Spectral Methods

for Network Community Detection and Graph Partitioning

M. E. J. Newman
Department of Physics, University of Michigan

Presenters: Yunqi Guo
Xueyin Yu
Yuanqi Li

Outline:

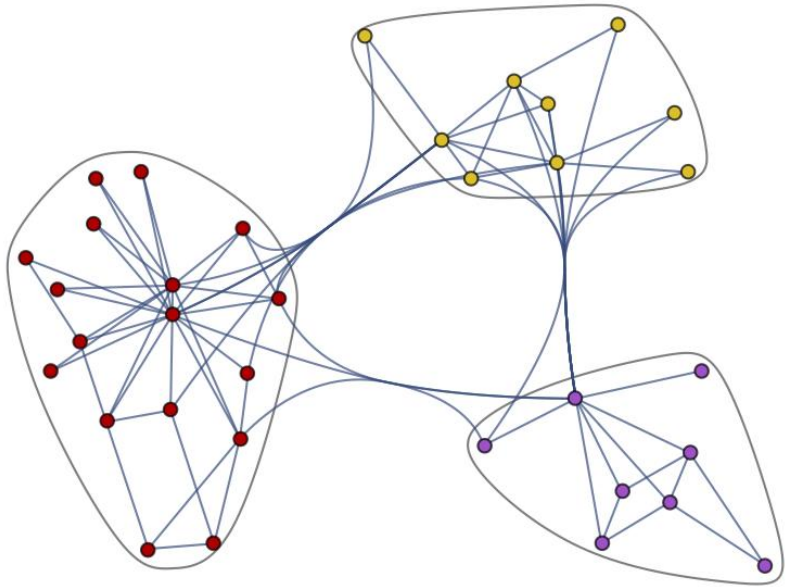
- Community Detection
  - Modularity Maximization
  - Statistical Inference
- Normalized-cut Graph Partitioning
- Analysis and Evaluation
  - Spectral Clustering vs K-means
- Conclusions
- Discussion/Q&A

Community Detection/Clustering

Community

a.k.a. Group, Cluster, Cohesive Subgroup, Module

It is formed by individuals such that those within a group interact with each other more frequently than with those outside the group.



Community Detection

Discovering groups in a network where individuals' group memberships are not explicitly given.

Community Detection Applications

- To detect suspicious events in Telecommunication Networks
- Recommendation Systems
- Link Prediction
- Detection of Terrorist Groups in Online Social Networks
- Lung Cancer Detection
- Information Diffusion
- ……

Methods for Finding Communities

- Minimum-cut method
- Hierarchical clustering
- Girvan–Newman algorithm
- Modularity maximization
- Statistical inference
- Clique-based methods

Modularity Maximization

Modularity

The fraction of edges within groups minus the expected fraction of such edges in a randomized null model of the network.

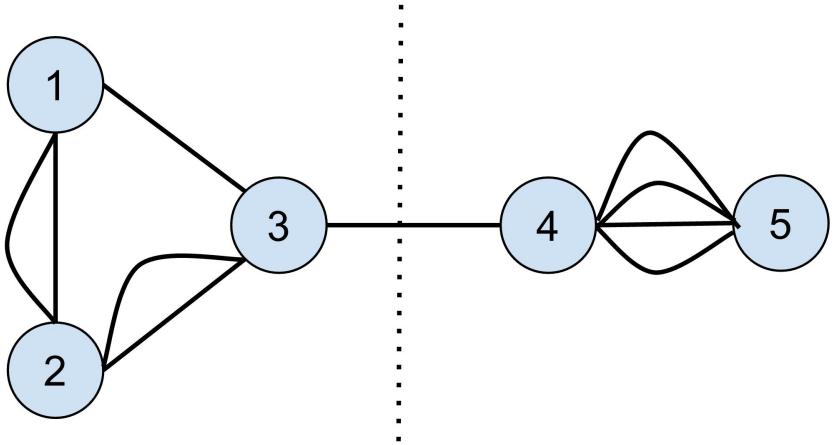
$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta_{g_i g_j}$$

$$\delta_{g_i g_j} = \frac{1}{2}(s_i s_j + 1)$$

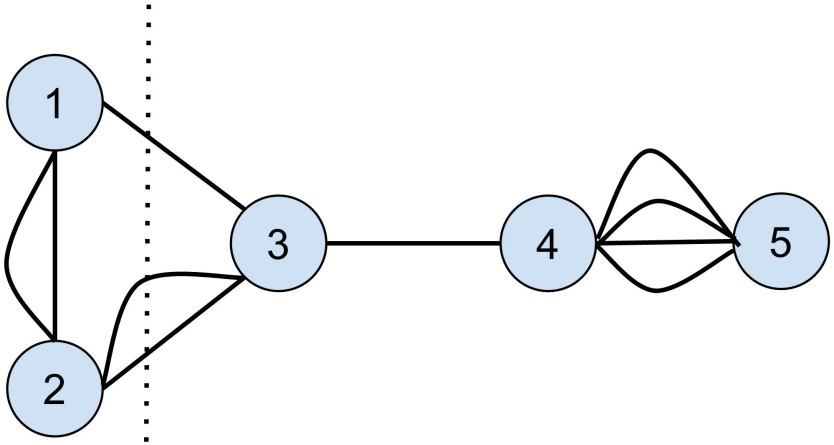
$$s_i = \begin{cases} +1 & \text{if vertex } i \text{ belongs to group 1} \\ -1 & \text{if vertex } i \text{ belongs to group 2} \end{cases}$$

A : adjacency matrix
k_i : the degree of vertex i
m : the total number of edges in the observed network
δ_{ij} : the Kronecker delta

Modularity



Q=0.79



Q=0.31

Modularity

The fraction of edges within groups minus the expected fraction of such edges in a randomized null model of the network.

