Graph Communities

Sarunas Girdzijauskas

Recap

- **Clustering**
  - Hierarchical Clustering
  - K-means
  - BFR algorithm
  - CURE algorithm
- **Label Propagation**
  - Network Classification
  - Community Detection

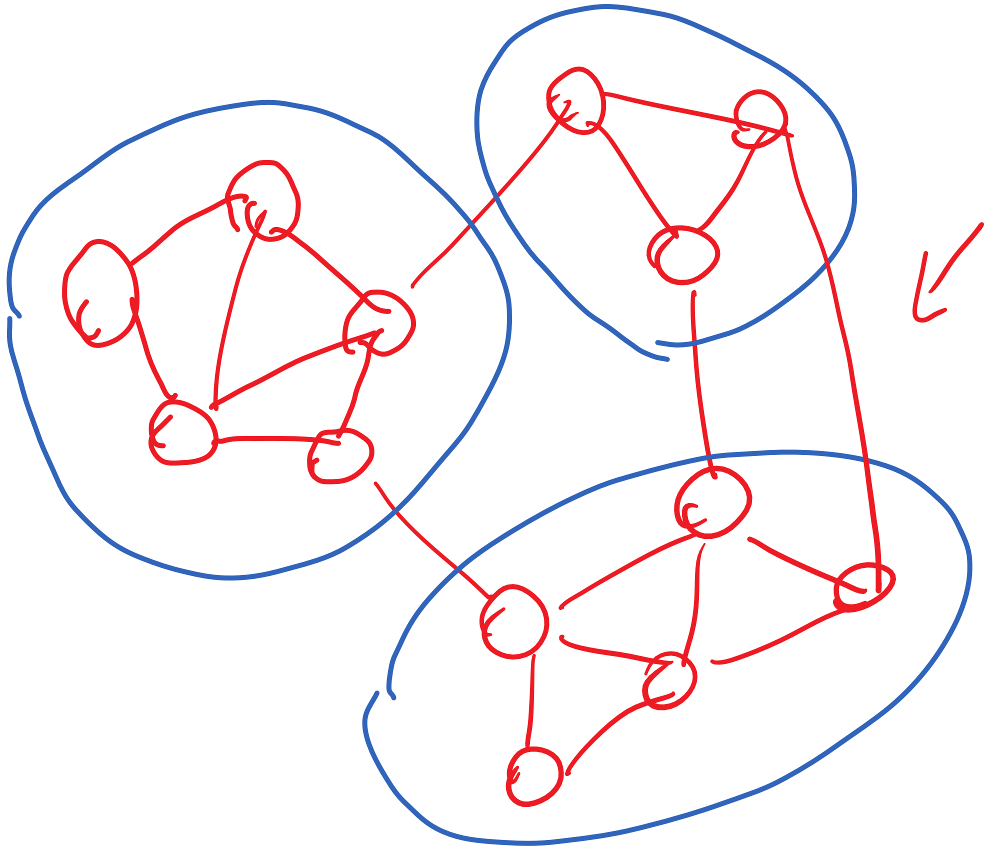
What if we do not have any labels – just the graph structure?

- What are clusters and communities in graphs? **Any ideas?**

- The notion of **community structure** captures the tendency of nodes to be organized into modules (communities, clusters, groups)
  - Members within a community are **more similar** among each other
- Typically, the communities in graphs (networks) correspond to **densely connected** entities (nodes)
- Set of nodes with **more/better/stronger** connections between its members, than to the rest of the network

This and following slides are ”stolen” from WWW2013 tutorial by Christos Giatsidis

Example



what if
we add
this?

Clusters and Communites in Graphs

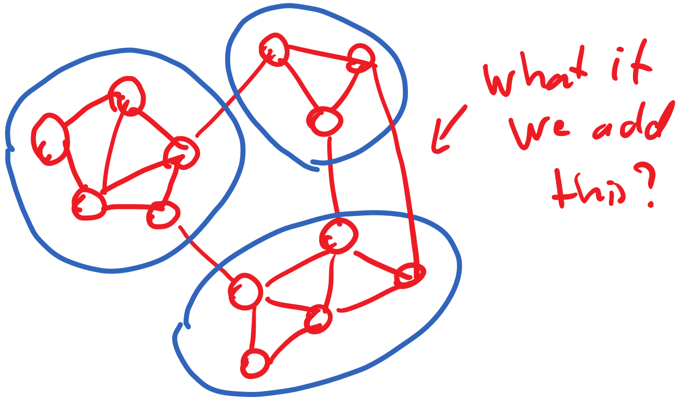
- There is no widely accepted single definition
  - It **depends heavily on the application** domain and the **properies of the graph**
  - Mostly we talk about **sparse bidirectional graphs**
  - But we could also have **dense, weighted, directional**
- Usually NP-hard problems

Definition/notion of Communities

- **Take FB graph. How would you define a ”good community”?**
- Most widely used notion of communities is based on the **number of edges within a group** (density) compared to the **number of edges between different groups**
  - *i.e., A community corresponds to a group of nodes with more **intra-cluster edges** than **inter-cluster edges***

How do we extract the communities?

