Graphs and Community Detection

UNIVERSITY OF SAN FRANCISCO

James D. Wilson

MATH 373

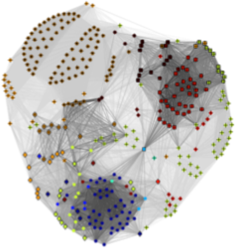
Plan for this Lecture

- Networks
- Exploratory analysis of networks via community detection
  - Spectral clustering
  - Modularity optimization methods
  - Stochastic block modeling methods
  - Extraction-based methods
- Inference on networks via random graph models
- Software and useful resources

Networks

- Means to visualize and model interactions of a complex system
- Treat an “actor" of the system as a vertex and place edges between actors that interact

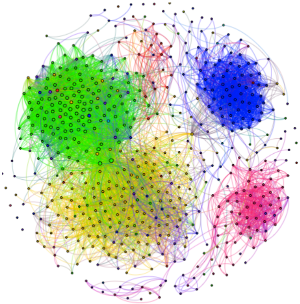
Application: Urovirulence and Antibiotic Resistance



- Interactions of virulence factor genes of *E. coli* from UTI patients

Reference: Parker et al. "Network analysis reveals sex- and antibiotic resistance-associated antivirulence targets in clinical uropathogens" *American Chemical Society: Infectious Diseases* (2015)

Application: Social Networks



- Friendships among my friends on Facebook

Reference: Wilson et al. "A testing based extraction algorithm for identifying significant communities in networks" *Annals of Applied Statistics* (2014)

Statistical Analysis of Networks

Primary goal: Analyze and interpret *relational* data

- Exploratory analysis and visualization
- Simulation and inference of graphical models
- Development of scalable algorithms
- Comparison of statistical algorithms and methodology

Notation

Graph $G = ([n], E)$:

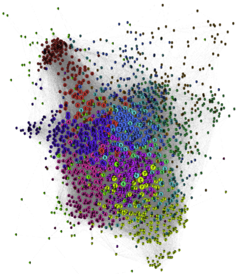
- Vertices ($[n] = \{1, \ldots, n\}$): actors of the system
- Edges ($E \subseteq [n] \times [n]$): placed between actors w/ relationship
- Adjacency Matrix $A \in \mathcal{R}^{n \times n}$:

$$A(u, v) = \text{edge weight between nodes } u, v \in [n]$$

- Degree sequence: $\mathbf{d} = (d(1), \ldots, d(n))$ where

$$\begin{aligned} d(i) &= \#\text{ of edges incident on node } i \\ &= \sum_{v \in [n]} A(u, v) \end{aligned}$$

Community Detection



Informally: Identify communities $C_1, \ldots, C_k \subseteq [n]$ such that

- Edge density within sets C_i is large
- Edge density between sets C_i is small

Aim: Capture relevant structure of a complex system

Community Detection Methods

Min-cut

- Identify cut of vertices that "cuts" the fewest edges

Modularity

- Partition that deviates most from organization in random graph

Spectral

- Focus on spectral properties of graph Laplacian

Stochastic Block Model

- Approximate Maximum Likelihood Estimation

Extraction

- Identify dense communities one at a time

Spectral Clustering and The Graph Laplacian

$G = (V, E)$ undirected graph on $V = \{1, \ldots n\}$ and adjacency matrix S

- Define $D = \text{diag}(d(1), \ldots, d(n)) \in \mathcal{R}^{n \times n}$ where
- Graph Laplacian L:

$$L = D - S$$

- Normalized graph laplacian L_{norm}:

$$L_{norm} = D^{-1} L = I - D^{-1} S$$

Properties of the Graph Laplacian

- λ is an eigenvalue of L_{norm} with eigenvector v iff λ and v solve the eigenproblem $Lv = \lambda Dv$
- 0 is an eigenvalue of L and L_{norm} with eigenvector $\mathbf{1}$
- L and L_{norm} are nonnegative definite and have n real-valued eigenvalues $0 = \lambda_1 \leq \ldots \leq \lambda_n$
- L is symmetric

Key Property of the Graph Laplacian

Theorem 1.

Let G be an undirected graph with non-negative weights and let L_{norm} be its normalized graph laplacian.