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta_{g_i g_j}$$

$$\delta_{g_i g_j} = \frac{1}{2}(s_i s_j + 1)$$

$$s_i = \begin{cases} +1 & \text{if vertex } i \text{ belongs to group 1} \\ -1 & \text{if vertex } i \text{ belongs to group 2} \end{cases}$$

A : adjacency matrix
k_i : the degree of vertex i
m : the total number of edges in the observed network
δ_{ij} : the Kronecker delta

Lagrange Multiplier

maximize $f(x, y)$

subject to $g(x, y) = \mathrm{c}.$

Lagrange Function:

$$\mathcal{L}(x, y, \lambda) = f(x, y) - \lambda \cdot (g(x, y) - c)$$

Stationary Point:

$$\nabla_{x,y,\lambda} \mathcal{L}(x, y, \lambda) = 0$$

For *n* variables:

$$\nabla_{x_1,\ldots,x_n,\lambda} \mathcal{L}(x_1, \ldots, x_n, \lambda) = 0$$

Eigenvector and Eigenvalue

Square matrix: **A**

Column vector: **v**

$$Av = \lambda v$$

v : eigenvector

λ : eigenvalue

Generalized Eigenvector Equation

A generalized eigenvector of an n × n matrix **A** is a vector which satisfies certain criteria which are more relaxed than those for an (ordinary) eigenvector.

e.g.

$$Av = \lambda Dv$$

Spectral Clustering

Spectral clustering techniques make use of the spectrum (eigenvalues) of the similarity matrix of the data to perform dimensionality reduction before clustering in fewer dimensions.

Normalized Laplacian:

$$L = D^{-1/2} A D^{-1/2}$$

Result

$$\mathbf{As} = \lambda \mathbf{Ds}$$

D: the diagonal matrix with elements equal to the vertex degrees $D_{ii} = k_i$

S : "Ising spin" variables.

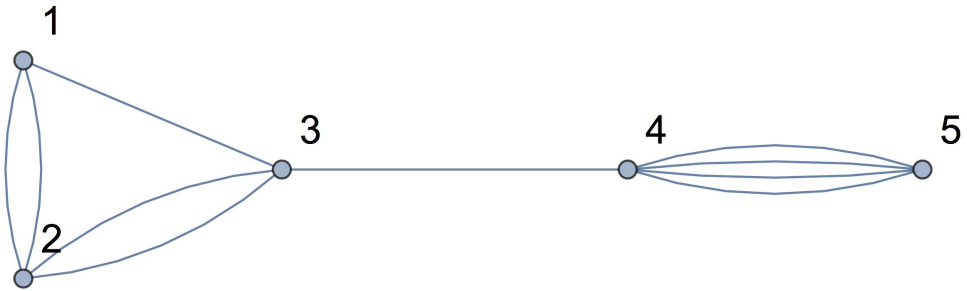
$$s_i = \begin{cases} +1 & \text{if vertex } i \text{ belongs to group 1,} \\ -1 & \text{if vertex } i \text{ belongs to group 2.} \end{cases}$$

$$Lu = \lambda u$$

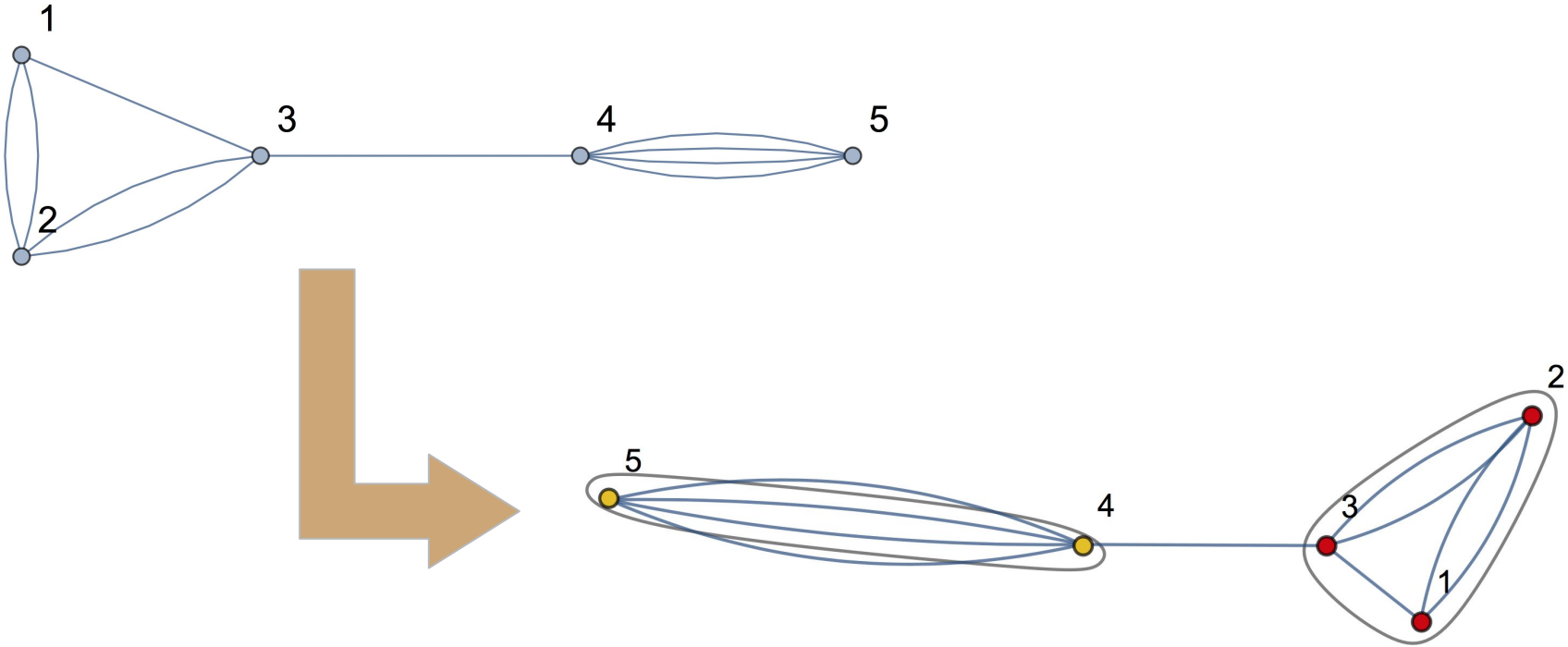
$$L = D^{-1/2} A D^{-1/2}$$

L : 'normalized' Laplacian of the network

Simple Example



Simple Example



Statistical Inference

Statistical Inference

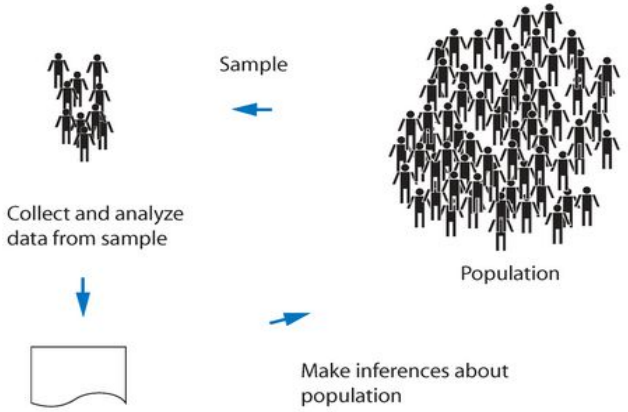
- Statistical inference is the use of probability theory to make inferences about a population from sampled data.

e.g.