- 1. Define a **quality measure** (evaluation measure, objective function) that quantifies the desired properties of communities
  - *What could be such measures?* ***Any ideas?***
  - *How do they relate to general clustering?*
- 2. Apply **algorithmic techniques** to assign the nodes of graph into communities, optimizing the objective function

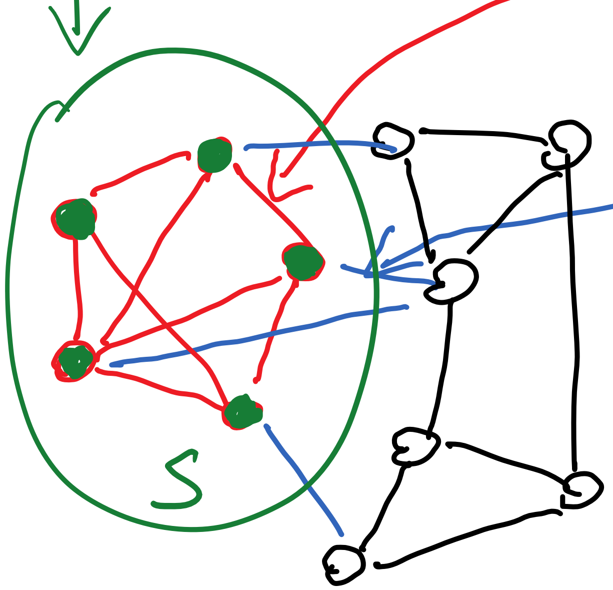


Community Evaluation measures

- We group the community evaluation measures according to
  - Evaluation based on **internal** connectivity (#of edges within community)
  - Evaluation based on **external** connectivity (# of edges across communities)
  - Evaluation based on **internal and external** connectivity
  - Evaluation based on **network model**

Recap on Notations

- **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



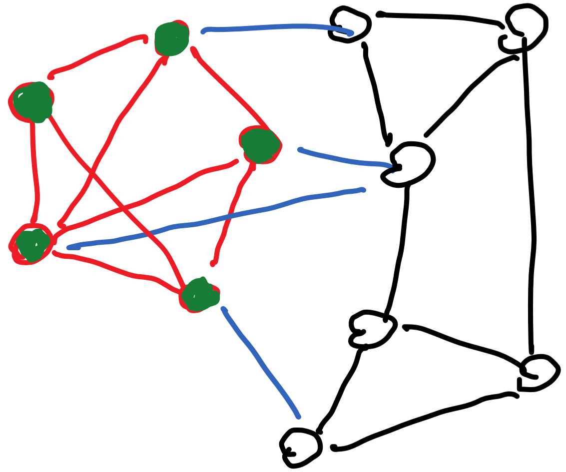
Internal Connectivity

- Edges inside
  - f(S) = m_s

Do you see any issue with this measure? Can you improve it?

- Internal Density

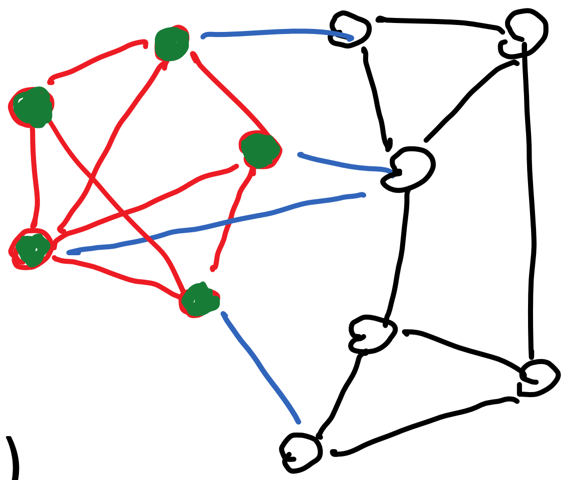
$$f(S) = \frac{m_s}{n_s(n_s-1)/2}$$



Internal Connectivity (cont.)

- Average degree
  - Average internal degree of nodes in S

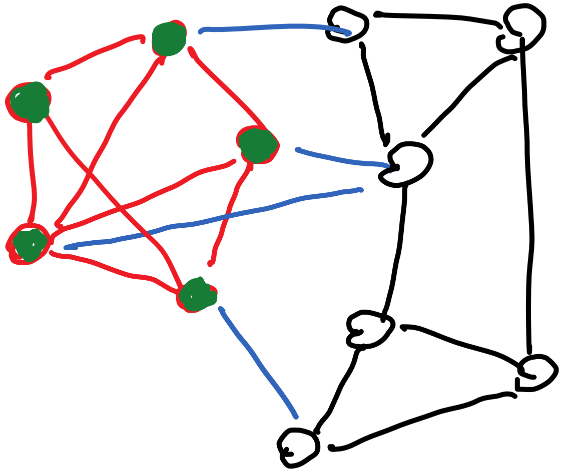
$$f(S) = \frac{2m_s}{n_s}$$



- Fraction over median degree (FOMD)
  - fraction of nodes in S with internal degree greater than d_m, where d_m =median degree of the whole graph

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

Internal Connectivity (cont. 2)



- Triangle participation ratio
  - Fraction of nodes in S that belong to a triangle

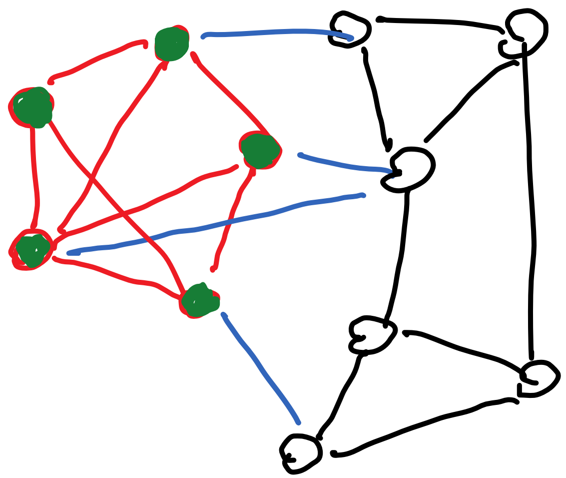
$$f(S) = \frac{\left|\{u : u \in S, \{(v,w) : v,w \in S, (u,v) \in E, (u,w) \in E, (v,w) \in E\} \neq \emptyset\}\right|}{n_s}$$

