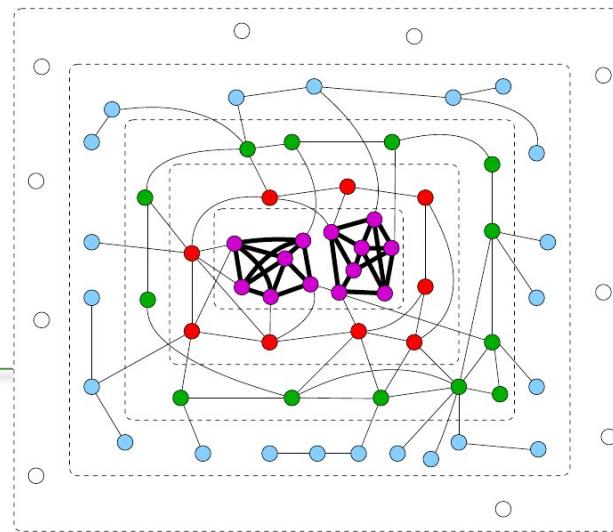


Graph Mining I

Community Detection and Graph Clustering



Michalis Vazirgiannis
LIX @ Ecole Polytechnique

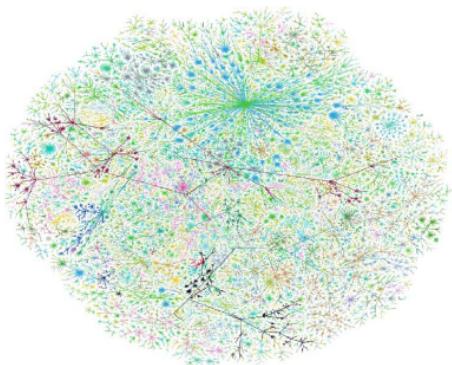
based on the “[Community Detection and Evaluation
in Social and Information Networks](#)” tutorial

15th International Conference on Web Information System Engineering (WISE)
Thessaloniki, October 12-14, 2014

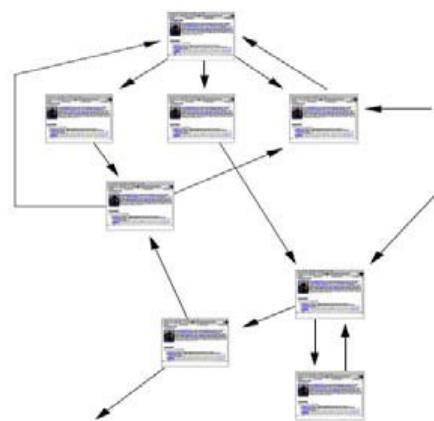
Outline

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

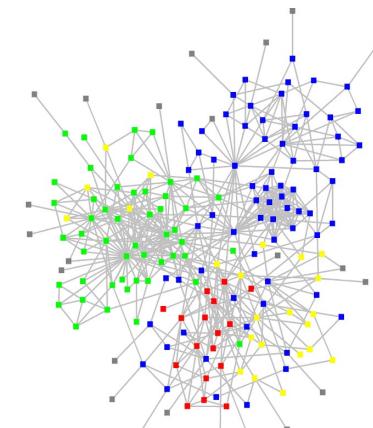
Networks are Everywhere



(a) Internet



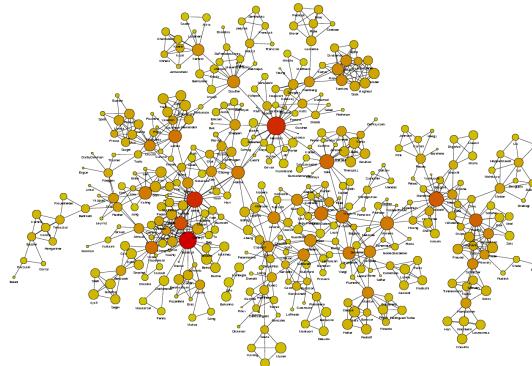
(b) World Wide Web



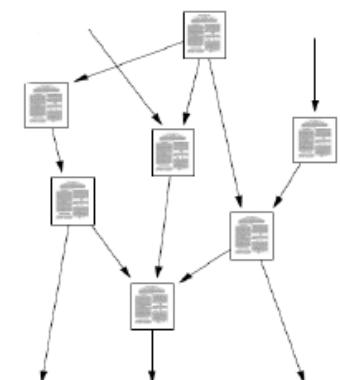
(c) Email network



(d) Social network



(e) Collaboration network



(f) Citation network

Social Networks Growth

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

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

Graphs are ubiquitous!