- Measure the heights of a random sample of 100 women aged 25-29 years

- Calculate sample mean is 165cms and sample standard deviation is 5 cms

-Make conclusions about the heights of all women in this population aged 25-29 years



Common Forms of Statistical Proposition

The conclusion of a statistical inference is a statistical proposition.

- A point estimate
- An interval estimate
- A credible interval
- Rejection of a hypothesis
- Clustering or classification of data points into groups

Statistical Inference

-Any statistical inference requires some assumptions.

-A statistical model is a set of assumptions concerning the generation of the observed data.

-Given a hypothesis about a population, for which we wish to draw inferences, statistical inference consists of:

1. Selecting a statistical model of the process that generates the data.
2. Deducing propositions from the model.

Stochastic Block Model (SBM)

-SBM is a random graph model, which tends to produce graphs containing communities and assigns a probability value to each pair i, j (edge) in the network.

- To perform community detection, one can fit the model to observed network data using a maximum likelihood method.

Definition of SBM

The stochastic block model studied by Brian, Karrer and M. E. J. Newman:

- G, A_{ij}
- ω_{rs}: the expected value of the adjacency matrix element Aij for vertices i and j lying in groups r and s, respectively
- The number of edges between each pair of vertices be independently Poisson distributed

Goal: To maximize the Probability (Likelihood) that Graph G is generated by SBM

$$L = \prod_{i<j} \frac{(\omega_{g_i g_j})^{A_{ij}}}{A_{ij}!} \exp(-\omega_{g_i g_j})$$

g_i, g_j is the group assignment of vertex i, vertex j

Drawback of SBM

- While formally elegant, SBM works poorly in practice.

- SBM generates networks whose vertices have a Poisson degree distribution, unlike the degree distributions of most real-life networks.

- The model is not a good fit to observed networks for any values of its parameters.

Degree-Corrected Block Model (DCBM)

- DCBM incorporates additional parameters.

- Let the expected value of the adjacency matrix element A_{ij} be $k_i k_j \omega_{g_i g_j}$.

- The likelihood that this network was generated by the degree-corrected stochastic block model:

$$L = \prod_{i<j} \frac{(k_i k_j \omega_{g_i g_j})^{A_{ij}}}{A_{ij}!} \exp(-k_i k_j \omega_{g_i g_j})$$

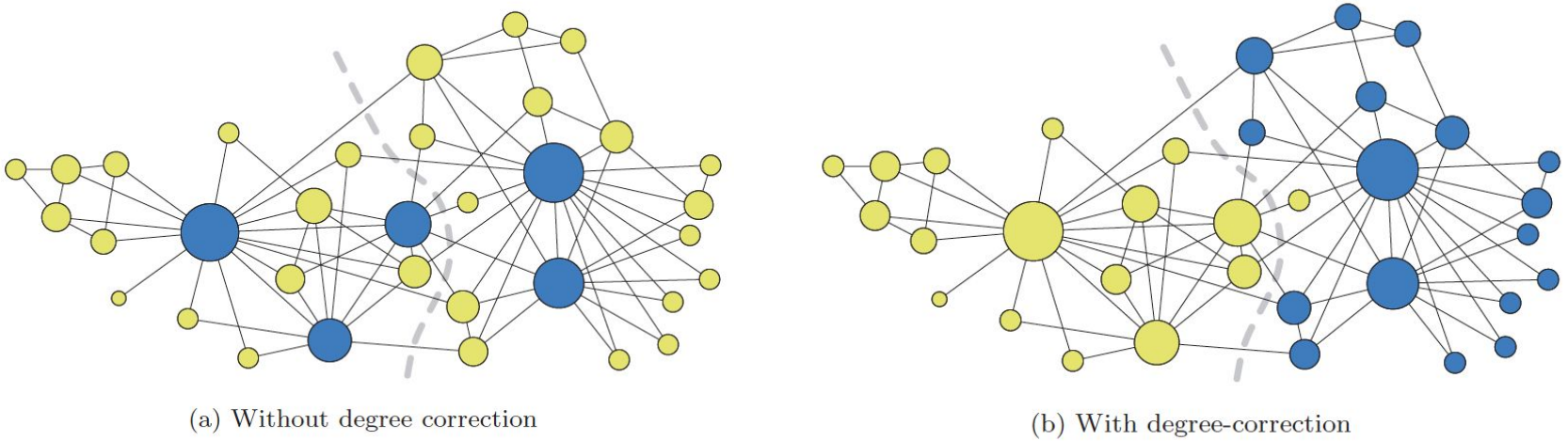
- The desired degrees k_i are equal to the actual degrees of the vertices in the observed network.

- The likelihood depends on the assignment of the vertices to the groups.

Advantage of DCBM

- DCBM improves the fit to real-world data to the point.

- DCBM appears to give good community inference in practical situations.



(a) Without degree correction

(b) With degree-correction

Divisions of the karate club network found using the (a) uncorrected and (b) corrected block models

Optimization Problem

- In maximum likelihood approach, best assignment of vertices to groups is the one that maximizes the likelihood.
- maximize the logarithm of the likelihood:

$$\mathscr{L} = \frac{1}{2}\sum_{ij}\left[A_{ij}\ln\omega_{g_i g_j} - k_i k_j \omega_{g_i g_j}\right]$$

- Assume ω_{in} /ω_{out} for pairs of vertices fall in the same group/ different groups:

$$\omega_{g_i g_j} = \tfrac{1}{2}\left[(\omega_{\text{in}} + \omega_{\text{out}}) + s_i s_j(\omega_{\text{in}} - \omega_{\text{out}})\right]$$

$$\ln\omega_{g_i g_j} = \tfrac{1}{2}\left[\ln(\omega_{\text{in}}\omega_{\text{out}}) + s_i s_j \ln\frac{\omega_{\text{in}}}{\omega_{\text{out}}}\right]$$

- Substitute these expressions into the likelihood:

$$\mathscr{L} = \sum_{ij}\left(A_{ij} - \nu k_i k_j\right)s_i s_j \qquad \nu = \frac{\omega_{\text{in}} - \omega_{\text{out}}}{\ln\omega_{\text{in}} - \ln\omega_{\text{out}}}$$