External Connectivity

- Any examples?
- Expansion
  - The number of edges per node that point outside S

$$f(S) = \frac{c_S}{n_S}$$

- Cut Ratio
  - Fraction of existing edges out of all possible edges leaving S



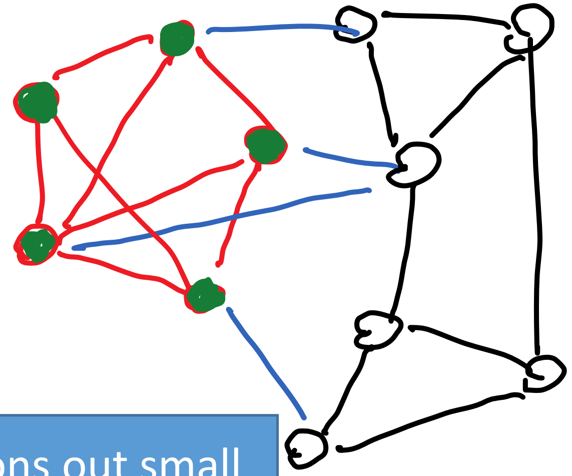
$$f(S) = \frac{c_S}{n_S(n - n_S)}$$

External and Internal Connectivity

- ***Ideas?***
- Conductance

The fraction of total edge volume that points outside S

$$f(S) = \frac{C_S}{2m_S + C_S}$$



The cut that partitions out small isolated points will no longer have small value

- Normalized Cut

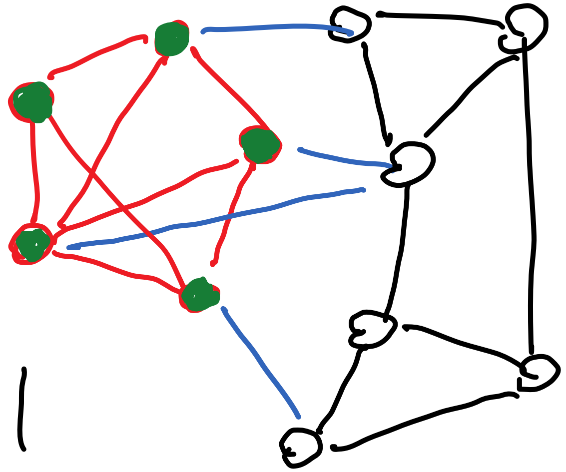
$$f(S) = \frac{C_S}{2m_S + C_S} + \frac{C_S}{2(m - m_S) + C_S}$$

External and Internal Connectivity (cont.)

- Average out degree fraction
  - The average fraction of edges of nodes in S

that point outside S

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



Evaluation based on network model

- Modularity Q
  - Measures the difference between the number of edges in S and the expected number of edges in a random graph model with the same degree sequence

Modularity matrix B (not sparse):

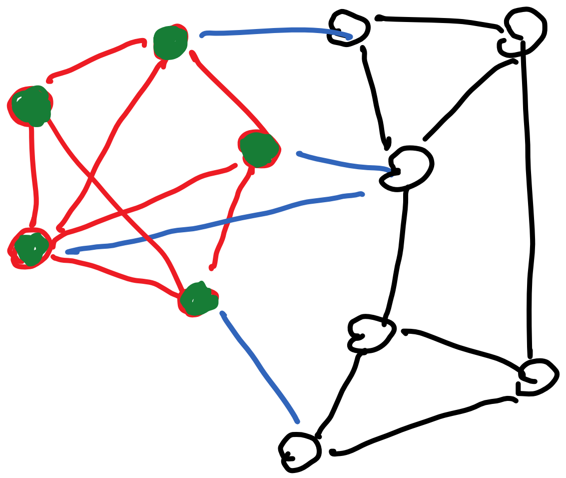
$$B_{ij} = A_{ij} - \frac{d_i d_j}{2m}$$

Modularity Q is given by the sum $A_{ij} - \frac{d_i d_j}{2m}$ of all pairs of vertices i, j that fall in the same cluster

can conveniently be written in matrix form as

$$Q = \frac{1}{4m} \mathbf{s}^T \mathbf{B} \mathbf{s},$$

where **s** is the column vector whose elements are the s_i=*1* if node *i* belongs to S and 0 otherwise.



Sum up all the Values in B ("edges") which belong to the cluster and normalize by all edges

Which evaluation measure to use?

- **Your ideas?**

- Consider real graphs with **known node assignment to communities (ground-truth information)** and test the behavior of the objective measures **[Yang and Leskovec '12]**
  - Compute scores for each ground-truth community based on aforementioned objective measures and check their accuracy.
  - **Conductance** and **Triad-participation-ratio** give the best performance in identifying 230 ground-truth communities from social, collaboration and information networks [Yang and Leskovec 2012]

- In the end it all the depends on the problem that you are solving
  - E.g., malicious attacker trying to isolate max number of nodes with least cuts (remember expanders)?
  - *E.g., how many "valid communities" could you extract from a FB graph?*

- Remark: The naming is sometimes ambiguous, e.g., conductance vs normalized cut.

Spectral Clustering

Spectral Clustering

- **Let's describe our problem with A *n x n* similarity/affinity matrix**
  - A_{ij} is similarity or affinity between some objects, nodes, data points etc.
  - can be sparse or not
  - All values non-negative $a_{ij}>=0$, for all i and j.
  - Similarity could be 0 to 1 (but anything works, and higher value implies greater similarity)
    - Highest values would be on the diagonal (most similar to itself)
  - **Symmetric matrix**
    - We assume our notion of similarity as symmetric in this class.
  - A is positive semi-definite
    - $x^TAx>=0$
    - A has n real non-negative eigenvalues