■ Technological networks:

- Internet
- Telephone networks
- Power grid
- Road, airline and rail networks

■ Information networks:

- World Wide Web
- Blog networks
- Citation networks

■ Social networks:

- Collaboration networks
- Organizational networks
- Communication networks

■ Biological networks:

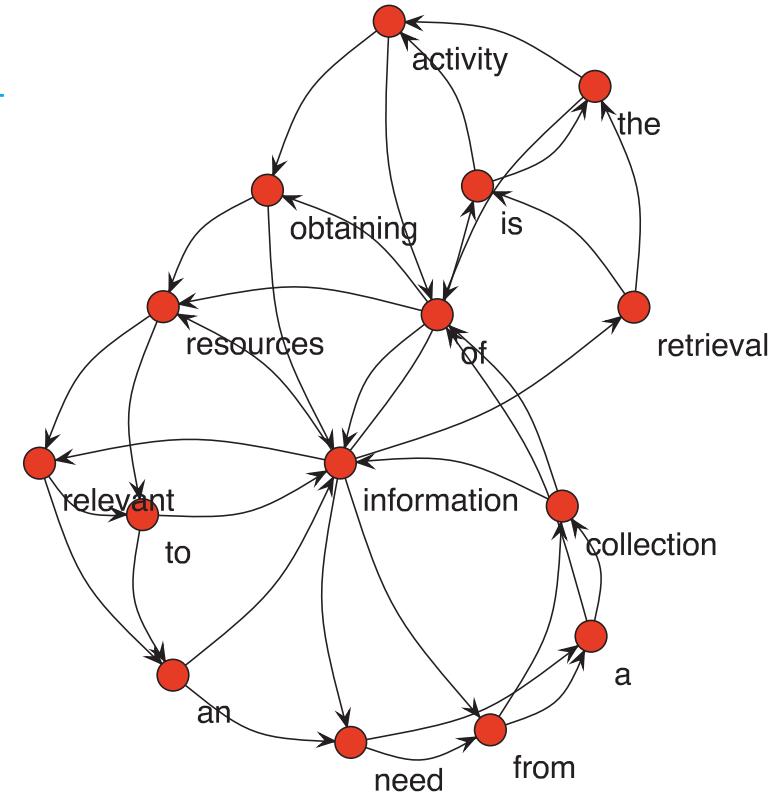
- Networks from Neuroscience
- Protein-protein interaction networks
- Gene regulatory networks
- Food webs

■ Software networks:

- Call graphs
- Software module/component interaction networks

Even representing text - Graph-of-word

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



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

How Big are the graphs?

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

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

Space/Time Complexity

Undirected graph space complexity

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

Social scale...

1 billion vertices, 100 billion edges

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

111 PB adjacency matrix

2.92 TB adjacency list

Edge List: $O(4m)$

2.92 TB edge list

Time complexity

Pagerank: $O(n^2)$

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

...

Why Graph Mining

Understand the structure and dynamics of complex interaction systems

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

Several application domains

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

Elements og Learning from Graph data

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

Communities in Real Networks

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

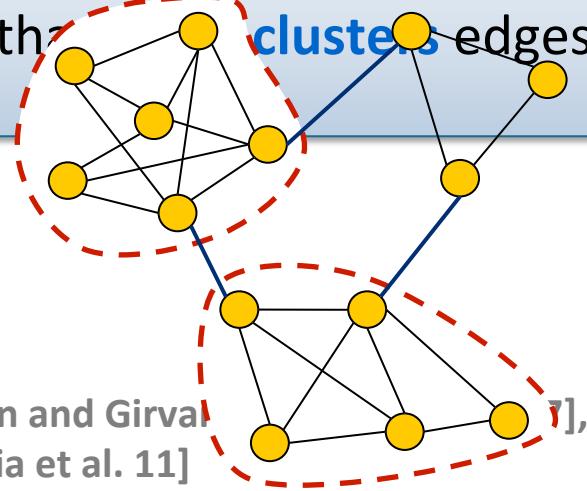
Outline

1. Introduction & Motivation
2. Community evaluation measures
3. Graph clustering algorithms

Basics

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

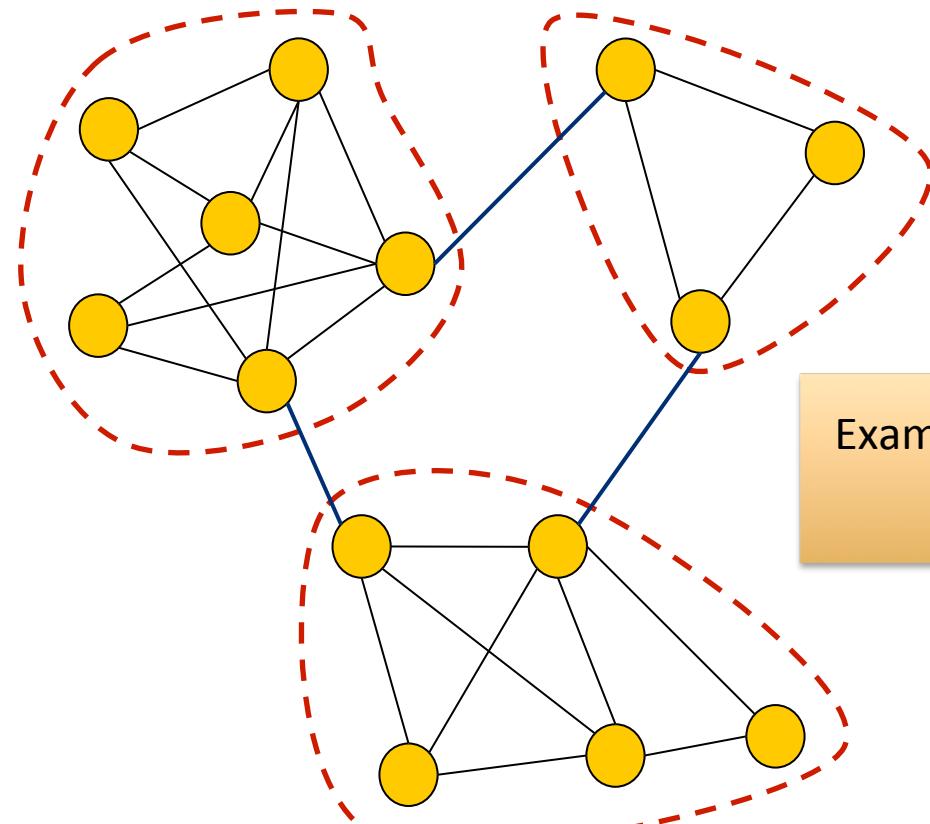
A community corresponds to a group of nodes with more **intra-cluster** edges than **cluster** edges