Using Spectral Method

- Introduce a Lagrange multiplier λ and differentiate:

$$\sum_j (A_{ij} - \nu k_i k_j) s_j = \lambda k_i s_i$$

- In matrix notation:

$$(\mathbf{A} - \nu \mathbf{k}\mathbf{k}^{\mathbf{T}})\mathbf{s} = \lambda \mathbf{D}\mathbf{s}$$

- Multiplying on the left by $\mathbf{1}^T$ and making use of $\mathbf{A1} = \mathbf{D1} = \mathbf{k}$ and $\mathbf{k}^T\mathbf{1} = 2m$

$$\mathbf{k}^T\mathbf{s} - 2m\nu\mathbf{k}^T\mathbf{s} = \lambda\mathbf{k}^T\mathbf{s} \qquad \mathbf{k}^T\mathbf{s} = 0$$

- Simplifies to: $\mathbf{As} = \lambda\mathbf{Ds}$

Normalized-cut Graph Partitioning

What is Graph Partitioning?

Graph partitioning is the problem of dividing a network into a given number of parts (denoted with p) of given sizes such that the cut size R, the number of edges running between parts is minimized.

p = number of parts to be partitioned into (we will focus on p=2 here)

R = number of edges running between parts

Graph Partitioning Tolerance

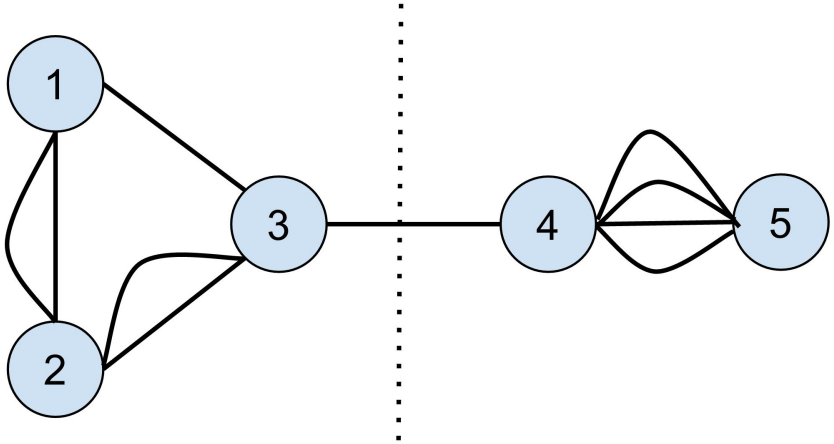
- In the most commonly studied case the parts are taken to be of ***equal size***.
- However, in many situations one is willing to tolerate a little inequality of sizes if it allows for a better cut.

Variants of Graph Partitioning - Ratio Cut

Ratio Cut:

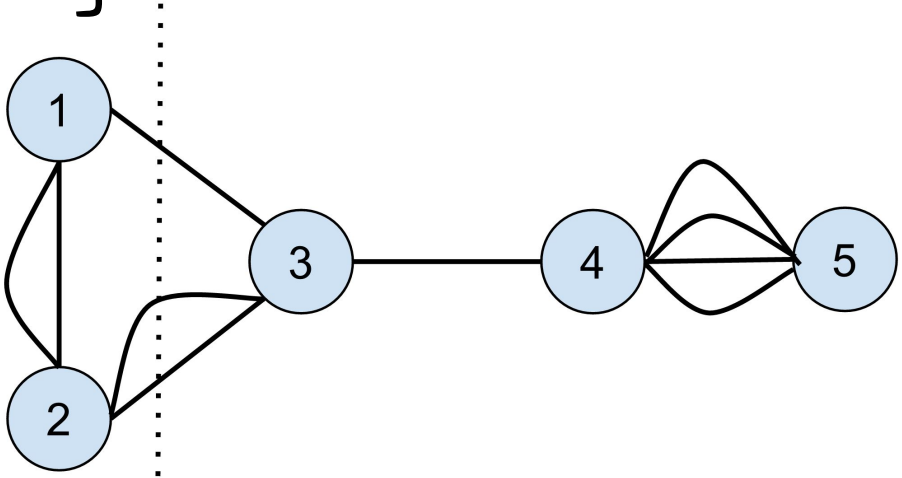
- Minimization objective: R/n_1n_2
- n_1 and n_2 are the sizes (#of vertices) of the two groups
- no more constraint on strictly equal n_i , but n_1n_2 is maximized when $n_1=n_2$, i.e. group partitions with unequal n_i are penalized
- favors divisions of the network where the groups contain equal number of vertices

Variants of Graph Partitioning - Ratio Cut



R=1
n_1=3
n_2=2

R/n_1n_2=1/6



R=3
n_1=2
n_2=3

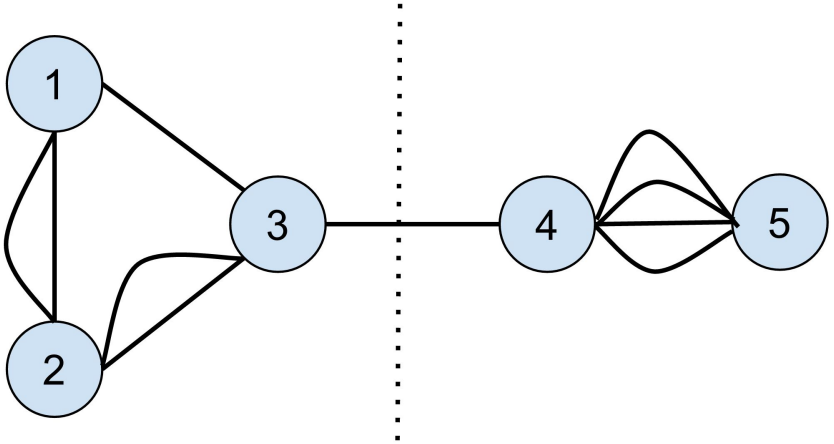
R/n_1n_2=3/6

Variants of Graph Partitioning - Normalized Cut

Normalized Cut:

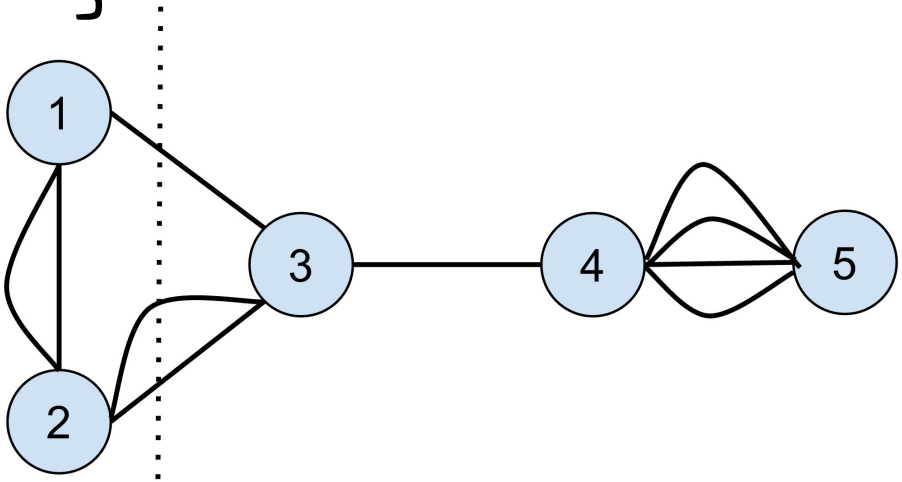
- Minimization objective: R/k_1k_2
- k_1 and k_2 are the sums of the degrees of the vertices in the two groups
  - Sum of degrees = 2x (#of edges)
- no more constraint on strictly equal k_i but k_1k_2 is maximized when $k_1=k_2$, i.e. group partitions with unequal k_i are penalized
- favors divisions of the network where the groups contain equal number of edges

Variants of Graph Partitioning - Normalized Cut



R=1
k_1=10
k_2=8

R/k_1k_2=1/80



R=3
k_1=4
k_2=10

R/k_1k_2=3/40

Using Spectral Method

Similar to the previous 2 derivations, we can use s_i to denote the group membership of each vertex, but rather than ±1, we define:

$$s_i = \begin{cases} \sqrt{\kappa_2/\kappa_1} & \text{if } i \text{ is in group 1} \\ -\sqrt{\kappa_1/\kappa_2} & \text{if } i \text{ is in group 2} \end{cases}$$

Again, use k to denote the vector with elements k_i,
use D to denote the diagonal matrix with $D_{ii}=k_i$:

$$\mathbf{k}^T\mathbf{s} = \sum_i k_i s_i = \sqrt{\frac{\kappa_2}{\kappa_1}} \sum_{i\in 1} k_i - \sqrt{\frac{\kappa_1}{\kappa_2}} \sum_{i\in 2} k_i$$
$$= \sqrt{\kappa_2\kappa_1} - \sqrt{\kappa_1\kappa_2} = 0 \qquad (1)$$

and

$$\mathbf{s}^T\mathbf{D}\mathbf{s} = \sum_i k_i s_i^2 = \frac{\kappa_2}{\kappa_1} \sum_{i\in 1} k_i + \frac{\kappa_1}{\kappa_2} \sum_{i\in 2} k_i = \kappa_2 + \kappa_1$$
$$= 2m \qquad (2)$$

Also:

$$s_i + \sqrt{\frac{\kappa_1}{\kappa_2}} = \frac{2m}{\sqrt{\kappa_1\kappa_2}}\,\delta_{g_i} \quad \text{If } i\in 1$$

$$s_i - \sqrt{\frac{\kappa_2}{\kappa_1}} = -\frac{2m}{\sqrt{\kappa_1\kappa_2}}\,\delta_{g_i} \quad \text{If } i\in 2 \qquad (3)$$

Then:

$$\sum_{ij} A_{ij}\left(s_i+\sqrt{\frac{\kappa_1}{\kappa_2}}\right)\left(s_j-\sqrt{\frac{\kappa_2}{\kappa_1}}\right)$$

$$=-\frac{(2m)^2}{\kappa_1\kappa_2}\sum_{ij}A_{ij}\,\delta_{g_i,1}\delta_{g_j,2}=-\frac{(2m)^2}{\kappa_1\kappa_2}R$$

Combining (1)(2)(3) (4)

$$\left(\mathbf{s}+\sqrt{\frac{\kappa_1}{\kappa_2}}\mathbf{1}\right)^T\mathbf{A}\left(\mathbf{s}-\sqrt{\frac{\kappa_2}{\kappa_1}}\mathbf{1}\right)=\mathbf{s}^T\mathbf{A}\mathbf{s}-2m$$

Use k=A1, 1^TA1=2m (5)

$$\frac{R}{\kappa_1\kappa_2}=\frac{2m-\mathbf{s}^T\mathbf{A}\mathbf{s}}{(2m)^2}$$

Combining (4)(5) (6)

Minimizing $\frac{R}{\kappa_1 \kappa_2} = \frac{2m - \mathbf{s}^T \mathbf{A}\mathbf{s}}{(2m)^2}$ Equivalent → Maximizing $\mathbf{A}\mathbf{s}$

$$\mathbf{A}\mathbf{s} = \lambda \mathbf{D}\mathbf{s} + \mu \mathbf{k}$$ Introducing Lagrange multipliers λ, μ (7)

$$\mathbf{k}^T \mathbf{s} = \lambda \mathbf{k}^T \mathbf{s} + 2m\mu$$ Use $1^T A = 1^T D = k^T$ (8)

$$\mathbf{A}\mathbf{s} = \lambda \mathbf{D}\mathbf{s}$$ Use $\mu = 0$ from (1) (9)

→ Same as the previous 2 problems!

Normalized Cut - Reverse Relaxation

Recall:

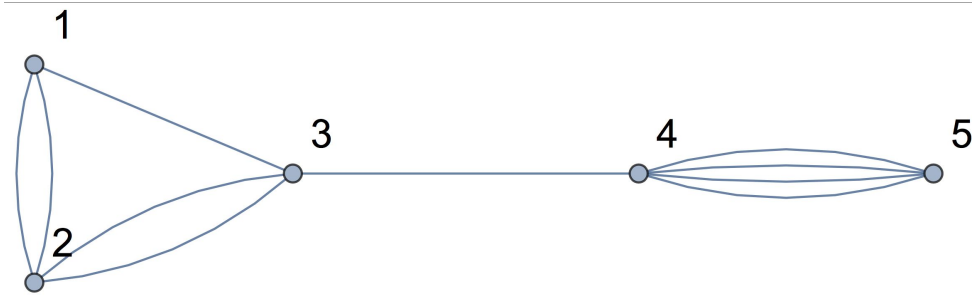
$$s_i = \begin{cases} \sqrt{\kappa_2/\kappa_1} & \text{if } i \text{ is in group 1} \\ -\sqrt{\kappa_1/\kappa_2} & \text{if } i \text{ is in group 2} \end{cases}$$