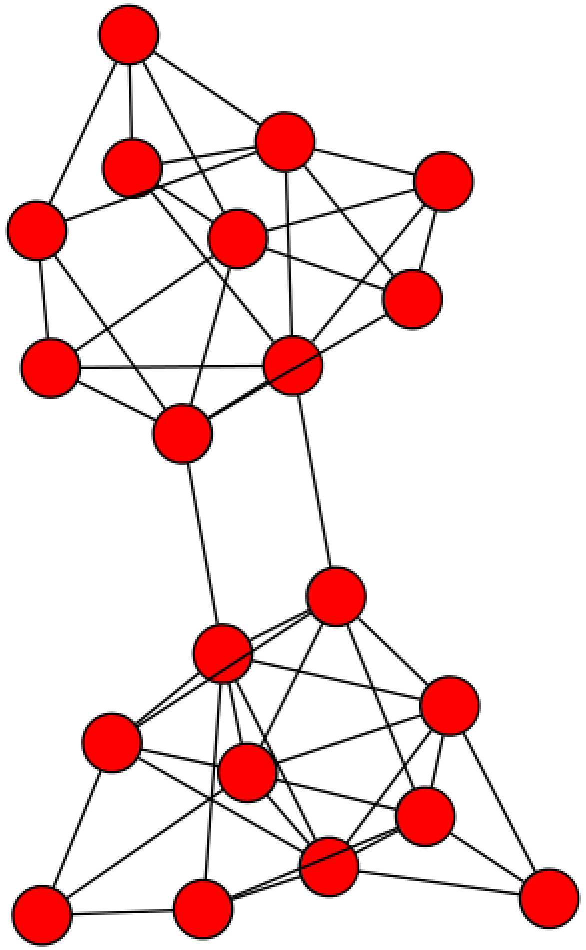
Affinity Matrix: Issues

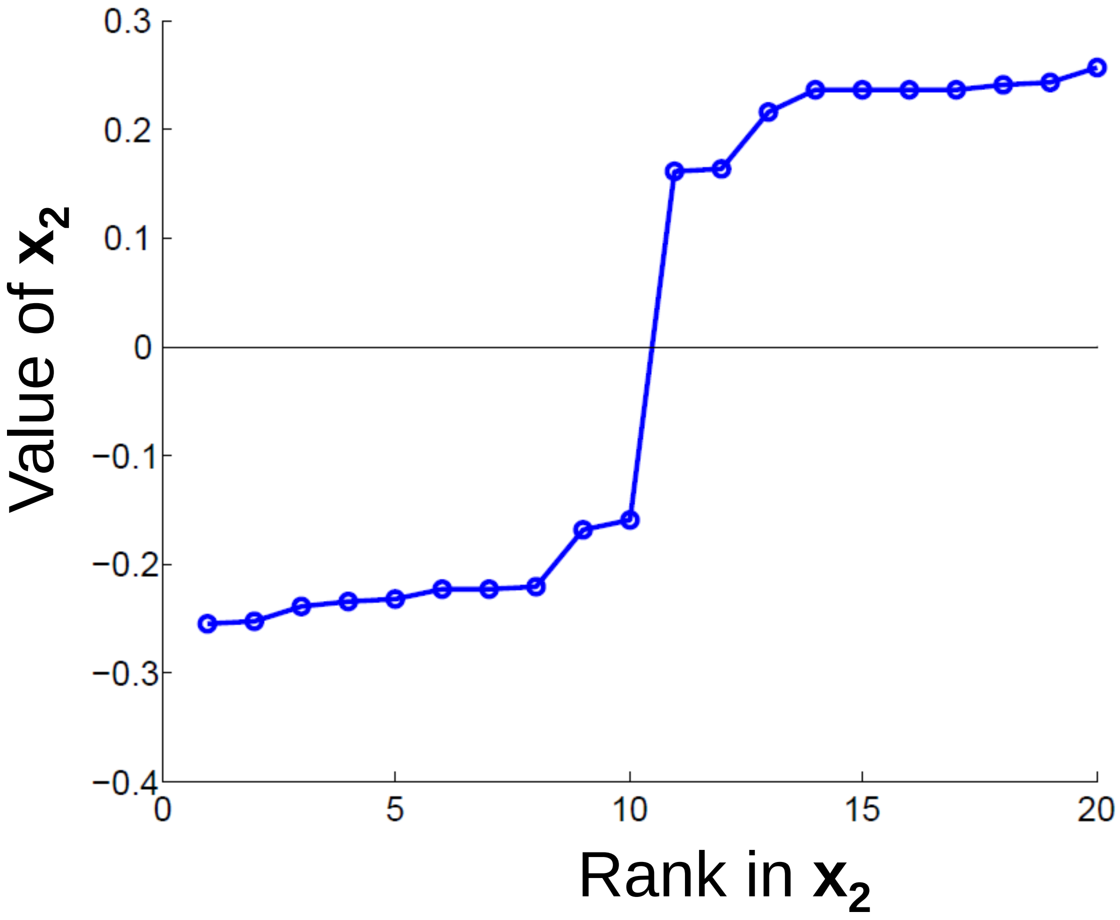
- How do we describe affinities?
  - Any ideas?
  - How strong is relationship, Cosine similarity between vectors, dot product etc.
- Should you use all the matrix (dense)?
  - Any ideas?
  - $O(N^2)$ cost to process
  - Fix with:
    - Thresholding
    - Keep k nearest neighbors (knn), but keep it symmetric!
    - **Graph interpretation**: Moving from fully connected graph to a sparse graph.
      - Affinity matrix vs adjacency matrix (weighted)

Spectral Partitioning

- Spectra of A helps to optimize for ”**avg. degree/weight**”
  - Look at the **eigenvectors** associated to the **largest eigenvalues**
- Spectra of D-A (Laplacian) Helps to optimize for ”**ratio cut**”
  - Look at the **eigenvectors** associated to the **smallest eigenvalues (non-zero)**
    - **2nd eigenvector – Fiedler vector (corresponds to second smallest eigenvlaue)**
    - 1st eigeinvalue is always zero
- Spectra of $D^{-½}A\ D^{-½}$ (normalized Laplacian) helps to optimize for ”**conductance**”
  - Look at the **eigenvectors** associated to the **largest eigenvalues**
- **Laplacian** L = D-A
- **Normalized Laplacian =** $D^{-½}A\ D^{-½}$
  - Sometimes also known as $I\text{-}D^{-½}A\ D^{-½}$