Example graph
with three
communities

[Newman '03], [Newman and Girvan '04], [Girvan and Newman '02], [Lancichinetti et al. '08], [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

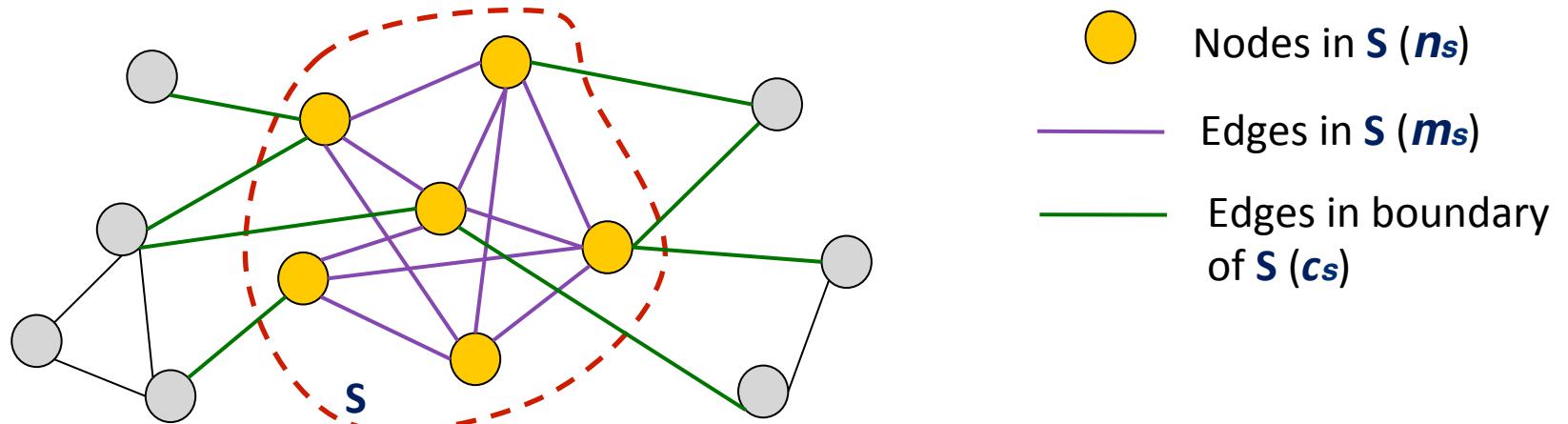
■ We group the community evaluation measures according to

- 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

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

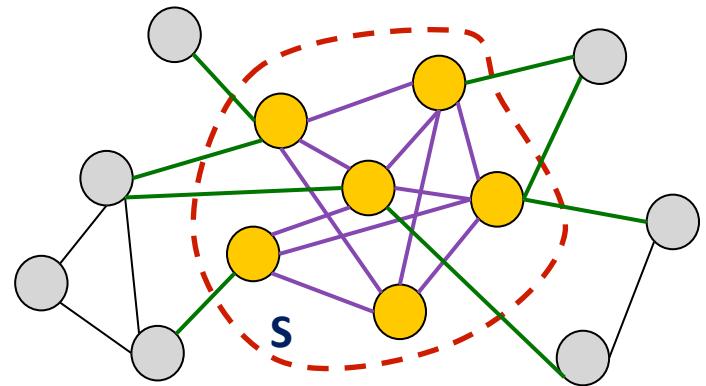


Evaluation based on internal connectivity (1)

■ Internal density [Radicchi et al. '04]

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

Captures the internal edge density of community S



■ Edges inside [Radicchi et al. '04]

$$f(S) = m_s$$

Number of edges between the nodes of S

Evaluation based on external connectivity

■ Expansion [Radicchi et al. '04]

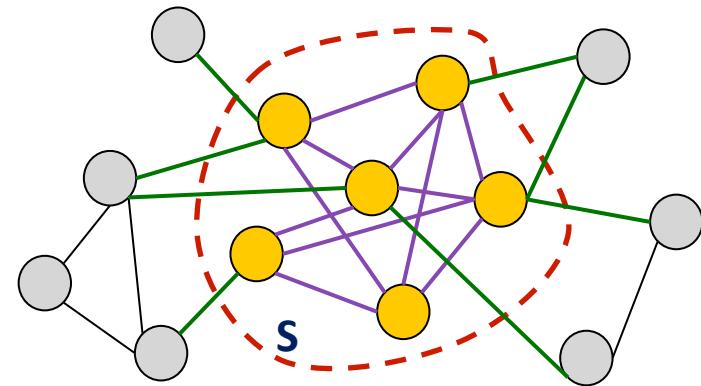
$$f(S) = \frac{c_s}{n_s}$$

Measures the number of edges per node that point outside S

■ Cut ratio [Fortunato '10]

