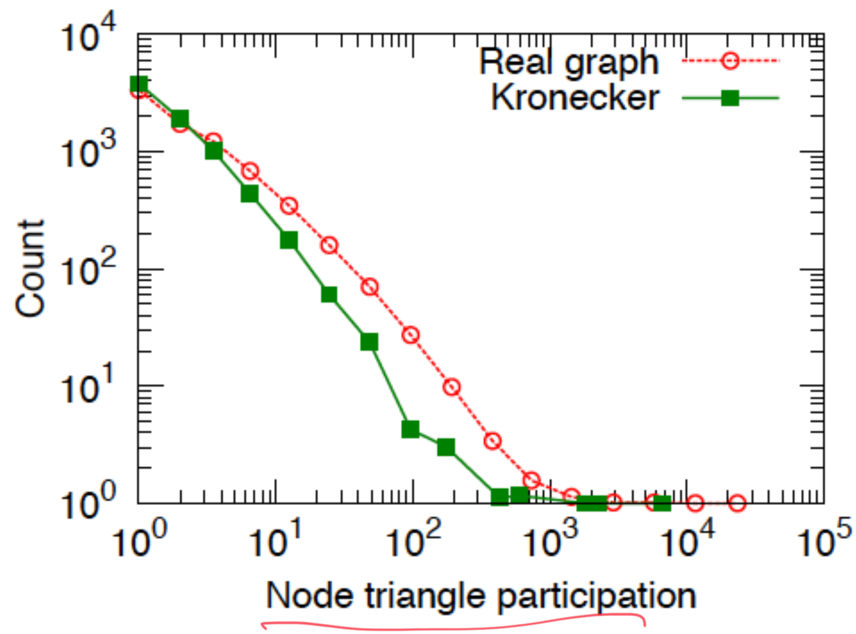
Properties of Kronecker Model

- The Kronecker (stochastic) graph model is able to reproduce a plethora of properties
  - Power-law degree distribution
  - Small diameter
  - Shrinking diameter
  - Densification power-law
  - Triangle participation
  - ...

[Leskovec et al., 2010]

Example: Fitting Kronecker Model to a Graph





Blog-to-Blog network

[Leskovec et al., 2010]

References

- J. Leskovec. Modeling Large Social and Information Networks. Tutorial at ICML, 2009.
- J. McAuley. Data Mining and Predictive Analytics, UCSD, 2015.
- D. Easley and J. Kleinberg. Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press, 2010.
- J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, Z. Ghahramani. Kronecker Graphs: An approach to modeling networks. JMLR, 2010.
- P. Holland, K. B. Laskey , and S. Leinhardt . Stochastic blockmodels : Some first steps. Social Networks 5, 1983

generators

- SBM
- Deep learning autoencoders

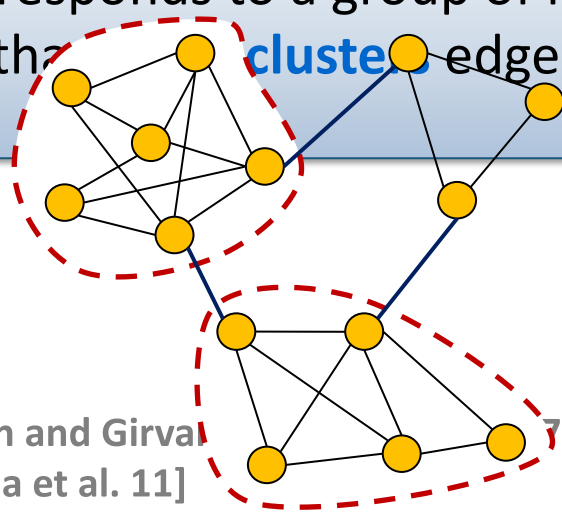
Outline

1. Introduction & Motivation
2. Graph Generators
3. **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)

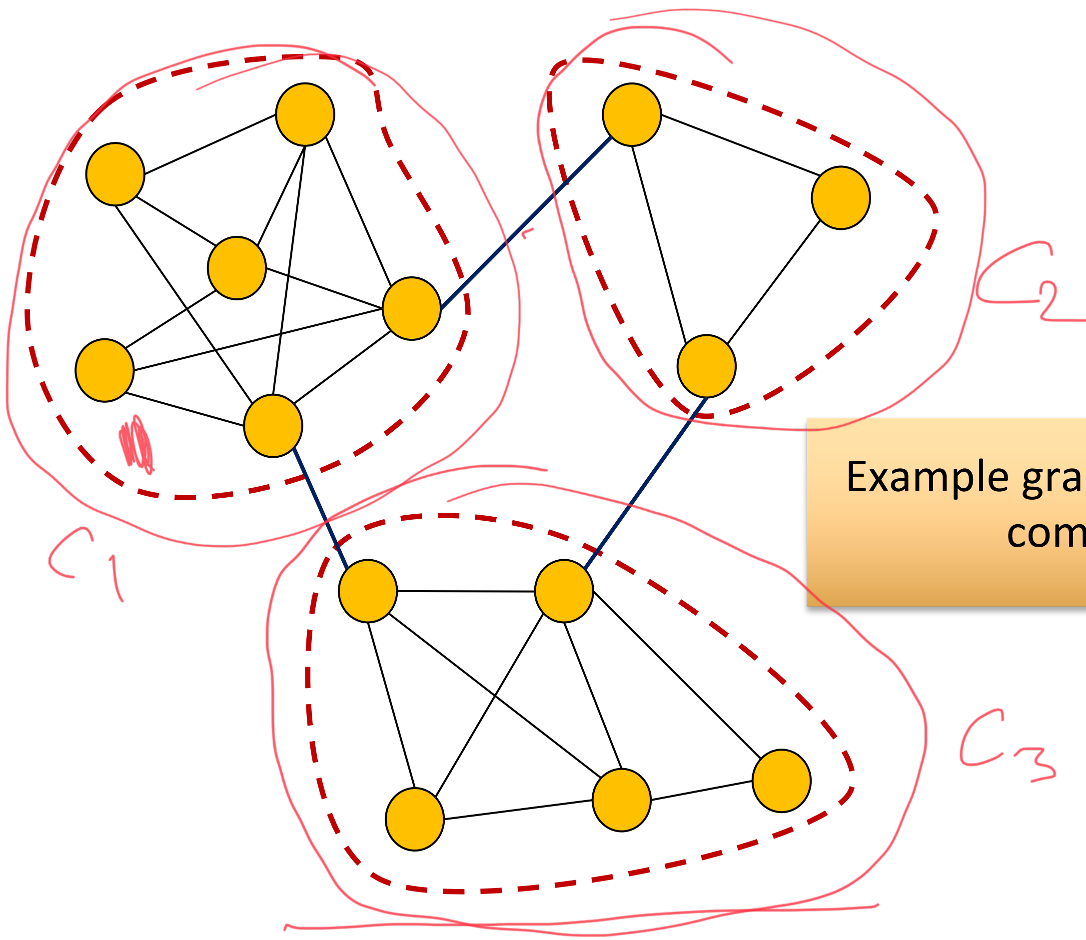
A community corresponds to a group of nodes with more **intra-cluster** edges tha **cluste** edges