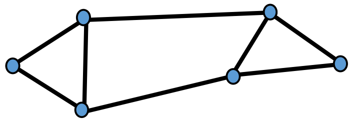
Example: Spectral Partitioning – simple Bisection (using Fiedler vector)





Spectral Partitioning Algorithm for ratio cut

- 1) **Pre-processing:**
  - Build Laplacian matrix L of the graph



	1	2	3	4	5	6
1	3	-1	-1	0	-1	0
2	-1	2	-1	0	0	0
3	-1	-1	3	-1	0	0
4	0	0	-1	3	-1	-1
5	-1	0	0	-1	3	-1
6	0	0	0	-1	-1	2

- 2) **Decomposition:**
  - Find eigenvalues λ and eigenvectors x of the matrix L
  - Map vertices to corresponding components of λ_2

λ=

0.0
1.0
3.0
3.0
4.0
5.0

X =

0.4	0.3	-0.5	-0.2	-0.4	-0.5
0.4	0.6	0.4	-0.4	0.4	0.0
0.4	0.3	0.1	0.6	-0.4	0.5
0.4	-0.3	0.1	0.6	0.4	-0.5
0.4	-0.3	-0.5	-0.2	0.4	0.5
0.4	-0.6	0.4	-0.4	-0.4	0.0

1	0.3
2	0.6
3	0.3
4	-0.3
5	-0.3
6	-0.6

How do we now find the clusters?

Spectral Partitioning

- **3) Grouping:**
  - Sort components of reduced 1-dimensional vector
  - Identify clusters by splitting the sorted vector in two
- **How to choose a splitting point?**
  - Naïve approaches:
    - Split at **0** or median value

1	0.3
2	0.6
3	0.3
4	-0.3
5	-0.3
6	-0.6

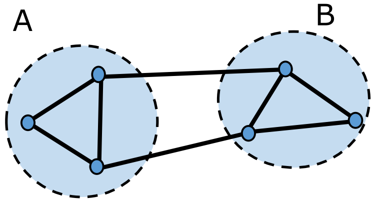
Split at 0:

Cluster A: Positive points

Cluster B: Negative points

1	0.3
2	0.6
3	0.3

4	-0.3
5	-0.3
6	-0.6



k-Way Spectral Clustering

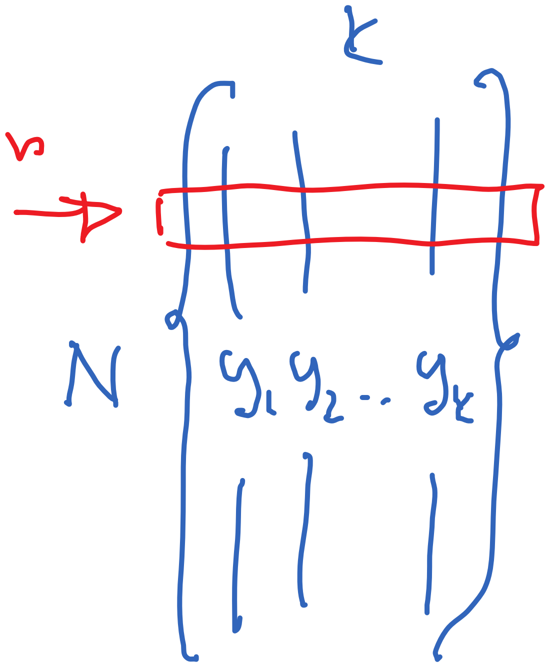
- **So far we considered bisection only.**
  - **How do we partition a graph into k clusters? Any ideas?**

- **Two basic approaches:**
  - **Recursive bi-partitioning** [Hagen et al., '92]
    - Recursively apply bi-partitioning algorithm in a hierarchical divisive manner
    - Disadvantages: Inefficient, unstable
  - **Cluster multiple eigenvectors** [Shi-Malik, '00]
    - Build a reduced space from multiple eigenvectors
    - Commonly used in recent papers
    - A preferable approach…