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

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

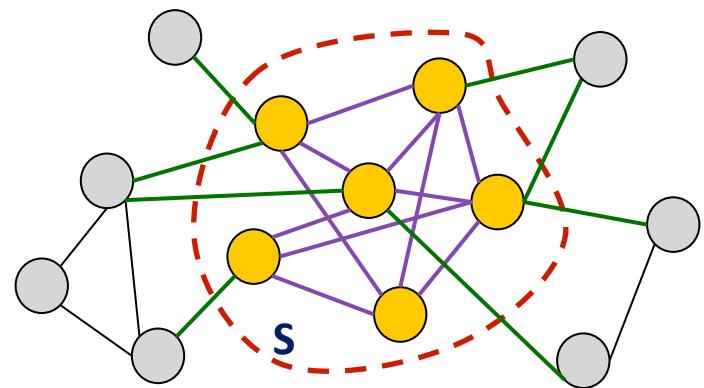


Evaluation based on internal and external connectivity (1)

■ Conductance [Chung '97]

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

Measures the fraction of total edge volume that points outside S



■ Normalized cut [Shi and Malic '00]

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

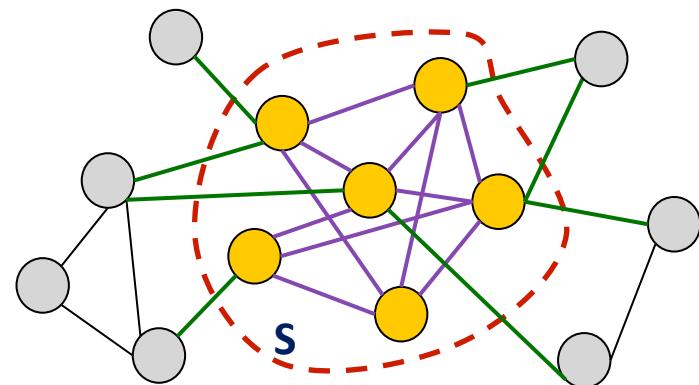
Measures the fraction of total edge volume that points outside S normalized by the size of S

Evaluation based on internal connectivity (3)

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

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

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



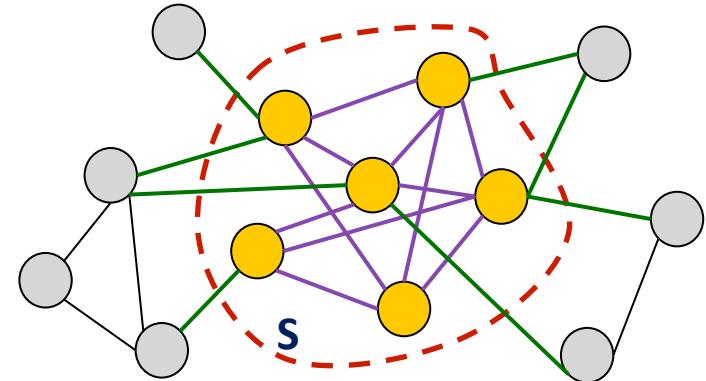
Evaluation based on network model

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

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

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



References (community evaluation measures)

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

References (community evaluation measures)

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

Outline

1. Introduction & Motivation
2. Graph fundamentals
3. Community evaluation measures
- 4. Graph clustering algorithms**
5. Alternative Methods for Community Evaluation
6. New directions for research in the area of graph mining

Notations

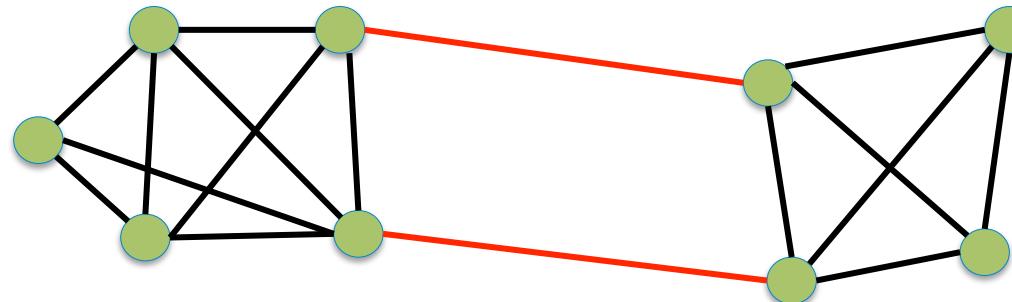
- Given Graph $G=(V,E)$ undirected:

- Vertex Set $V=\{v_1, \dots, v_n\}$, 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 : $d_i = \sum_{v_j \in V} w_{ij}$
- $\nu(V) = \sum_{v_i \in V} d_i$
- for $A \subset V$ $\overline{A} = V - A$
- Given
 - $A, B \subset V$ & $A \cap B = \emptyset$, $w(A, B) = \sum_{v_i \in A, v_j \in B} w_{ij}$
- D : Diagonal matrix where $D(i, i) = d_i$
- W : Adjacency matrix $W(i, j) = w_{ij}$