Example graph with three communities

[Newman '03], [Newman and Girva 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

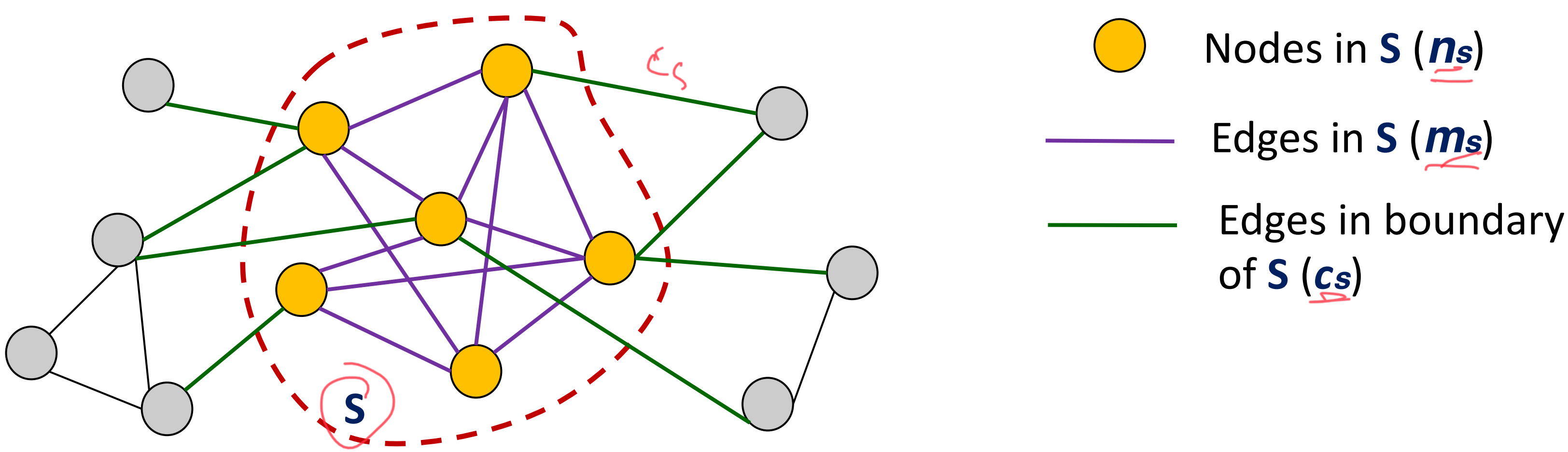
- Focus on
  - Intra-cluster edge density (# of edges within community),
  - Inter-cluster edge density (# of edges across communities)
  - Both two criteria

- 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**, $m_s = |\{(u,v): u \in S, v \in S\}|$
- c_s is the number of edges on the boundary of **S**, $c_s = |\{(u,v): u \in S, v \notin S\}|$
- d_u is the degree of node u
- f **(S)** represent the clustering quality of set S



Evaluation based on external connectivity

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

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

good f(S) ↓

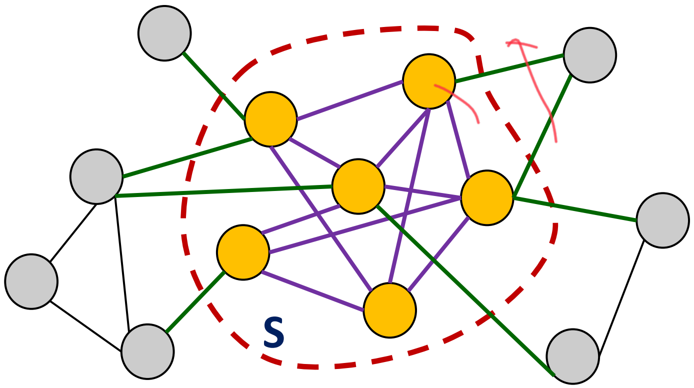
> Measures the number of edges per node that point outside **S**

- **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**



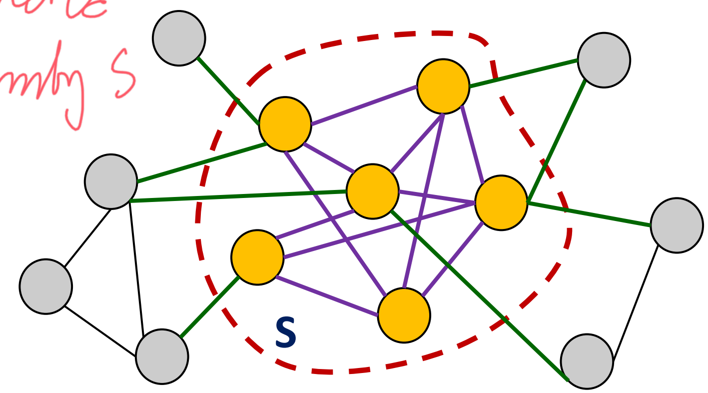
Evaluation based on internal connectivity (1)

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

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

fully connected
community S

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



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

$$f(S)=m_s$$

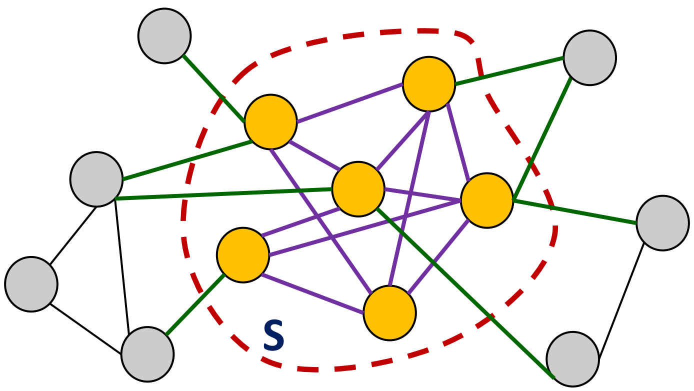
Number of edges between the nodes of **S**