Cluster multiple eigenvectors

- You end up with n x k matrix (k eigenvectors)
- Take matrix of eigenvectors

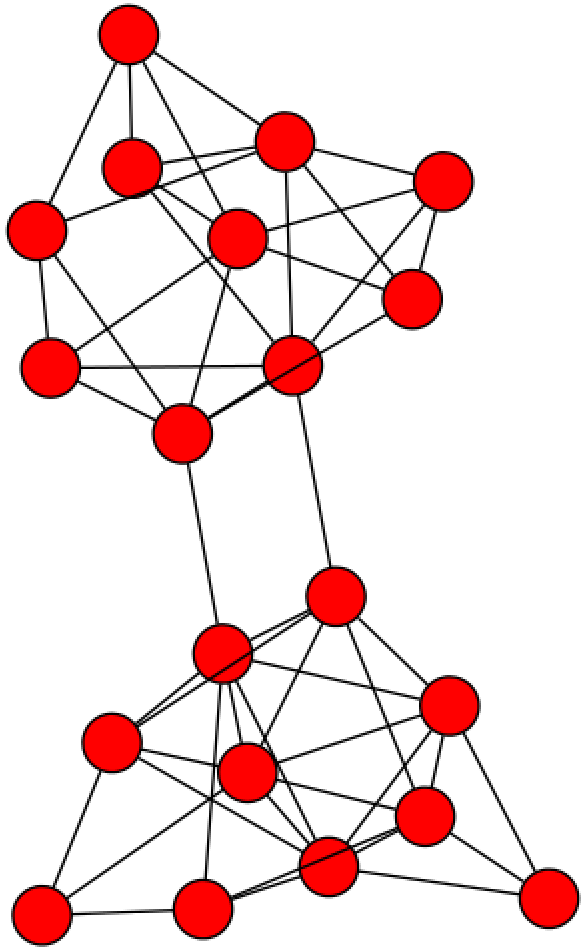
Treat as a point in k-dim space

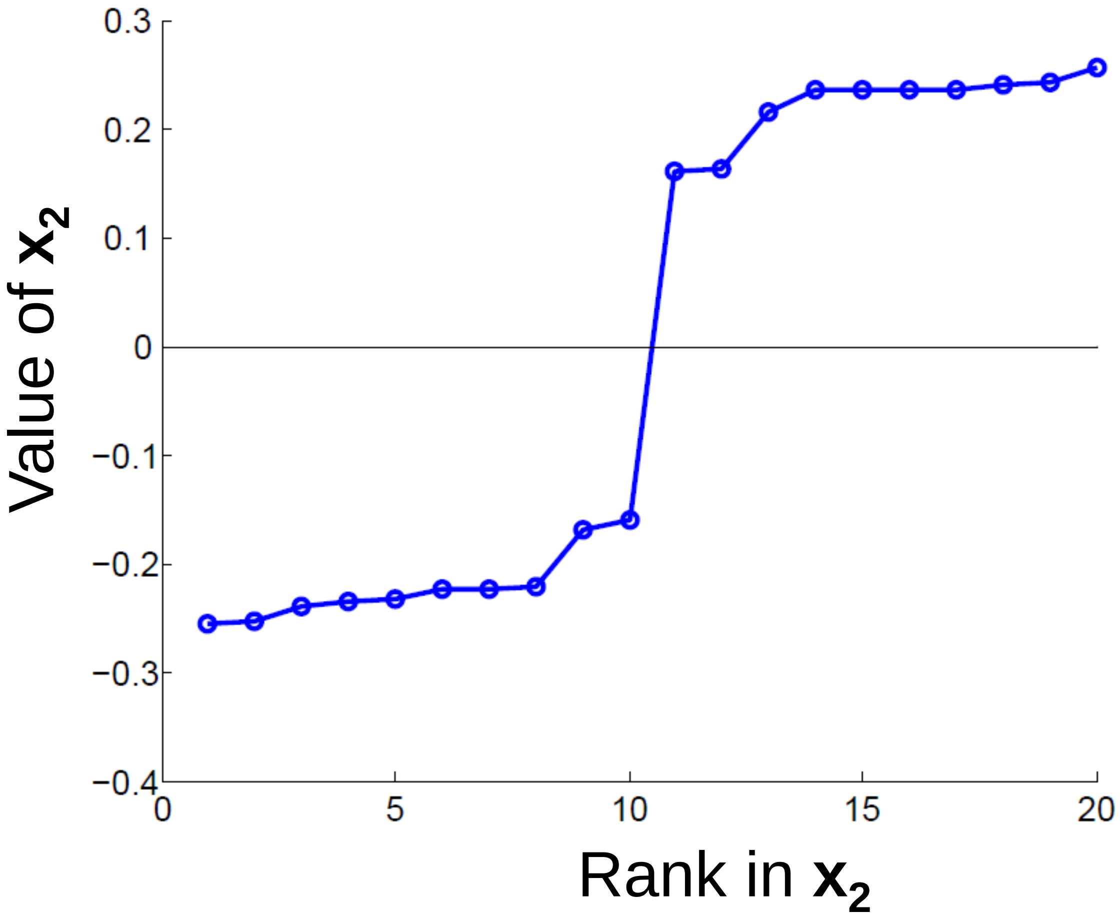


- Think of
  - We transformed n points in n dimensional space to n points in k dimensional space
  - Run simple clustering (e.g., k-means) to find final clusters
  - We expect that in this k-dim eigenspace the points will become quite distinct