S_i is ***NOT*** constant like before

-> optimal cutting point may not necessarily be 0

-> the most correct way is to go through every possible cutting point to find the minimum R/k_1k_2

Normalized Cut - Reverse Relaxation

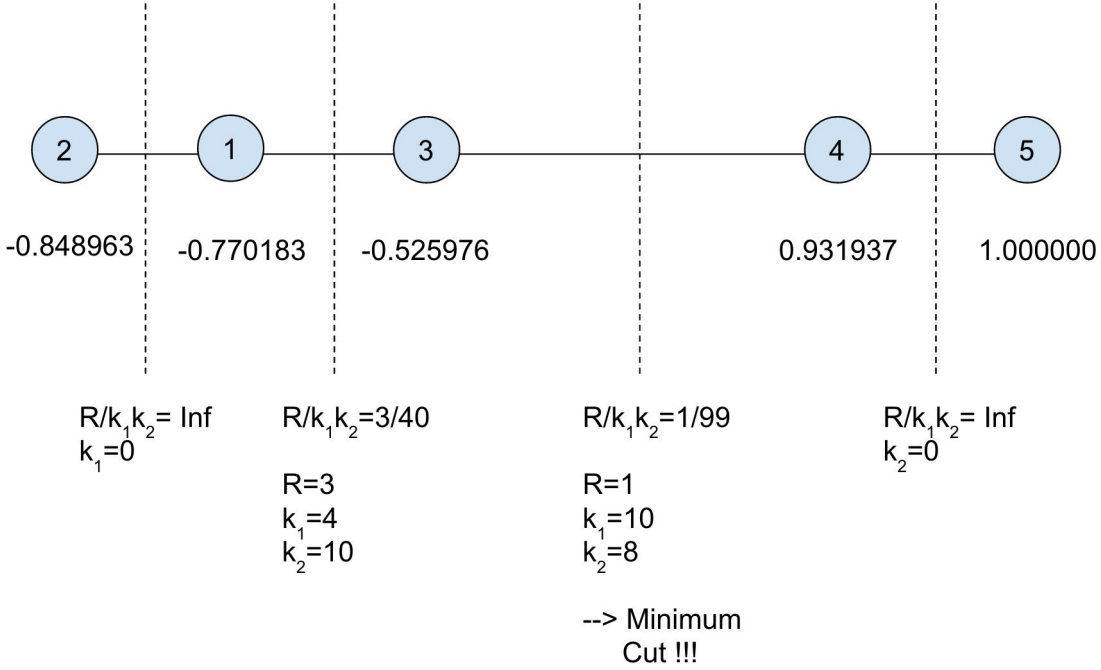


Using the same example, we can get the eigenvector that corresponds to the second largest eigenvalue to be:

{-0.770183, -0.848963, -0.525976, 0.931937, 1.000000}

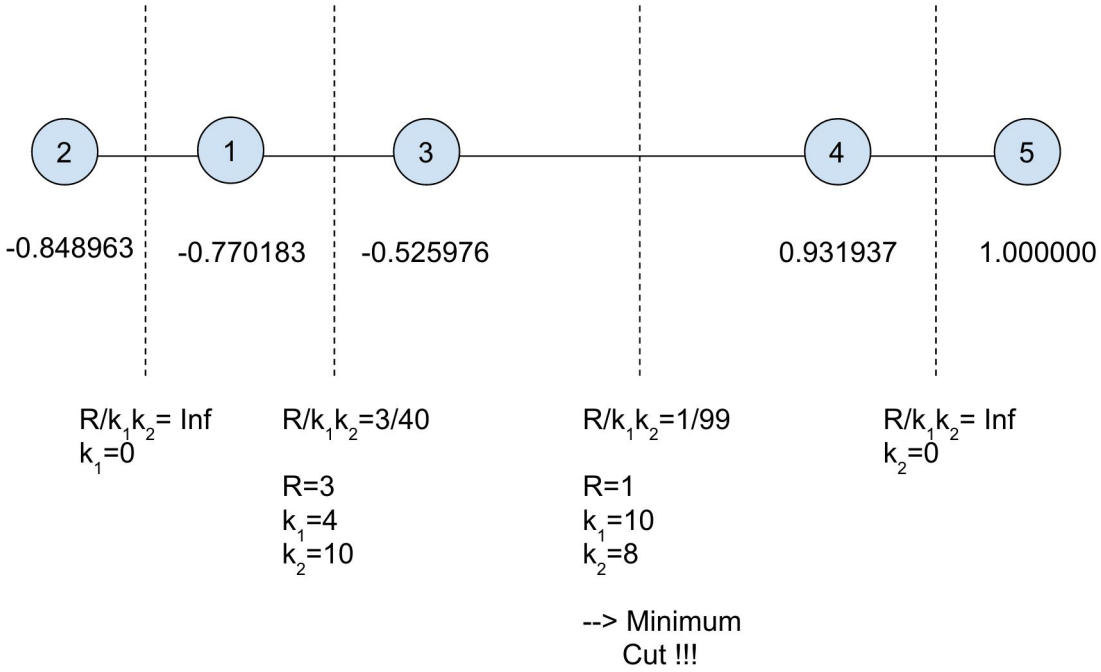
Normalized Cut - Reverse Relaxation

Sort vertices by corresponding value in eigenvector:



Normalized Cut - Reverse Relaxation

Sort vertices by corresponding value in eigenvector:



Note that if we were still to use 0 as the cutting point, it would give us the same result.