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]

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

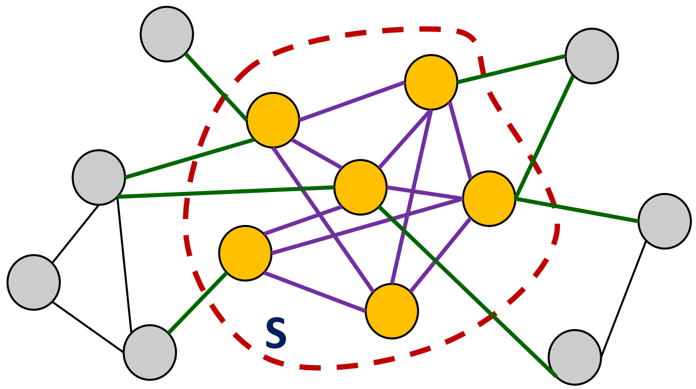
Measures the fraction of total edge volume that points outside **S** normalized by the size of **S**

Evaluation based on internal connectivity (3)

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

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

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



Evaluation based on network model

- **Modularity** [Newman and Girvan ‘04], [Newman ‘06]

$$\boxed{f(S)=\frac{1}{4}\big(\underbrace{m_s-E(m_s)}_{\geq 0}\big)}$$

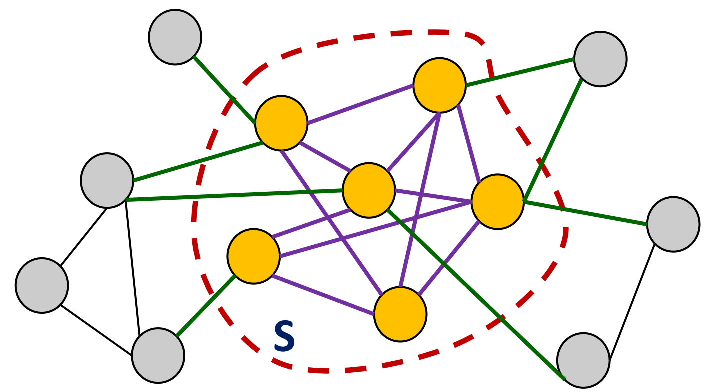
≤ 0

< 0

$\frac{|G|}{|V|} m_s$

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



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 $W(i,j)=w_{ij}$

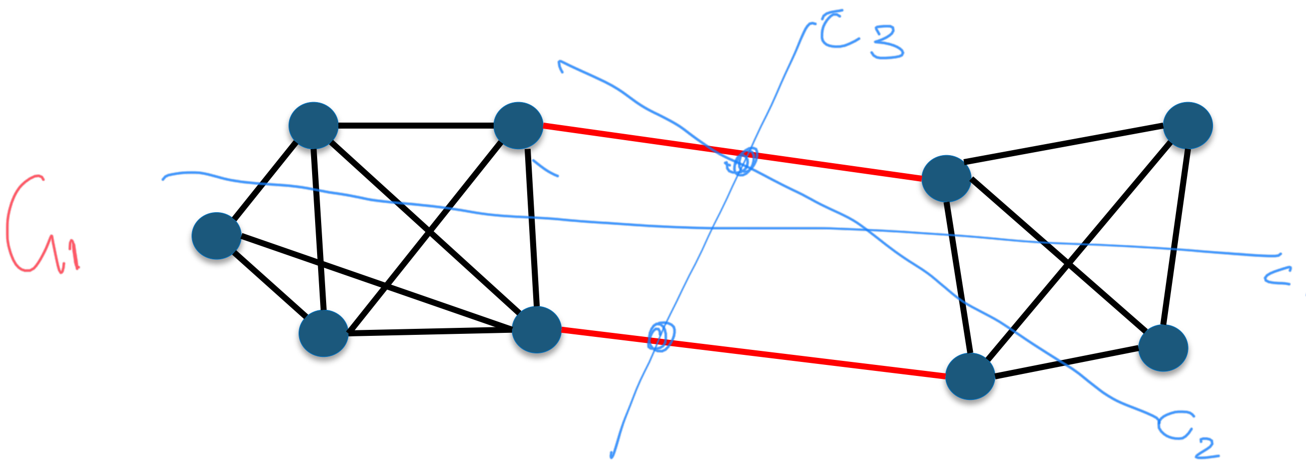
Graph-Cut

- For k clusters:

  cluster

  rest of graph

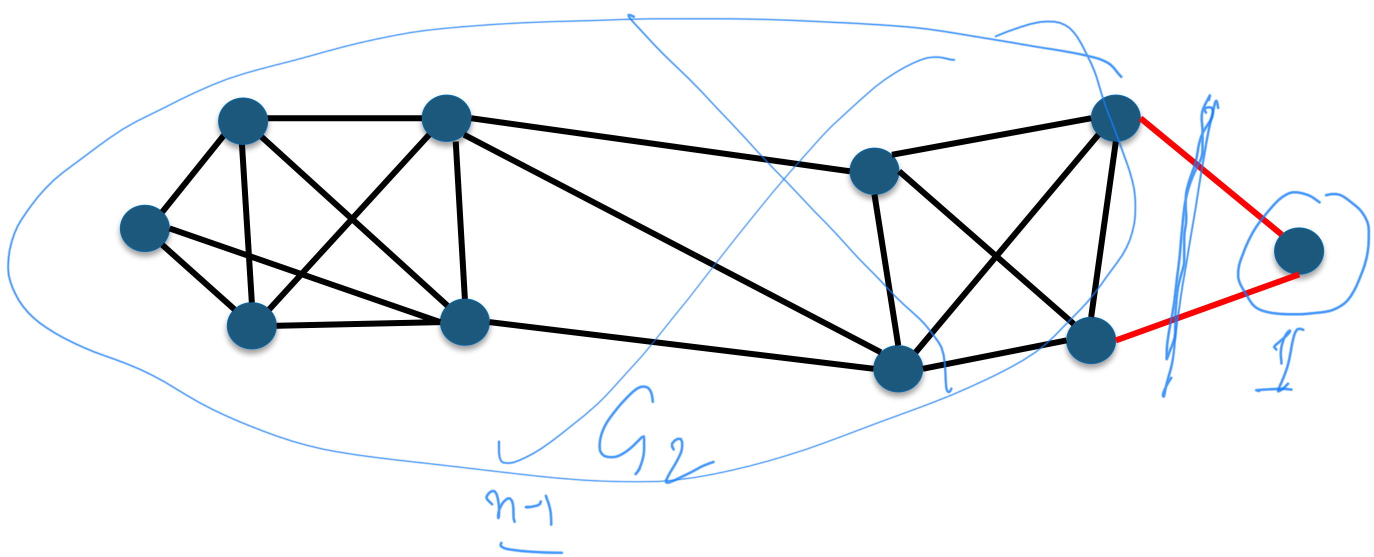
  - $cut(A_1,..,Ak) = 1/2\sum_{i=1}^{k} w(Ai,\overline{A_i})$
    - undirected graph:1/2 we count twice each edge