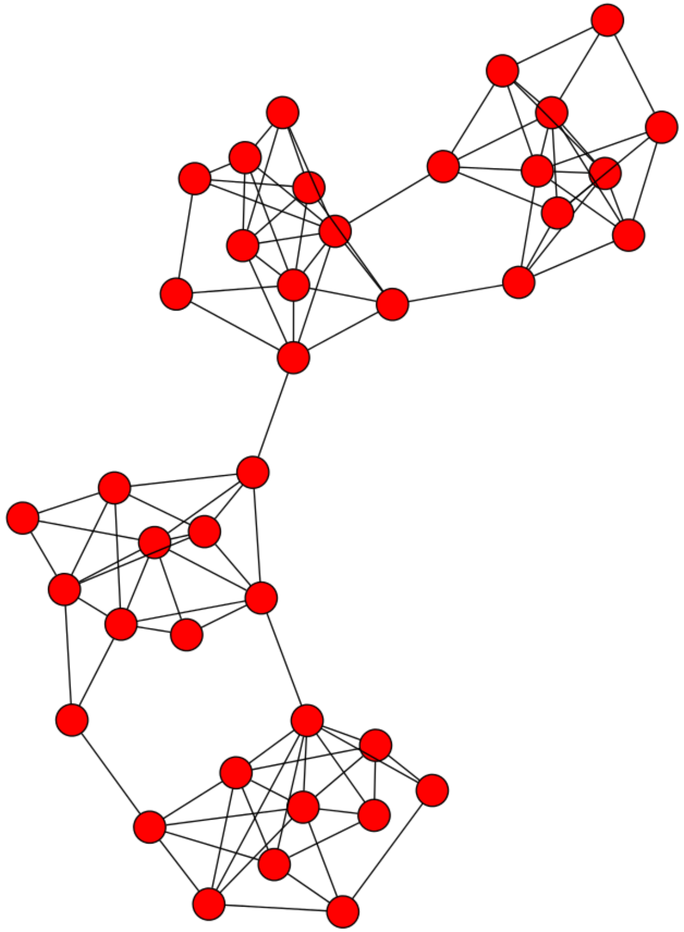
easy to separate

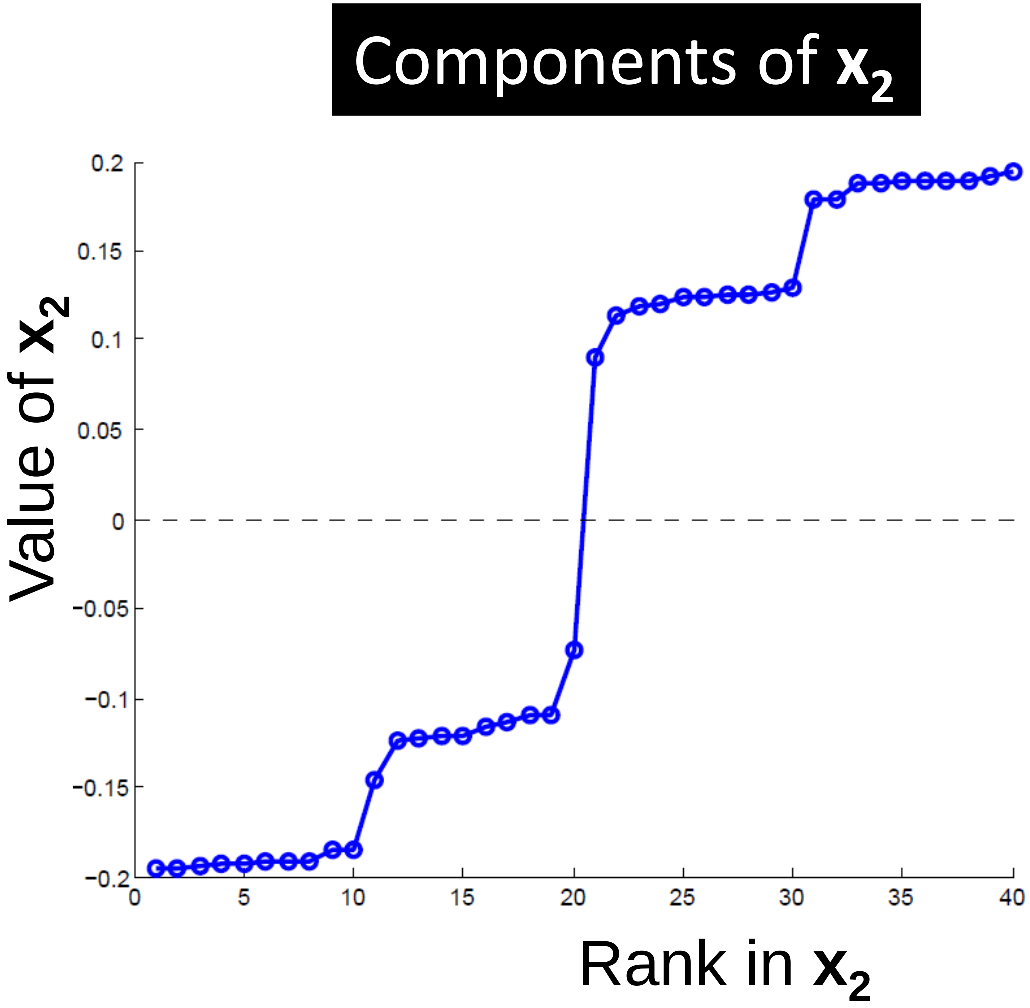
Example: Spectral Partitioning



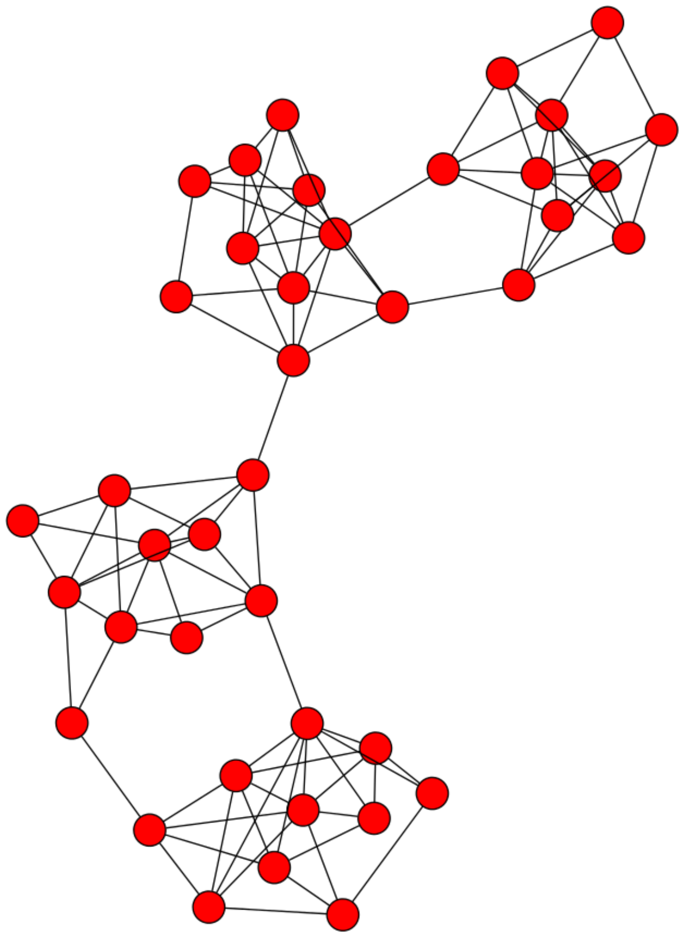


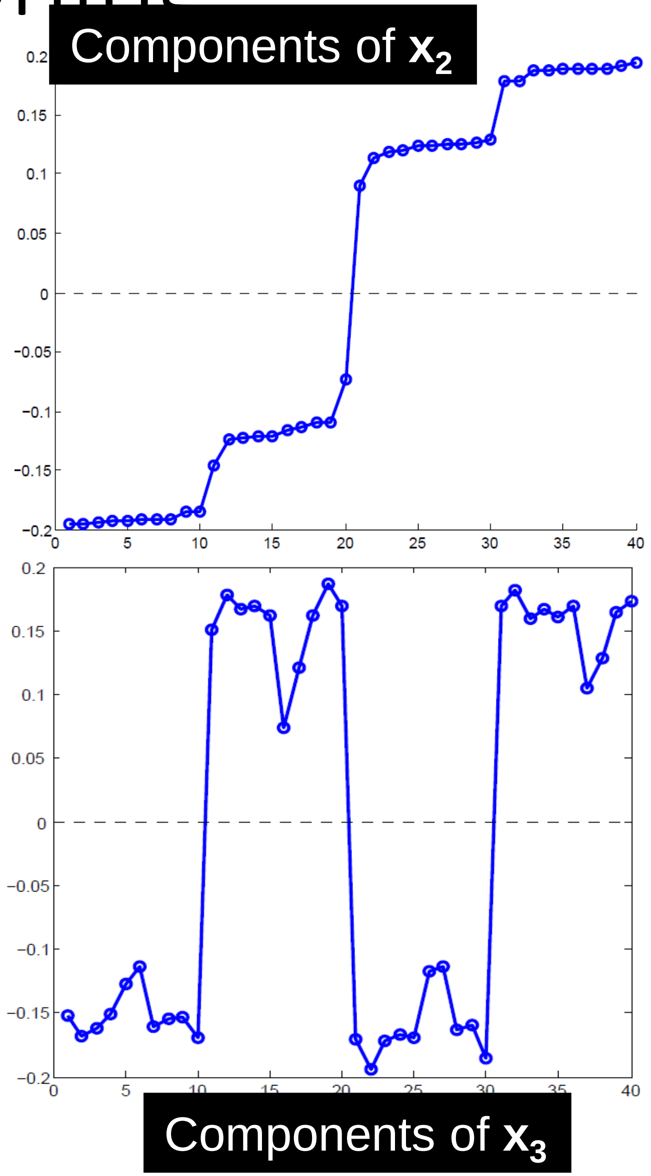
Example: Spectral Partitioning





Example: Spectral partitioning



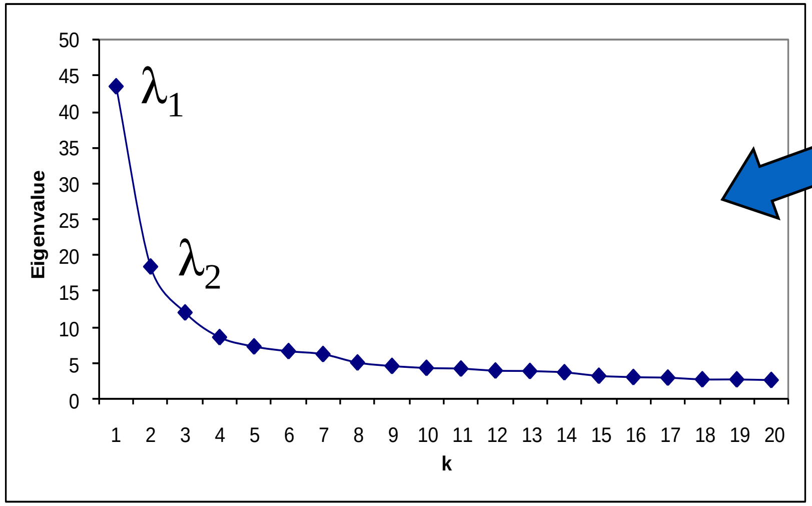


How to select k?

- **Eigengap:**
  - The difference between two consecutive eigenvalues
- **Most stable clustering is generally given by the value k that maximizes eigengap Δ_k:**

$$\Delta_k = |\lambda_k - \lambda_{k-1}|$$

- **Example:**



$$\max \Delta_k = |\lambda_2 - \lambda_1|$$

$\Rightarrow$ **Choose $k = 2$**

See Matlab example

More tricks: Community Detection

Modularity matrix B (not sparse):

$$B_{ij} = A_{ij} - \frac{d_i d_j}{2m}$$

- We want to find two natural communities in a graph G.
  - Calculate eigenvector of the largest (most positive) eigenvalue of the modularity matrix B.
  - Assign vertices to communities according to the signs of the vector elements:
    - Positive signs in one

    group and negative into other