Let k = the multiplicity of the eigenvalue 0 of L_{norm}. Then,

(1) k is the number of connected components $C_1, \ldots, C_k$ in G

(2) The eigenspace of 0 is spanned by the indicator vectors $\mathbf{1}_{C_i}$

Remark:

- If G clustered into k disjoint connected components, then we can perfectly identify the k clusters using the k smallest eigenvectors

Spectral Clustering

Algorithm

Input: Adjacency matrix $A \in \mathcal{R}_+^{n \times n}$, number of communities k

1. Calculate normalized graph laplacian L_{norm}
2. Compute

$$X = \text{the } n \times k \text{ matrix of the } k \text{ smallest eigenvectors of } L_{norm}$$

3. Cluster the rows of X using k-means

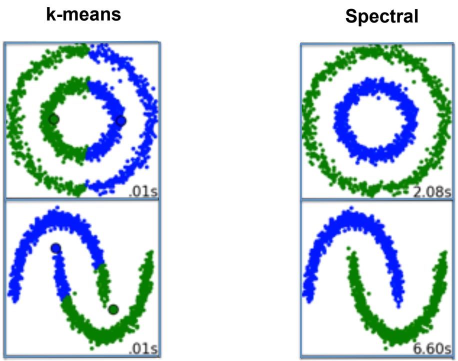
Output: Clusters $C_1, \ldots, C_k$

Properties of Spectral Clustering

- Requires a prespecified number of clusters *k*
- Works perfectly in an ideal scenario
- Requires the use of another clustering method (k-means)
- The solution to a relaxed version of the normalized-cut problem

Reference: Ulrike Von Luxburg "A tutorial on spectral clustering" (2006)

A Quick Comparison



- Spectral can find special structure due to connectivity/similarity
- k-means is faster and better suited for compact clusters

Stochastic Block Model (SBM)

- Model-based approach to community detection
- $G = (V = [n], E)$ with binary adjacency matrix A
- Assumes that G has k blocks generated as follows:

1. Community labels $\mathbf{c} = (c_1, \ldots, c_n)$ generated at random:

$$c_1, \ldots, c_n \overset{iid}{\sim} \text{multinomial}(1, \pi = \{\pi_1, \ldots, \pi_k\})$$

2. Conditional on $\mathbf{c}$, $A(u, v)$ are independent Bernoulli rvs with

$$\mathbb{E}[A(u, v)|\mathbf{c}] = P_{c_u, c_v}$$

Reference: Holland, et al. "Stochastic block models: first steps" (1983)

Stochastic Block Model (SBM)

- Observe $G = G_o$, calculate likelihood $\mathcal{L}(\Theta | G_o, k)$ with $\Theta = \{\mathbf{P}, c\}$
- Finding c becomes an estimation problem:

$$\widehat{\Theta} = \arg\max_{\Theta} \mathcal{L}(\Theta | G_o, k)$$

- Requires approximate algorithms like MCMC or variational EM
- Issue: Approximate algorithms can be slow!

Properties of SBM

- Requires pre-specified k
- “Best" performance on identifying disjoint communities
- Block labels are consistent (as $n \to \infty$) if for all $i \neq \ell$:

$$nP_{i,i} - nP_{i,\ell} \geq \sqrt{k(nP_{i,i} + (k-1)nP_{i,\ell})}$$

- Algorithms like MCMC and variational EM can be slow!

Modularity

- Aim: find the partition of G whose communities contain the highest density of edges relative to the expected density of edges

Remarks:

- Requires a notion of what a *random* network looks like
- The choice of a null network model affects resulting communities

Reference: Mark E Newman "Modularity and community structure in networks" (2004)

Modularity

- Graph $G = (V = [n], E)$, adjacency matrix $A = [A_{u,v}]$
- Modularity ($\mathcal{Q}$): Measures the “significance" of partition **c**:

$$\mathcal{Q}(\mathbf{c}) = \frac{1}{2|E|} \sum_{u,v} \left[\left(A(u,v) - \frac{d(u)d(v)}{2|E|} \right) \mathbb{I}\{c_u = c_v\} \right]$$

- Measures the average departure of observed edge density from expected edge density

Modularity Maximization

Aim: Find the labels $c^* \in \{1, \ldots, k\}^n$ that maximizes modularity:

$$c^* = \arg\max_c \{\mathcal{Q}\}$$

- NP hard optimization problem
- *Many* approximate algorithms developed

Reference: Santo Fortunato, "Community detection in graphs" (2009). [100+ page review paper]

Community Extraction

Basic Idea:

- Identify communities $C_i \subseteq V$ one at a time via iterative search
- Remove/avoid $C_1, \ldots, C_i$ when searching for C_{i+1}

Virtues:

- Possible to accommodate overlap
- Automatic selection of number of communities
- Parallelizable! Can easily scale to large networks.