- 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):

argmin $A_1 \dots A_k$

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

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

$$Ncut(A_1, \dots Ak) = \sum_{i=1}^{k} \frac{cut(Ai, \overline{A_i})}{v(A_i)}$$ $= \sum_i d_i(v_i) \mid v_i$ in 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 :

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

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

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

weight edge

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

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

Graph Laplacian

- How is the previous useful in Spectral clustering?

assign node/cluster

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

1 max → $f_i = f_j$ → $f_i - f_j = 0$

0 not conn → $f_i \neq f_j$ → " max

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

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

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

min

$$= 2(\boldsymbol{f}^T\boldsymbol{D}\boldsymbol{f} - \boldsymbol{f}\boldsymbol{T}\boldsymbol{W}\boldsymbol{f}) = 2\boldsymbol{f}^T\underbrace{(\boldsymbol{D} - \boldsymbol{W})}_{L:\ \text{Laplacian}}\boldsymbol{f} = \boxed{2\boldsymbol{f}^T\boldsymbol{L}\boldsymbol{f}}$$

$\min(f^T L f)$

$f = \operatorname{eig}(L)$

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

Properties of L

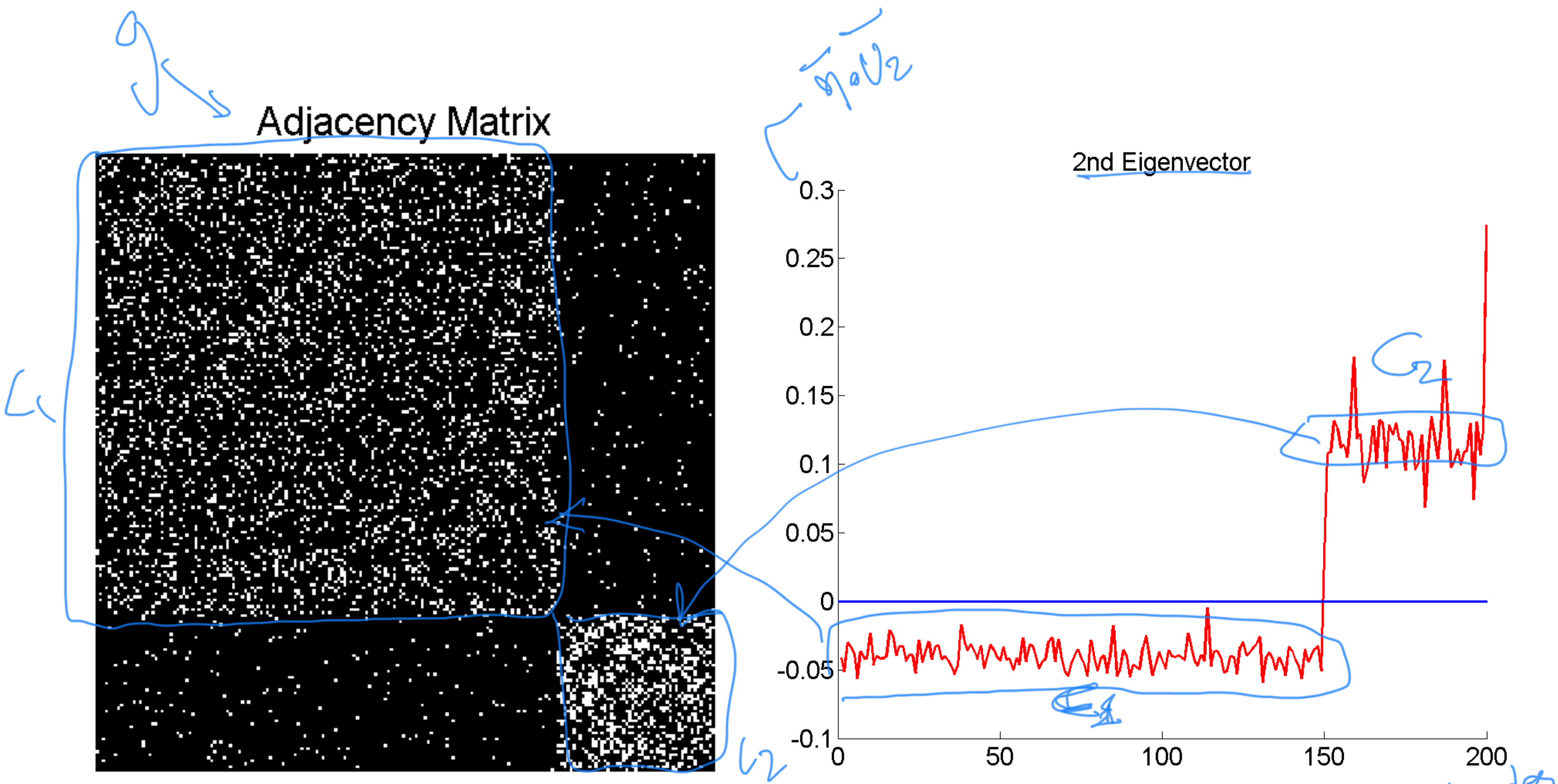
- L is
  - Symmetric
  - Positive
  - Semi-definite
- The smallest eigenvalue of L is 0
  - The corresponding eigenvector is $\mathbb{1}$
- L has n non-negative, real valued eigenvalues
  - $\underbrace{0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n}$

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 , $fi<0$ to the other

Example



$\mathcal{L} = D - W \rightarrow \mathcal{L} = U \Lambda U^T$

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 :