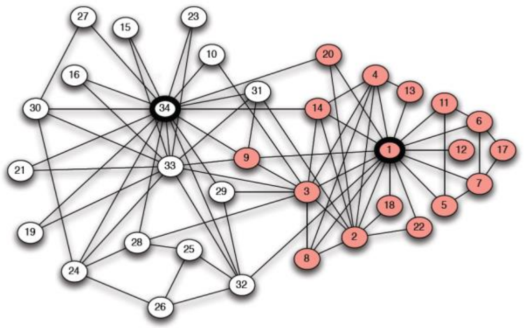


Figure 1.7: From the social network of friendships in the karate club from Figure 1.1, we can find clues to the latent schism that eventually split the group into two separate clubs (indicated by the two different shadings of individuals in the picture).

Recap.

- Many community evaluation measures!
- We focus on four classes:
  - Evaluation based on **internal** connectivity
    - E.g., avg. degree
  - Evaluation based on **external** connectivity
    - E.g., ratio cut.
  - Evaluation based on **internal and external** connectivity
    - E.g., Conductance
  - Evaluation based on **network model**
    - Modularity

Cluster using **eigenvectors** associated to the **largest eigenvalues** of matrix A

Cluster using **eigenvectors** associated to the **smallest eigenvalues** of laplacian L

Cluster using **eigenvectors** associated to the **largest eigenvalues** of normalized Laplacian $D^{-½}A\ D^{-½}$

Cluster using **eigenvectors** associated to the **largest eigenvalues** of modularity matrix B