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.

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

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

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

\[ \mathbf{A} : \text{ adjacency matrix} \]
\[ k_i : \text{ the degree of vertex } i \]
\[ m : \text{ the total number of edges in the observed network} \]
\[ \delta_{ij} : \text{the Kronecker delta} \]
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.

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

\[ \delta_{g_i g_j} = \frac{1}{2} \left( s_i s_j + 1 \right) \]

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

\( A \) : adjacency matrix  
\( k_i \) : the degree of vertex i  
\( m \) : the total number of edges in the observed network  
\( \delta_{ij} \) : the Kronecker delta
Lagrange Multiplier

maximize $f(x, y)$
subject to $g(x, y) = c$.

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

$\lambda$: eigenvalue
Generalized Eigenvector Equation

A generalized eigenvector of an $n \times n$ 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

\[ \mathbf{A}s = \lambda \mathbf{D}s \]

\( \mathbf{D} \): the diagonal matrix with elements equal to the vertex degrees \( D_{ii} = k_i \)

\( \mathbf{S} \): “Ising spin” variables.

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

\[ \mathbf{L}u = \lambda u \]

\[ L = D^{-1/2} A D^{-1/2} \]

\( \mathbf{L} \): ‘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:

- $G, A_{ij}$
- $\omega_{rs}$: the expected value of the adjacency matrix element $A_{ij}$ 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.

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:
  \[ \mathcal{L} = \frac{1}{2} \sum_{ij} [A_{ij} \ln \omega_{g_i g_j} - k_i k_j \omega_{g_i g_j}] \]
- Assume \( \omega_{in} / \omega_{out} \) for pairs of vertices fall in the same group/ different groups:
  \[ \omega_{g_i g_j} = \frac{1}{2} \left[ (\omega_{in} + \omega_{out}) + s_i s_j (\omega_{in} - \omega_{out}) \right] \]
  \[ \ln \omega_{g_i g_j} = \frac{1}{2} \left[ \ln(\omega_{in} \omega_{out}) + s_i s_j \ln \frac{\omega_{in}}{\omega_{out}} \right] \]
- Substitute these expressions into the likelihood:
  \[ \mathcal{L} = \sum_{ij} (A_{ij} - \nu k_i k_j) s_i s_j \]
  \[ \nu = \frac{\omega_{in} - \omega_{out}}{\ln \omega_{in} - \ln \omega_{out}} \]
Using Spectral Method

- Introduce a Lagrange multiplier $\lambda$ and differentiate:

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

- In matrix notation:

$$(A - \nu kk^T)s = \lambda Ds$$

- Multiplying on the left by $1^T$ and making use of $A1 = D1 = k$ and $k^T1 = 2m$:

$$k^T s - 2\nu k^T s = \lambda k^T s \quad k^T s = 0$$

- Simplifies to: $As = \lambda Ds$
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.

$p = \text{number of parts to be partitioned into (we will focus on } p=2 \text{ here)}$

$R = \text{number of edges running between parts}$
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
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:

$$s_i = \begin{cases} \sqrt{\frac{\kappa_2}{\kappa_1}} & \text{if } i \text{ is in group 1} \\ -\sqrt{\frac{\kappa_1}{\kappa_2}} & \text{if } i \text{ is in group 2} \end{cases}$$
Again, use \( k \) to denote the vector with elements \( k_i \), use \( D \) to denote the diagonal matrix with \( D_{ii} = k_i \):

\[
k^T s = \sum_i k_i s_i = \sqrt{\frac{k_2}{k_1}} \sum_{i \in 1} k_i - \sqrt{\frac{k_1}{k_2}} \sum_{i \in 2} k_i
\]

\[
= \sqrt{\kappa_2 \kappa_1} - \sqrt{\kappa_1 \kappa_2} = 0
\]

(1)

and

\[
s^T D s = \sum_i k_i s_i^2 = \frac{k_2}{k_1} \sum_{i \in 1} k_i + \frac{k_1}{k_2} \sum_{i \in 2} k_i = k_2 + k_1
\]

\[
= 2m
\]

(2)

Also:

\[
s_i + \sqrt{\frac{k_1}{k_2}} = \frac{2m}{\sqrt{\kappa_1 \kappa_2}} \delta g_i \quad \text{if } i \in 1
\]

\[
s_i - \sqrt{\frac{k_2}{k_1}} = -\frac{2m}{\sqrt{\kappa_1 \kappa_2}} \delta g_i \quad \text{if } i \in 2
\]

(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) \hspace{5cm} (4)

\[ \left( s + \sqrt{\frac{\kappa_1}{\kappa_2}} \mathbf{1} \right)^T A \left( s - \sqrt{\frac{\kappa_2}{\kappa_1}} \mathbf{1} \right) = s^T A s - 2m \] Use k=A1, 1^TA1=2m \hspace{5cm} (5)

\[ \frac{R}{\kappa_1 \kappa_2} = \frac{2m - s^T A s}{(2m)^2} \] Combining (4)(5) \hspace{5cm} (6)
Minimizing \[ \frac{R}{\kappa_1 \kappa_2} = \frac{2m - s^T A s}{(2m)^2} \]

Equivalent

Maximizing \( A s \)

\[ A s = \lambda D s + \mu k \] \hspace{1cm} \text{Introducing Lagrange multipliers} \ \lambda, \ \mu \quad (7)

\[ k^T s = \lambda k^T s + 2m \mu \] \hspace{1cm} \text{Use} \ 1^T A = 1^T D = k^T \quad (8)

\[ A s = \lambda D s \] \hspace{1cm} \text{Use} \ \mu = 0 \ \text{from} \ (1) \quad (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 \( R/k_1k_2 \)
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:

Vertices:
- 2: -0.848963
- 1: -0.770183
- 3: -0.525976
- 4: 0.931937
- 5: 1.000000

Equations:
- \( \frac{R}{k_2} = \text{Inf} \) if \( k_1 = 0 \)
- \( \frac{R}{k_1, k_2} = \frac{3}{40} \)
- \( \frac{R}{k_1, k_2} = \frac{1}{99} \)
- \( \frac{R}{k_1, k_2} = \text{Inf} \) if \( k_2 = 0 \)

Values:
- \( R = 3 \)
- \( k_1 = 4 \)
- \( k_2 = 10 \)
- \( R = 1 \)
- \( k_1 = 10 \)
- \( k_2 = 8 \)

Minimum Cut:
- \( k_2 = 8 \)

--> Minimum Cut !!!
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

Assign each object to the most similar center

K=2
Arbitrarily choose K objects as initial cluster centers

Update the cluster means

Reassign

Update the cluster means

Reassign
K-means Clustering

- Relatively efficient: $O(tkn)$
  - $n$: # objects, $k$: # clusters, $t$: # iterations; $k, t \ll n$.
- 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
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

8. Wei Wang's CS145 Lecture Notes
Questions?