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

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

Ratio Cut

- We have $min_f f^T L f$ subject to
  $$f^T 1 = 0,\ fTf = n$$

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

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

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

Normalized Cut

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

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

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

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

Normalized Cut

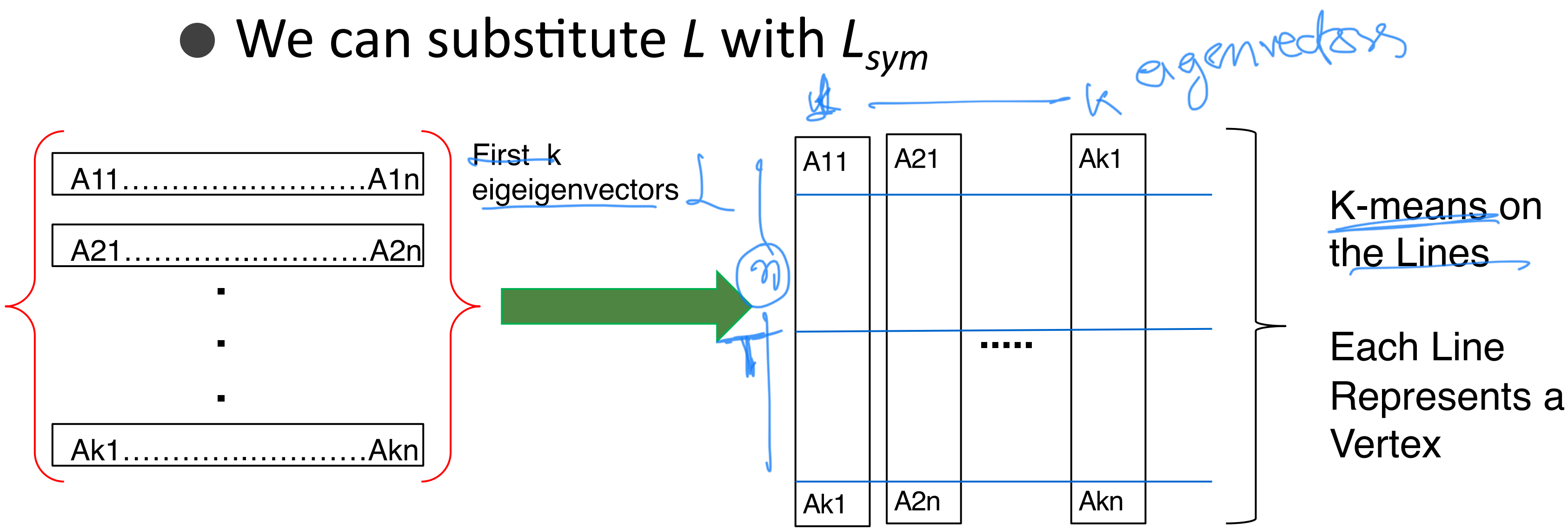
- Similarly we come to : $min_f f^T L f$
  subject to $f^T D 1 = 0,\ fTDf = v(V)$
- Assume $h = D^{1/2} f$
  - $min_h h^T D^{-1/2} \mathrm{L} D^{-1/2} \mathrm{h}$ subject to
    $$h^T D^{1/2} 1 = 0, \qquad h^T h = v(V)$$
  - The answer is in the eigenvector of the second smallest eigenvalue of $L_{sym}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

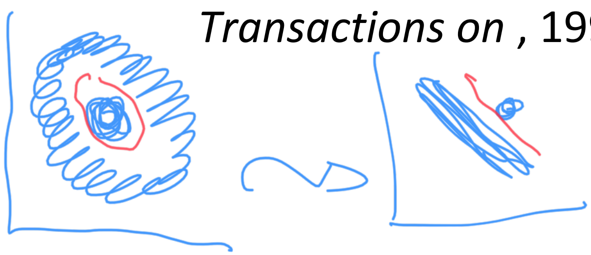
Multi-Way Graph Partition

- 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}



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

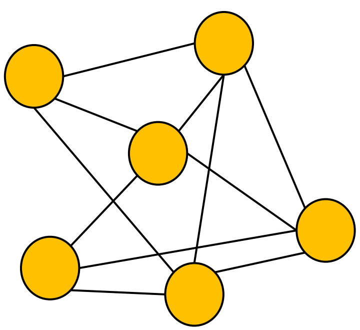


Graph Clustering Algorithms

- **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 $m = |E| = ½ \Sigma_i k_i$



Formal definition of modularity

- **Modularity Q**

1/0

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

where

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

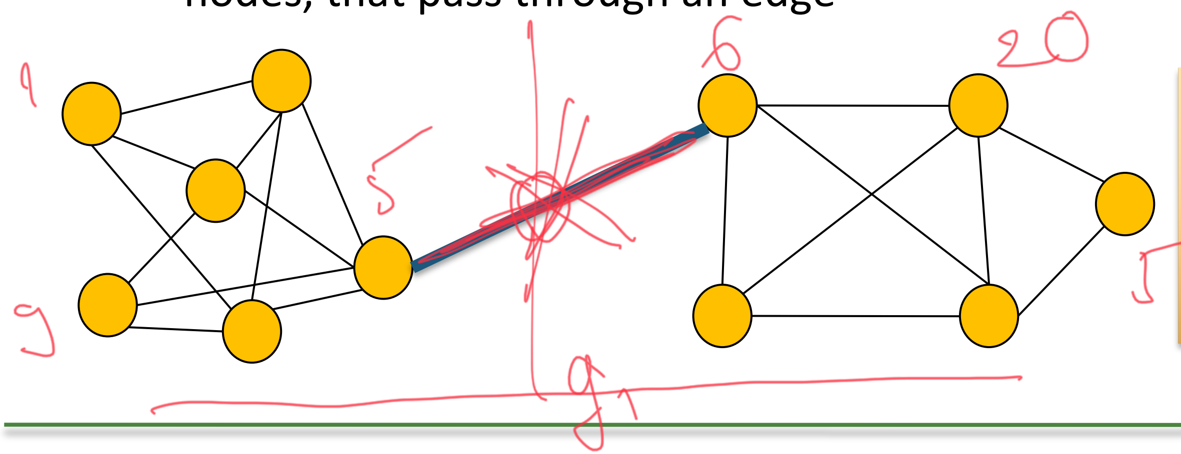
Applications of modularity

- 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



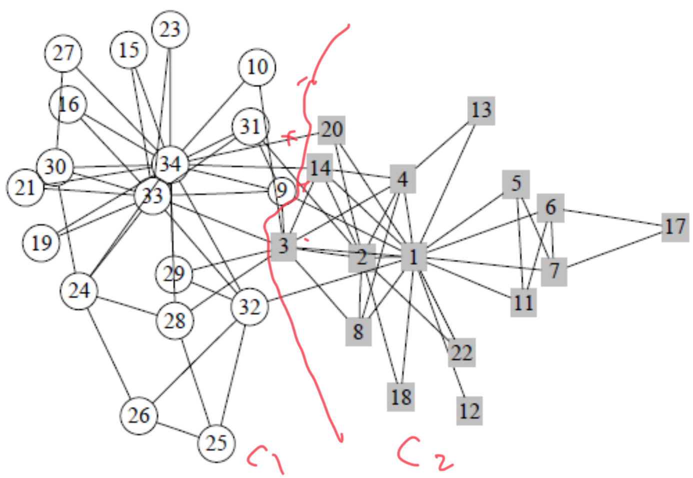
Edge betweenness is higher for edges that connect different communities