Graph-Cut

■ For k clusters:

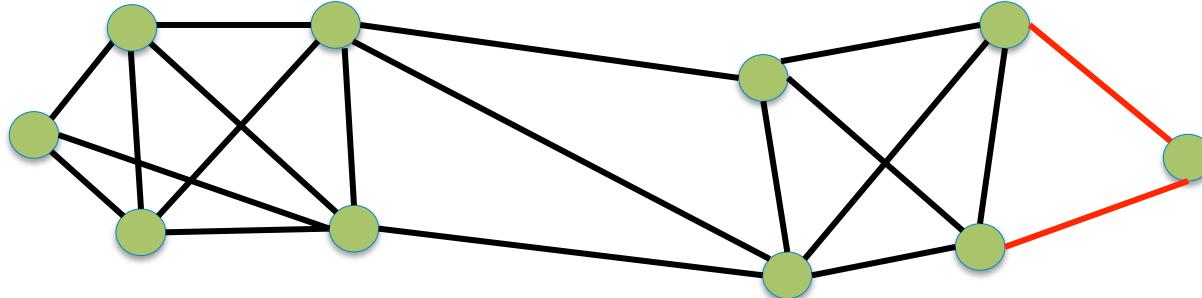
- $cut(A_1, \dots, A_k) = 1/2 \sum_{i=1}^k w(A_i, \overline{A}_i)$
 - undirected graph: 1/2 we count twice each edge



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

Min-Cut

- Easy for $k=2$: $\text{Mincut}(A_1, A_2)$
 - 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):

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

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

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

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

From Graph Cuts to Spectral Clustering

- For simplicity assume $k=2$:

- Define $f: V \rightarrow \mathbb{R}$ for Graph G :

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

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

$$\begin{aligned} & \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2 \\ &= \sum_{v_i \in A, v_j \in \bar{A}} w_{ij} (1 + 1)^2 + \sum_{v_i \in \bar{A}, v_j \in A} w_{ij} (-1 - 1)^2 \\ &= 8 * \text{cut}(A, \bar{A}) \end{aligned}$$

Graph Laplacian

- How is the previous useful in Spectral clustering?

$$\begin{aligned} & \sum_{i,j=1}^n w_{ij}(f_i - f_j)^2 \\ &= \sum_{i,j=1}^n w_{ij}f_i^2 - 2 \sum_{i,j=1}^n w_{ij}f_i f_j + \sum_{i,j=1}^n w_{ij}f_j^2 \\ &= \sum_{i,j=1}^n d_i f_i^2 - 2 \sum_{i,j=1}^n w_{ij}f_i f_j + \sum_{i,j=1}^n d_j f_j^2 \\ &= 2 \left(\sum_{i,j=1}^n d_{ii} f_i^2 - \sum_{i,j=1}^n w_{ij} f_i f_j \right) \\ &= 2(\mathbf{f}^T \mathbf{D} \mathbf{f} - \mathbf{f}^T \mathbf{W} \mathbf{f}) = 2\mathbf{f}^T (\mathbf{D} - \mathbf{W}) \mathbf{f} = 2\mathbf{f}^T \mathbf{L} \mathbf{f} \end{aligned}$$

- \mathbf{f} : a single vector with the cluster assignments of the vertices
- $\mathbf{L} = \mathbf{D} - \mathbf{W}$: 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 $\mathbf{1}$

■ L has n non-negative, real valued eigenvalues

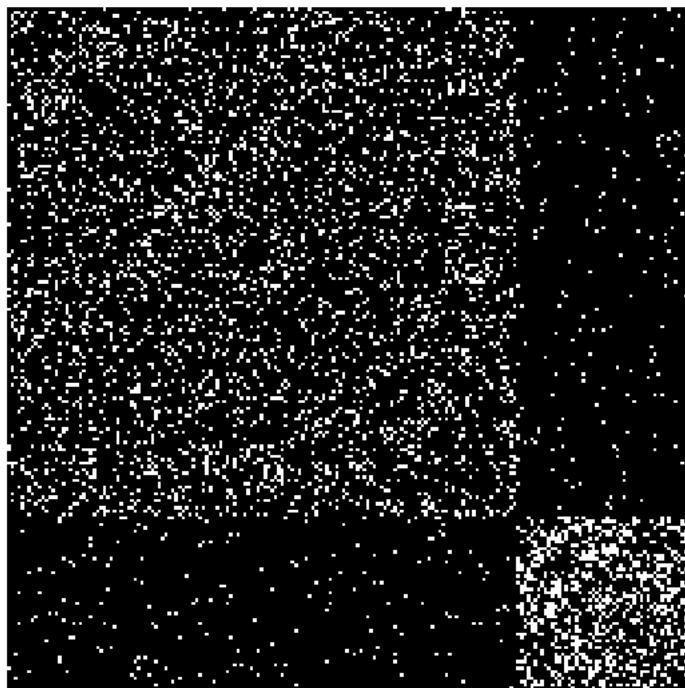
- $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$