In practice:

since $k_1 \approx k_2$, $s_i \approx \pm 1$

Therefore, 0 is still a good cutting point

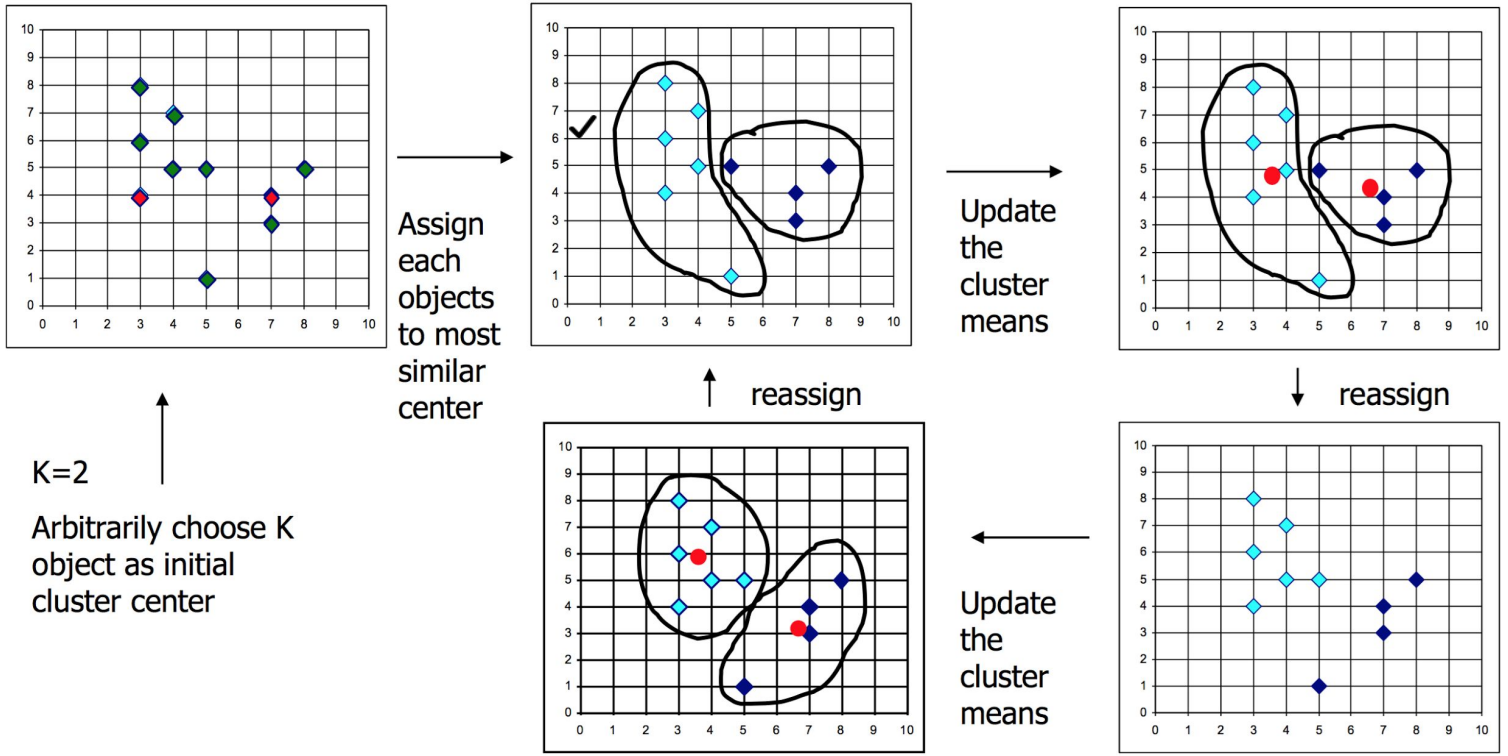
K-means Clustering

Algorithm:

1. Arbitrarily choose k objects as the initial cluster centers
2. Until no change, do:

- (Re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster
- Update the cluster means, i.e., calculate the mean value of the objects for each cluster

K-means Clustering

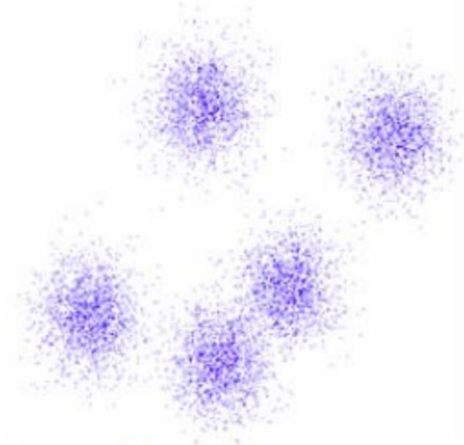


K-means Clustering

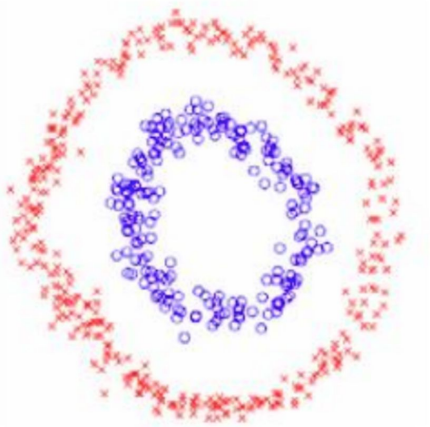
- Relatively efficient: O(tkn)
  - n: # objects, k: # clusters, t: # iterations; k, t << n.
- Often terminate at a local optimum
- Applicable only when mean is defined
- Unable to handle noisy data and outliers
- Unsuitable to discover non-convex clusters

Spectral clustering vs K-means

- **Spectral Clustering: good for connectivity clustering**
- **K-means Clustering: good for compactness clustering**



Compactness

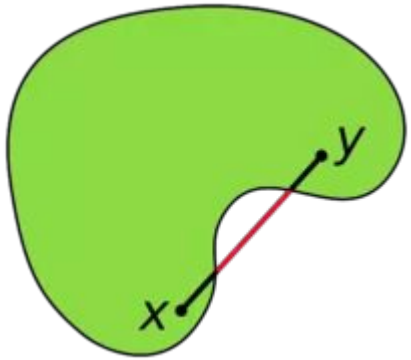


Connectivity

Spectral clustering vs K-means

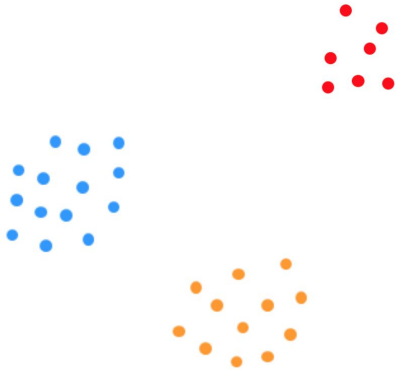
- **Non-convex Sets/Clusters**

  Convex sets: In Euclidean space, an object is convex if for every pair of points within the object, every point on the straight line segment that joins them is also within the object.