Newman-Girvan algorithm (1)

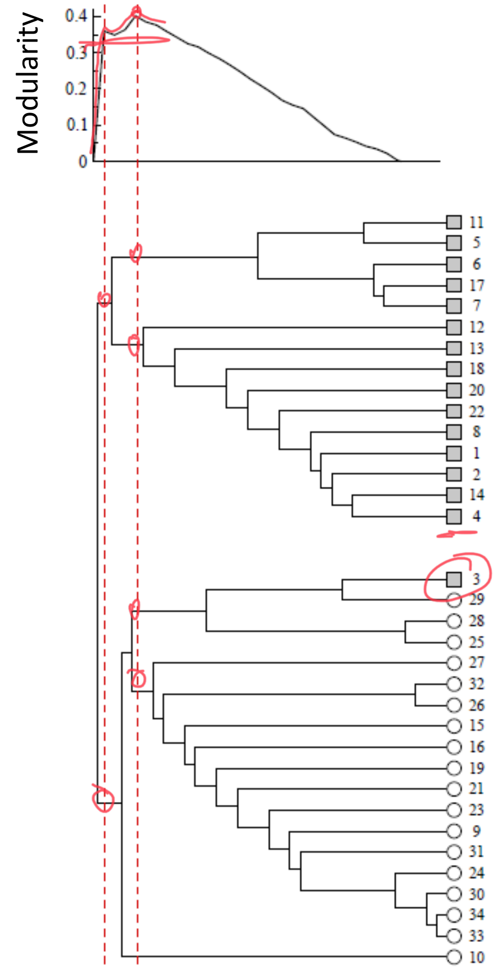
- **Basic steps:**
  1. Compute betweenness centrality for all edges in the graph
  2. Find and remove the edge with the highest score
  3. Recalculate betweenness centrality score for the remaining edges
  4. Go to step 2
- How do we know if the produced communities are **good ones** and stop the algorithm?
  - 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: $\mathbf{O(m^2 n)}$ (or $\mathbf{O(n^3)}$ 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**

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

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

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

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

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

$\{B\} = U \Lambda U^T$

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

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

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

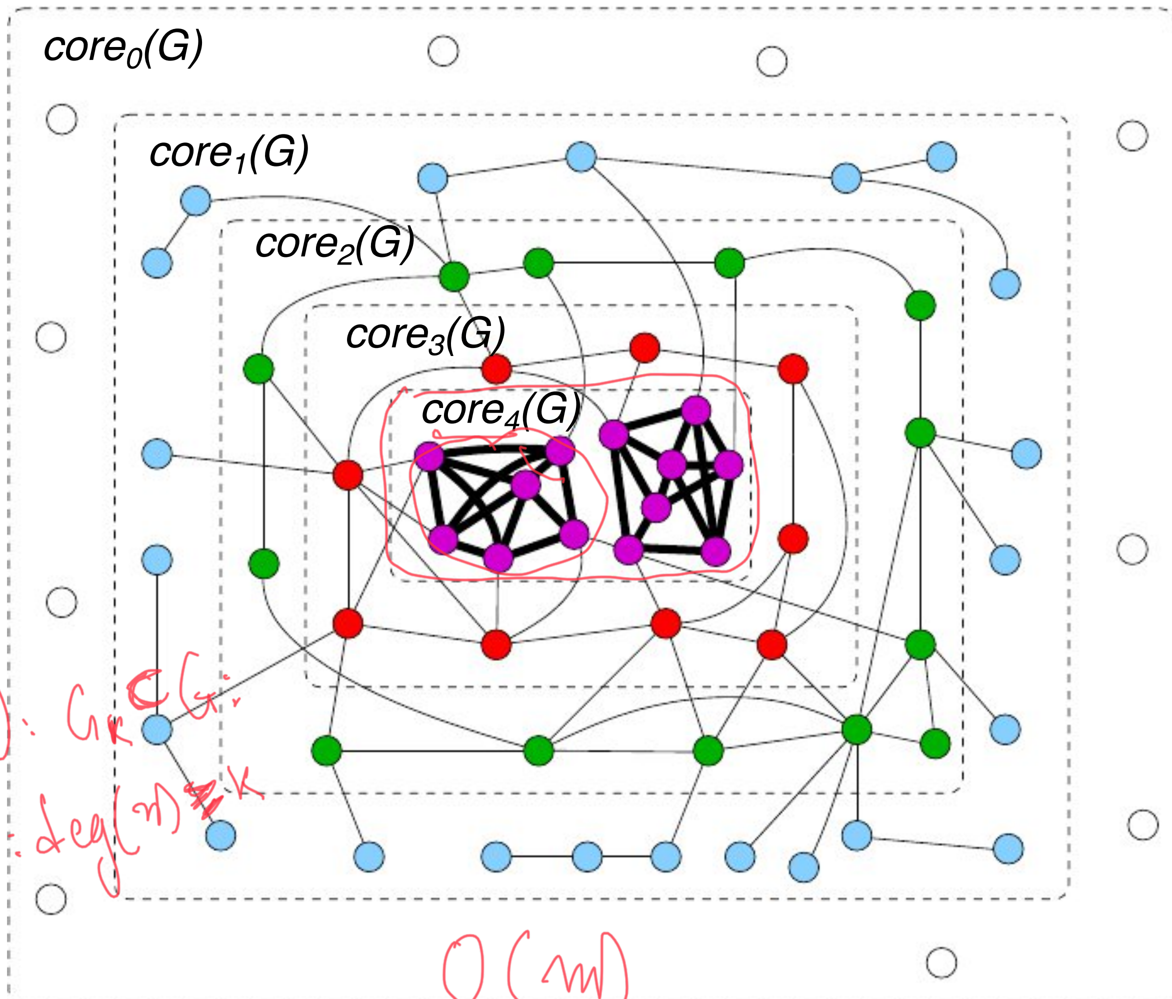
Extensions of modularity

- 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



(densest subgraph)

k-core

$(1-\frac{1}{e})$

k-core(G): $G_k \subseteq G$: $\forall n \in G_k : deg(n) \geq k$

k in $\mathbb{N}$

$O(m)$

edges → sparse graphs!