Two Way Cut from the Laplacian

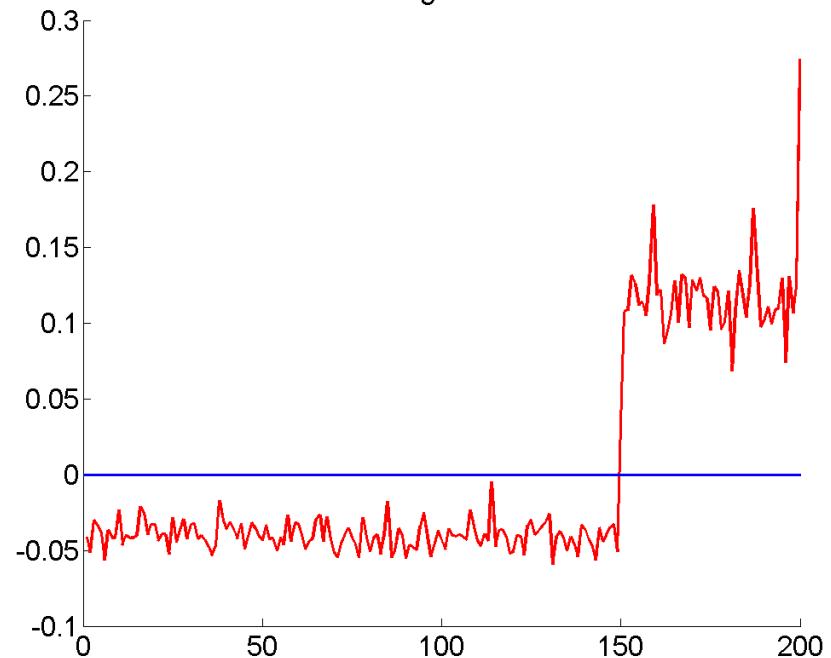
- We could solve $\min_f f^T L f$ where $f \in \{-1,1\}^n$
- NP-Hard for discrete cluster assignments
 - Relax the constraint to $f \in R^n$:
$$\min_f f^T L f \text{ subject to } f^T f = n$$
- The solution to this problem is given by:
 - (**Rayleigh-Ritz Theorem**) the eigenvector corresponding to smallest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

Example

Adjacency Matrix



2nd Eigenvector



Ratio Cut

$$\blacksquare \text{RatioCut}(A_1, \dots, A_k) = \sum_{i=1}^k \frac{\text{cut}(A_i, \bar{A}_i)}{|A_i|}$$

- Define $f: V \rightarrow \mathbb{R}$ for Graph G :

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

- $\sum_{i,j=1}^n w_{ij} (f_i - f_j)^2 = 2\text{cut}(A, \bar{A}) \left(\sqrt{\frac{|\bar{A}|}{|A|}} + \sqrt{\frac{|A|}{|\bar{A}|}} + 2 \right)$
 $= 2|V|\text{RatioCut}(A, \bar{A})$

Ratio Cut

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

$$f^T 1 = \sum_i^n f_i = \sum_{v_i \in A} \sqrt{\frac{|A|}{|A|}} + \sum_{v_i \in \bar{A}} -\sqrt{\frac{|A|}{|\bar{A}|}} = |A| \sqrt{\frac{|A|}{|A|}} - |\bar{A}| \sqrt{\frac{|A|}{|\bar{A}|}} = 0$$
$$f^T f = \sum_i^n f_i^2 = |\bar{A}| + |A| = n$$

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

Normalized Cut

- $Ncut(A_1, \dots, A_k) = \sum_{i=1}^k \frac{cut(A_i, \bar{A}_i)}{v(A_i)}$

- Define $f: V \rightarrow \mathbb{R}$ for Graph G :

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

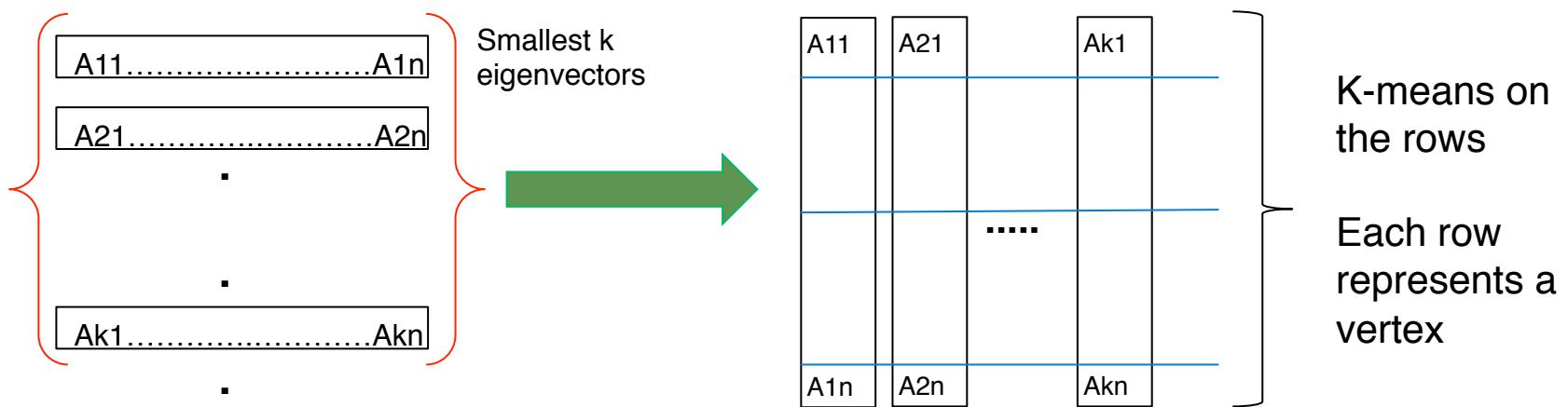
- $\sum_{i,j=1}^n w_{ij} (f_i - f_j)^2 = 2cut(A, \bar{A}) \left(\sqrt{\frac{v(\bar{A})}{v(A)}} + \sqrt{\frac{v(A)}{v(\bar{A})}} + 2 \right)$
 $= 2v(V)Ncut(A, \bar{A})$