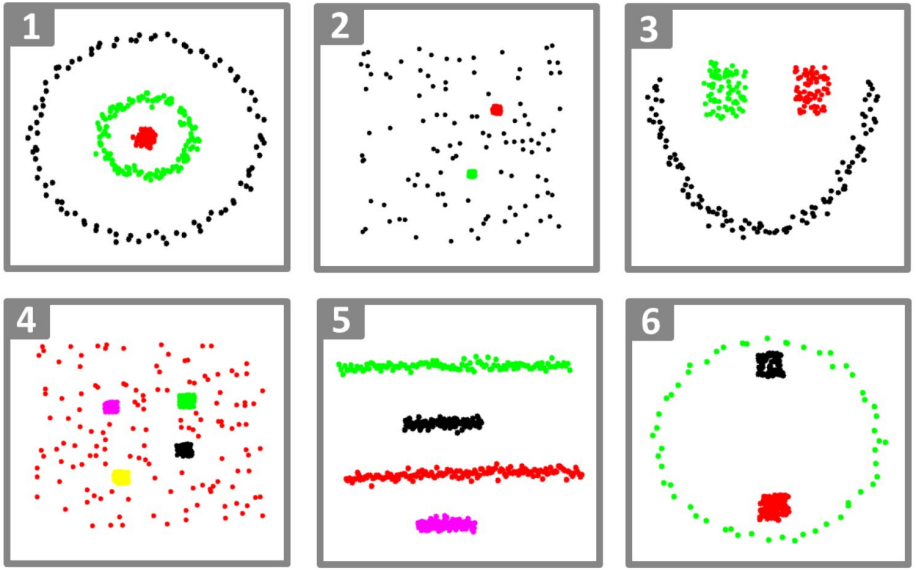


K-means will fail to effectively cluster non-convex data sets:

This is because K-means is only good for clusters where vertices are in close proximity to each other (in the Euclidean sense).

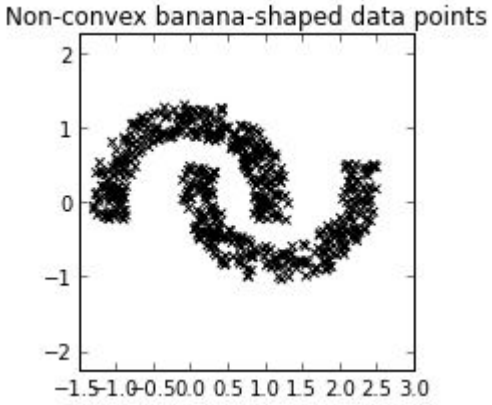


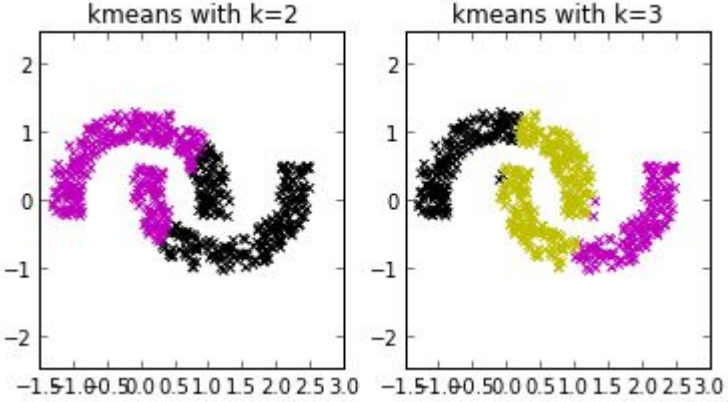
K-means will work



K-means will ***NOT*** work

Using K-means on Non-convex Clusters:

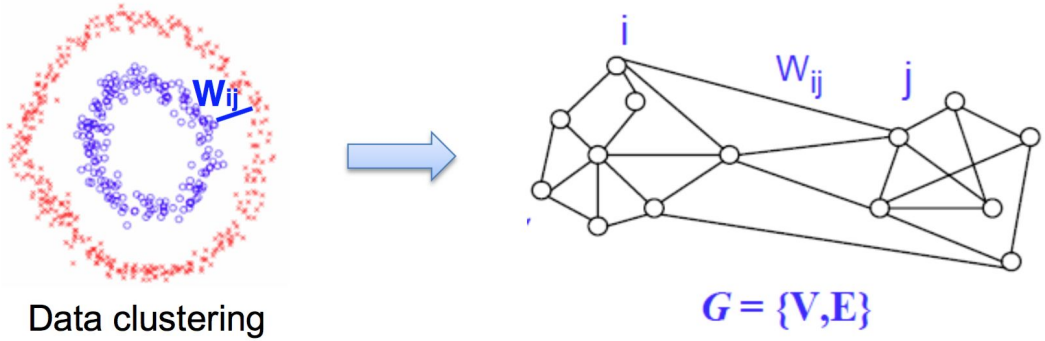




Spectral clustering vs K-means

Data clustering and graph clustering:

We can convert data clustering to graph clustering, where W_{ij} represents the weight of the edge between vertex i and j. W_{ij} is greater when the distance between i and j is shorter.



Spectral clustering vs K-means

Key Advantages:

- **K-means Clustering:**
  - **Relatively efficient: O(tkn) compared to $O(n^3)$ of Spectral Clustering**

- **Spectral Clustering:**
  - **Can handle both convex and non-convex data sets**

Conclusions

- Modularity Maximization, Statistical Inference and Normalized-cut Graph Partitioning are fundamentally/mathematically equivalent.
- Good approximate solutions to these problems can be obtained using spectral clustering method.
- Spectral clustering can effectively detect both convex and non-convex clusters.
- Computational complexity for spectral clustering is $O(n^3)$, which makes it less suitable for very large data sets.

Main References

1. https://www.quora.com/What-are-the-advantages-of-spectral-clustering-over-k-means-clustering
2. https://www.cs.cmu.edu/~aarti/Class/10701/slides/Lecture21_2.pdf
3. https://pafnuty.wordpress.com/2013/08/14/non-convex-sets-with-k-means-and-hierarchical-clustering/
4. Karrer, Brian, and Mark EJ Newman. "Stochastic blockmodels and community structure in networks." *Physical Review E* 83.1 (2011): 016107.
5. https://en.wikipedia.org/wiki/Spectral_clustering
6. Donghui Yan, Ling Huang and Michael I. Jordan. “Fast approximate spectral clustering” 15th ACM Conference on Knowledge Discovery and Data Mining (SIGKDD), Paris, France, 2009. [Long version].
7. Anton, Howard (1987), Elementary Linear Algebra (5th ed.), New York: Wiley, ISBN 0-471-84819-0
8. Wei Wang’s CS145 Lecture Notes

Questions?