Fractional k-cores

Co-authorship edge weight:

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

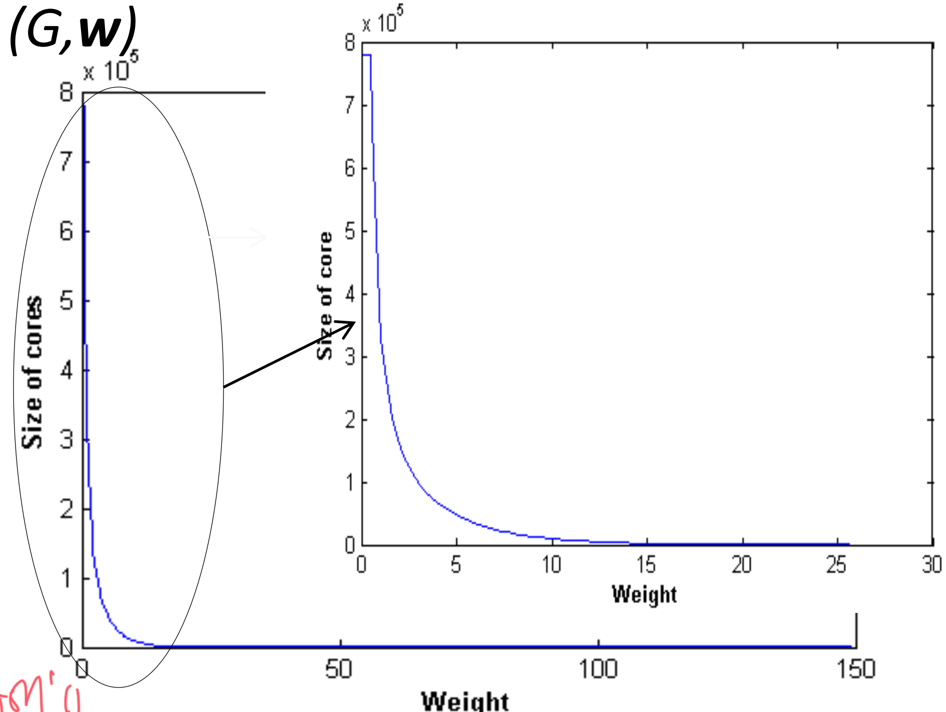
- For every edge *e = {x, x'}* we set
- *The weighted co-authorship affinity among x and x': collaboration !*

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

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

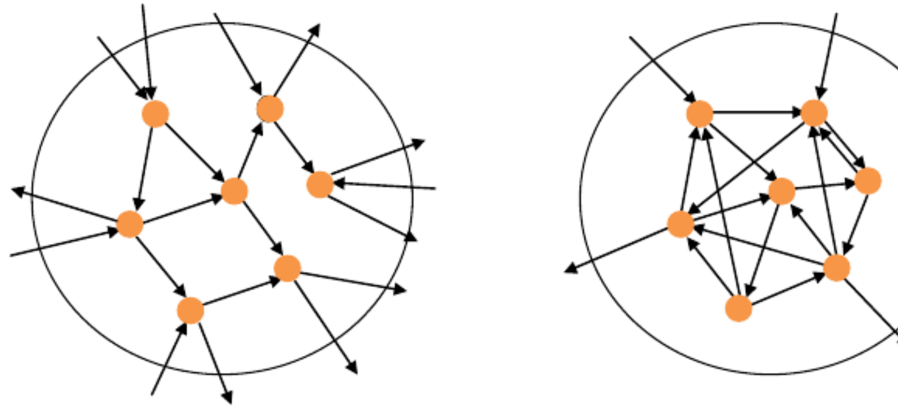
- the total co-authorship value of an author