Community Extraction Methods

Methods:

- OSLOM: Lancichinetti, et al. "Finding statistically significant communities in networks" (2011) – resampling based method
- Extraction: Zhao, et al. "Community extraction for social networks" (2011) – score-based residualizing
- ESSC: Wilson, et al. "A testing based extraction algorithm for identifying significant communities in networks" (2014) – hypothesis testing based extraction

Statistical Inference via Random graph models

Aim: Model the occurrence of an observed graph $G = ([n], E)$ from a family of graphs $\mathcal{G}$.

Use: the distribution on $\mathcal{G}$ gives a means to make inference on complicated network systems.

Reference: Anna Goldenberg et al. "A Survey of Statistical Network Models" (2009)

Applications of Random graph models

- "Small-world brain networks" (2006)
- "Exponential random graphs for social networks" (2012)
- "The structure of adolescent romantic and sexual networks" (2004)
- "Childhood peer network characteristics: genetic influences and links with early mental health trajectories" (2015)
- "Network biology: understanding the cellâs functional organization" (2004)

Statistical Network Models

- Erdős-Rényi model: Independent edges with probability

$$\mathbb{P}(\{u, v\} \in E) = p$$

- Configuration model: Independent edges with probability

$$\mathbb{P}(\{u, v\} \in E) = \frac{d(u)d(v)}{\sum_w d(w)}$$

- Beta model: Let $\beta \in (0, \infty)^n$. Independent edges with probability

$$\text{logit}(\mathbb{P}(\{u, v\} \in E)) = \beta_u + \beta_v$$

Statistical Network Models

- Latent space model: Independent edges with

$$\text{logit}(\mathbb{P}(\{u, v\} \in E)) = X\beta - \delta_{u,v}$$

  - X = design matrix of covariates
  - $\delta_{u,v}$ = latent distance between nodes $u, v \in [n]$

- Stochastic block model: Conditional on community labels **c**, edges are independent with probability

$$\mathbb{P}(\{u, v\} \in E) = P_{c_u, c_v}$$

Statistical Network Models

- Exponential Random Graph model (ERGM): let A denote the observed binary adjacency matrix. Then,

$$P(\mathcal{A} = A) \propto \exp\{\theta^T h(A)\}$$

    - $h(A) : \{0, 1\}^{\binom{n}{2}} \rightarrow \mathcal{R}^p$ = network statistics (both endogeneous and exogeneous).

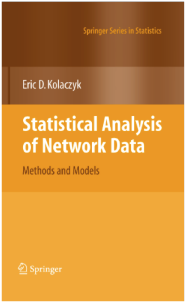
- Generalized Exponential Random Graph model (GERGM): extension of ERGM to general weighted networks.

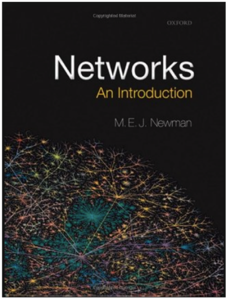
Reference: Wilson et al. "Stochastic weighted graphs: flexible model specification and simulation" (2015)

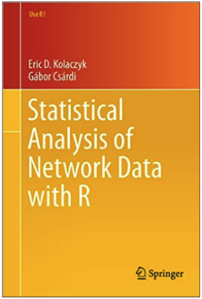
Network Model Challenges

- Appropriately incorporating edge dependencies
- Models that do incorporate dependencies are intractable and rely on approximation / MCMC methods (e.g, ERGM and GERGM)
- Scalability
- Dynamic random graph models

Useful Network Resources







Useful R packages: *igraph* (also available in python); *statnet*; *gergm*