Normalized Cut

- Similarly: $\min_f f^T Lf$ subject to
 $f^T D \mathbf{1} = 0, f^T Df = v(V)$
- Assume $h = D^{1/2}f$
 - $\min_h h^T D^{-1/2} L D^{-1/2} h$ subject to
 $h^T D^{1/2} \mathbf{1} = 0, h^T h = v(V)$
 - The answer is in the eigenvector of the second smallest eigenvalue of $L_{sym} = D^{-1/2} L D^{-1/2}$
Shi and Malik (2000)
- L_{sym} is the normalized Laplacian
 - has n non-negative, real valued eigenvalues
 - $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$

Multi-Way Graph Partition

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



References

- 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

Graph Clustering Algorithms

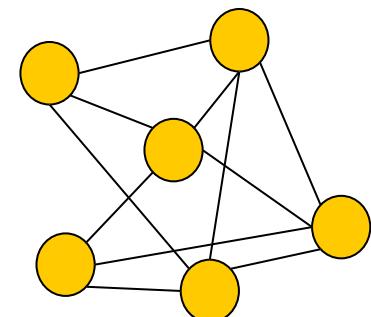
- Spectral Clustering
- Modularity Based Methods

Basics

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

Main idea

- **Modularity** function [Newman and Girvan '04], [Newman '06]
- Initially introduced as a measure for assessing the strength of communities
 - $Q = (\text{fraction of edges within communities}) - (\text{expected number of edges within communities})$
- 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
- $\delta(\cdot)$ 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

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

- 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

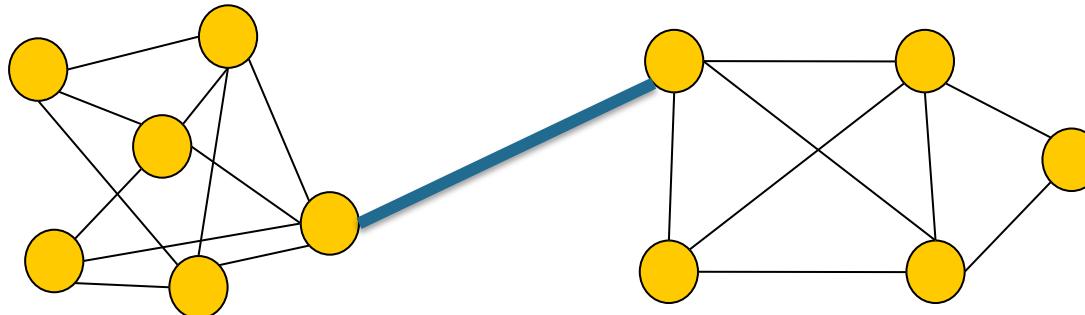
■ Modularity can be applied:

- 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

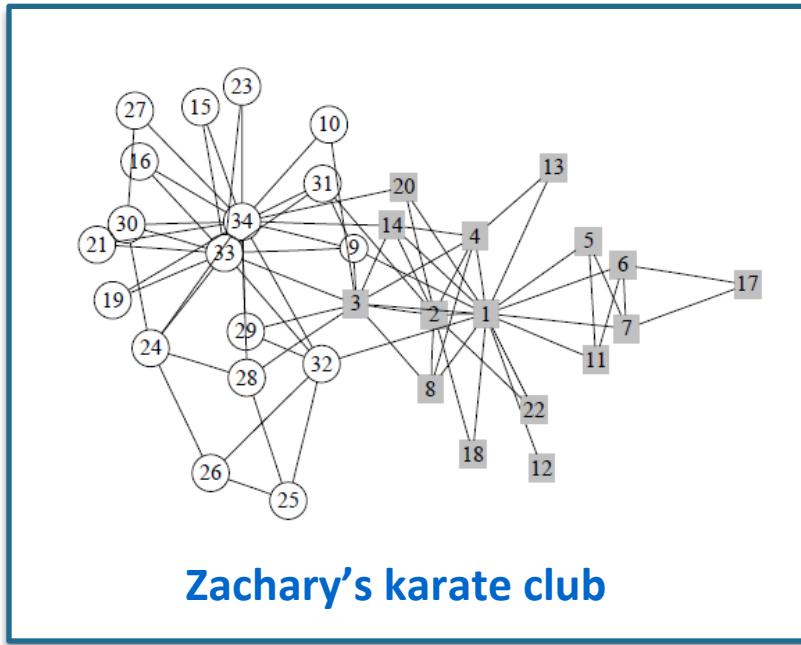
■ How do we know if the produced communities are **good ones** and stop the algorithm?