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



2012 → giatsidis ... Asonam'11

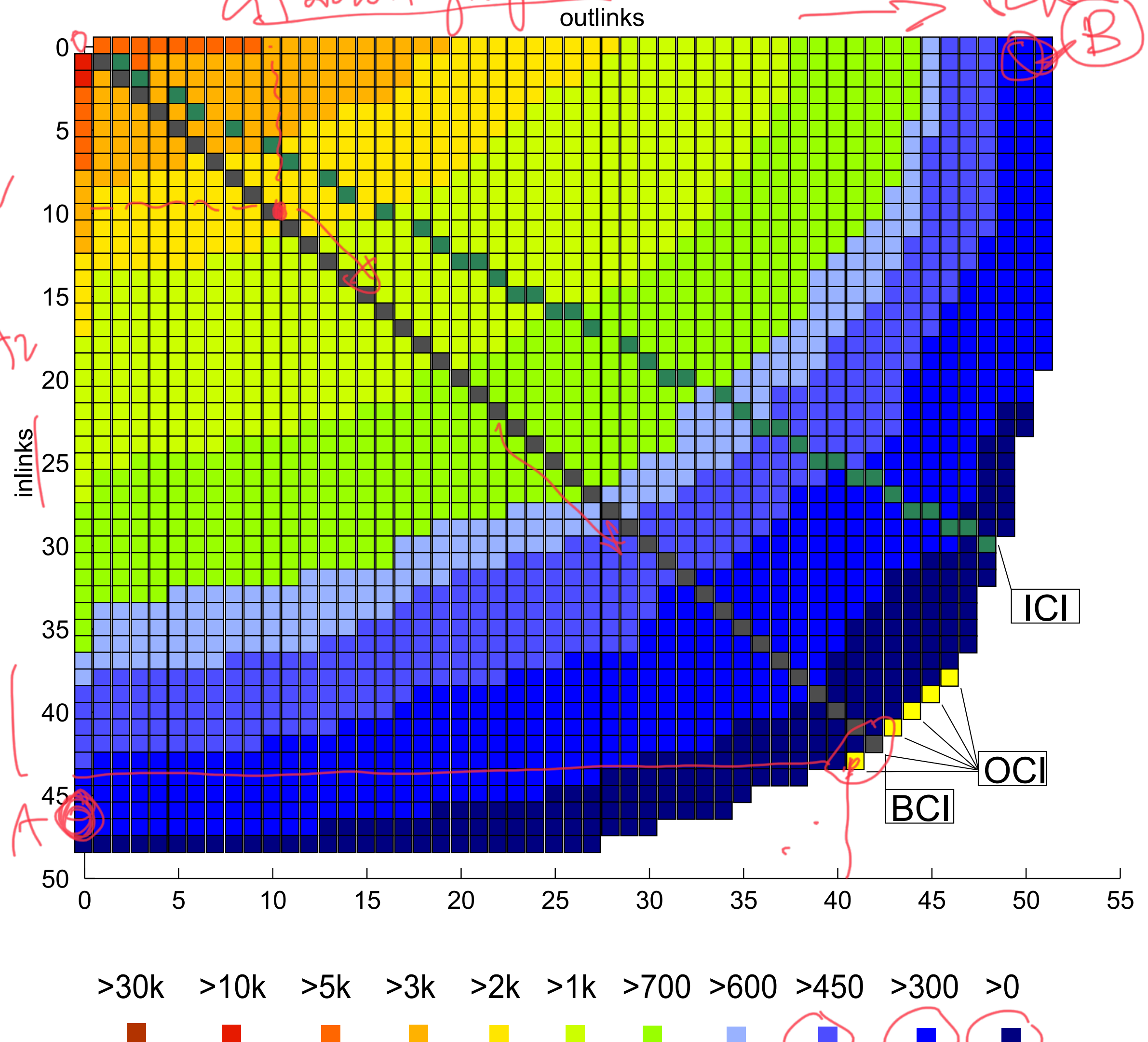
Degeneracy on directed graphs



- 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: (k,l)-core → in$(n) \geq k$
out$(n) \geq l$

D-core matrix for DBLP



The Extreme DBLP D-core authors

2012

≥ 43 in
≥ 44 out

- 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 Turing 2017
- Serge Abiteboul
- Richard Hull
- Victor Vianu
- Jeffrey D. Ullman
- Michael Kifer
- Philip A. Bernstein
- Vassos Hadzilacos
- Elisa Bertino
- Stefano Ceri
- Georges Gardarin

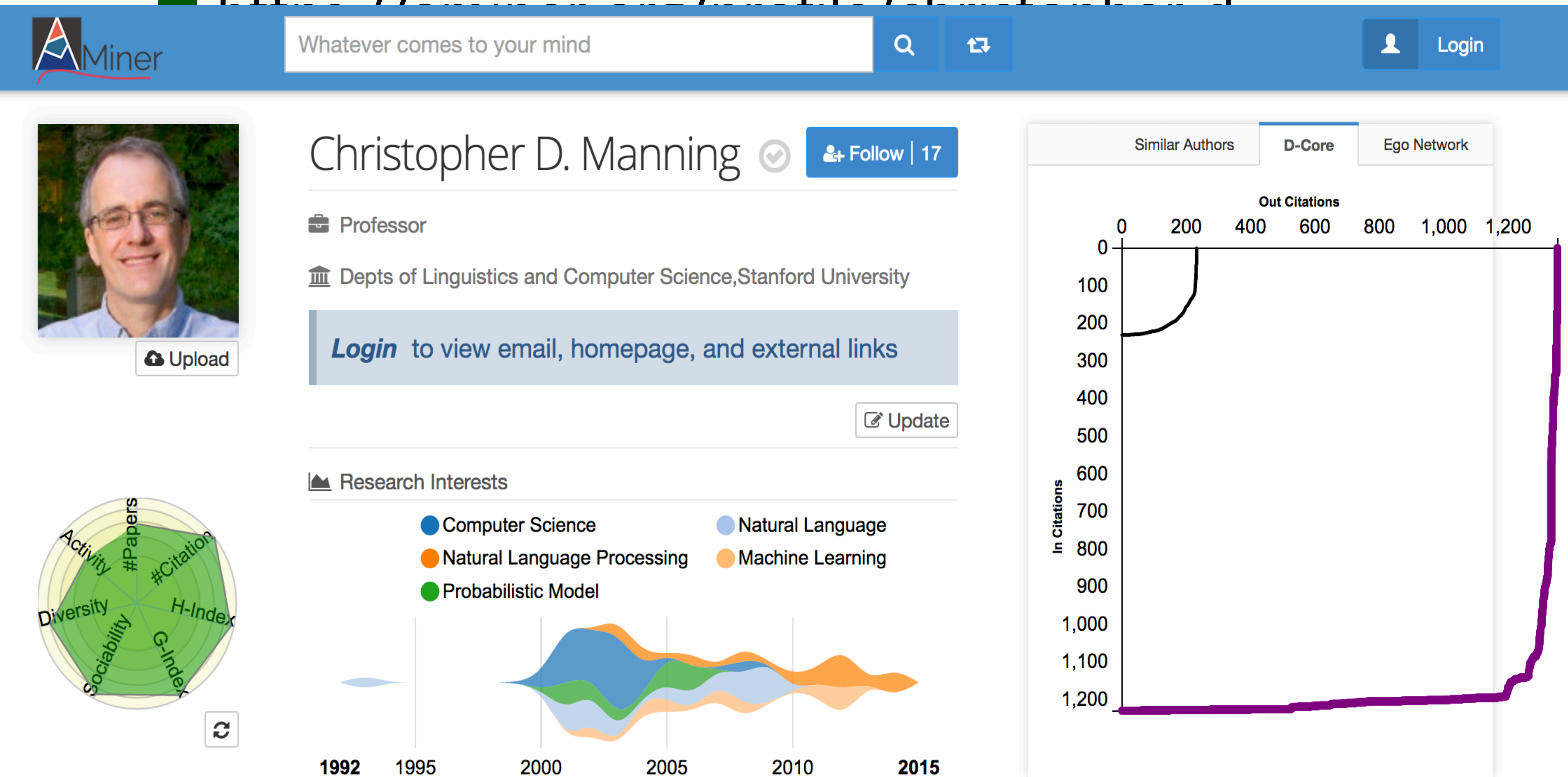
- Patrick Valduriez
- Ramez Elmasri
- Richard R. Muntz
- David B. Lomet
- Betty Salzberg
- Shamkant B. Navathe
- Arie Segev
- Gio Wiederhold
- Witold Litwin
- Theo HÃ¤rder
- FranÃ § ois Bancilhon
- Raghu Ramakrishnan
- Michael J. Franklin
- Yannis E. Ioannidis
- Henry F. Korth
- S. Sudarshan
- Patrick E. O'Neil
- Dennis Shasha
- Shamim A. Naqvi $O(m)$
- 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 adopted by aminer.org



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

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/

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.

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.