- 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 ($Q \approx 0.3-0.7$ indicates good partitions)

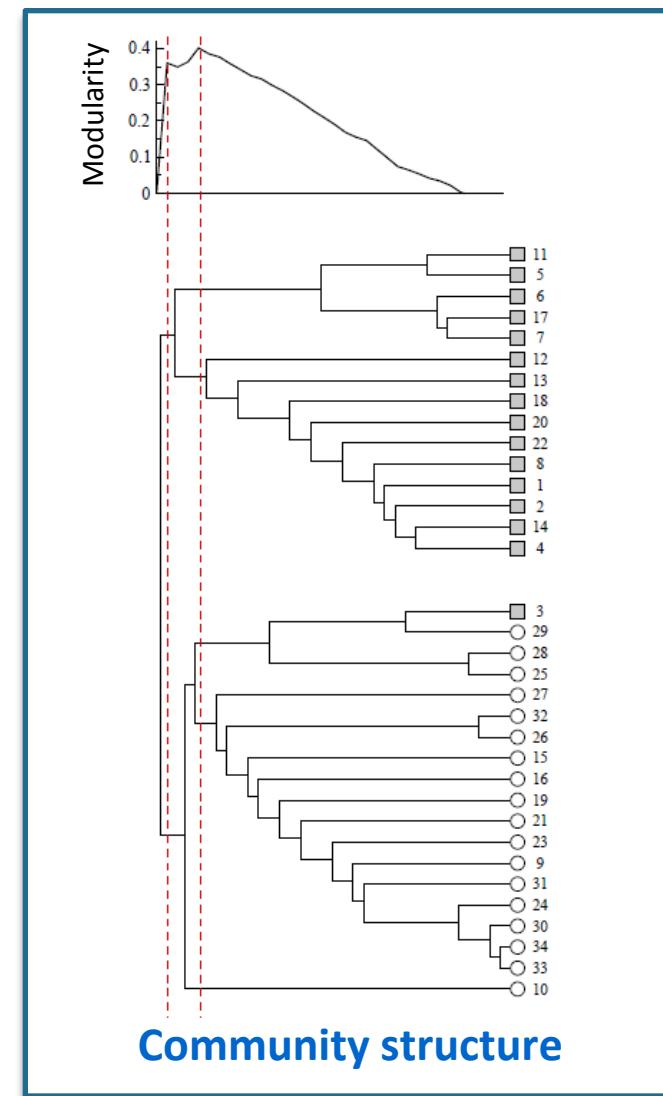
■ Complexity: **$O(m^2n)$** (or **$O(n^3)$** on a sparse graph)

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

Newman-Girvan algorithm (2)



[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]
 - Approximation techniques and heuristics
- Four main categories of techniques
 1. Greedy techniques
 2. **Spectral optimization**
 3. Simulated annealing
 4. Extremal optimization

[Fortunato '10]

Spectral optimization (1)

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

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

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

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

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

Spectral optimization (2)

- Modularity matrix \mathbf{B}

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

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

$$\mathbf{s} = \sum_i a_i \mathbf{u}_i \quad \text{where} \quad a_i = \mathbf{u}_i^T \mathbf{s}$$

- Modularity can now expressed as

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

Where β_i is the eigenvalue of \mathbf{B} corresponding to eigenvector \mathbf{u}_i

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

Spectral optimization (3)

■ Spectral modularity optimization algorithm

1. Consider the eigenvector \mathbf{u}_1 of \mathbf{B} corresponding to the largest eigenvalue
2. Assign the nodes of the graph in one of the two communities \mathbf{X} ($s_i = +1$) and \mathbf{Y} ($s_i = -1$) based on the **signs** of the corresponding components of the eigenvector

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

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

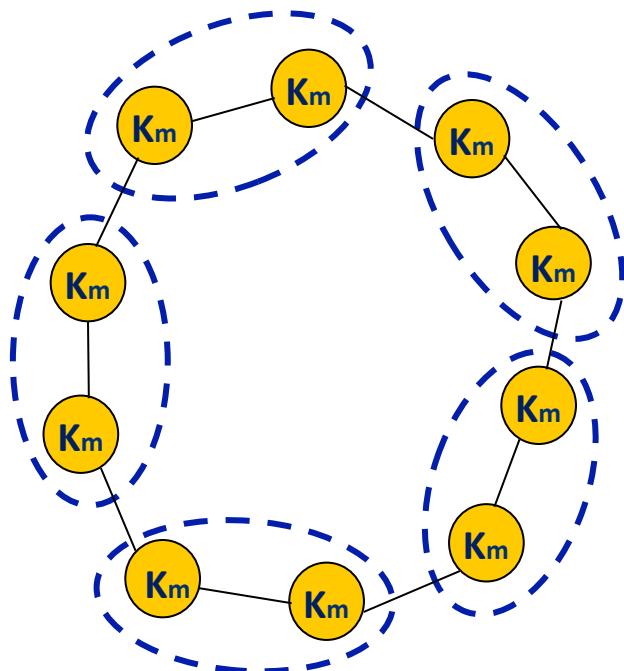
[Newman '06], [Newman '06b]

Extensions of modularity

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

Resolution limit of modularity

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



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

References (modularity)

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

References (modularity)

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