2018

# Discovery of Community Structures in Static and Dynamic Networks

Shiwen Shen
*University of South Carolina*

# Discovery of Community Structures in Static and Dynamic Networks

by

Shiwen Shen

Bachelor of Science
Shanghai University of Finance and Economics 2011

Master of Science
University of Illinois at Urbana-Champaign 2013

---

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Statistics

College of Arts and Sciences

University of South Carolina

2018

Accepted by:

Edsel Peña, Major Professor

Lianming Wang, Committee Member

Yen-Yi Ho, Committee Member

Marco Valtorta, Committee Member

Cheryl L.Addy, Vice Provost and Dean of the Graduate School

# Acknowledgments

First and foremost I want to thank my PhD advisor Dr. Edsel A. Peña. It is my great honor to be his student. I first met him in the class of Advanced Statistical Inference in spring 2014, in which I was fascinated by his passionate and rigorous way of thinking, as well as his enthusiasm in in-depth after-class discussion with students. Of course, I was one of the students who bothered him a lot. In the end of that semester, I started to join the weekly research meeting organized by him, his students, and Dr. James Lynch. It is a place where a wide range of statistical topics are presented on a whiteboard, and criticized by all the members. From which, I have learned not only statistical knowledge, but also the attitude and standard a good researcher should hold, and the skills and confidence required to be a good presenter. Moreover, I appreciate all his contributions of time, ideas, and funding to make my PhD experience rich and pleasant. His sustained enthusiasm in research was contagiously influencing me and motivating me greatly during the difficult moments in pursuing my degree.

Research group members have also contributed to my research remarkably. When I stepped into the group as a newcomer, Dr. AKM Fazlur Rahman, Dr. Bereket Kindo, and Dr. Piaomu Liu were there, who provided me with many useful experience and suggestions. Afterwards, Dr. Beidi Qiang, Taeho Kim, Jeff Thompson, Lu Wang joined, and surprisingly they all have very different research interests, and from whose presentations I tasted distinct flavors of recipes in mixing statistical ideas. Lili Tong and Tahmidul Islam joined the group last year, and Tahmidual happens to be my office mate. I enjoyed every moment when we were challenging each other with hard

statistical and mathematical problems.

For this dissertation I would like to thank my committee members: Dr. Lianming Wang, Dr. Yen-Yi Ho, and Dr. Marco Valtorta for their time, interest, helpful comments, and insightful questions.

I gratefully acknowledge all the professors in the department of statistics, especially those who I have taken classes with, including Dr. Joshua Tebbs, Dr. Tim Hanson, Dr. Iris Lin, Dr. Shan Huang, Dr. John Grego, and Dr. Paramita Chakraborty. Their excellent teaching helped me to set up basic knowledge required to conduct high quality researches in statistics. Besides, I appreciate Dr. Don Edwards for the funding and delightful conversations in the 2016 summer SRCOS meeting, and Dr. Maureen Petkewich for trusting me and bringing me the opportunities to teach diverse undergraduate courses.

I appreciate Center for Colon Cancer Research in USC for the great opportunity to serve as a statistical consultant. I enjoy working with faculties and students, including Dr. Franklin Berger, Dr. Pavel Ortinski, Dr. Carole Oskeritzian, Dr. John Fuseler, and Ahmed Aladhami, to discover biological patterns using statistical methods.

Last but not least, I would like to thank my family for all their love, support, encouragement, and raised me with a love of science and curiosity. I am very fortunate to meet Xinyi Zhao in USC, who became to my girl friend afterwards. Additionally, I am grateful to my roommates Yizheng Wei, Shuai Yuan, and Yifan Li, and all my friends for the time we have spent together in Columbia.

Thank you all very much!

Shiwen Shen

*University of South Carolina*

July 2018

# Abstract

With the development of computer technology, researchers are able to observe and collect enormous amount of data, where the independent and identical distributed assumption is violated. For example, in sociology, individuals in an organization interact with each other to change the underlying social structure; in biology, understanding the gene-gene interaction helps researchers to detect potential diseases; in politics, voters are mutually influenced before the election via private/public speeches and parades, which might ultimately change the election results. It is crucial to study how individuals interact with each other from the data, which would lead to tremendous contributions to the society.

Centuries ago, mathematicians started to describe the interaction of objects with mathematical language in the field of graph theory. The concepts of vertices/nodes and edges are the cornerstone of graph theory. Vertex can be used to describe individual, and edge is a way to portray interaction between a pair of vertices. Taking advantage of the accumulated discoveries in graph theory, statisticians are able to develop stochastic models to make inference of the data, which can be represented by network structures.

My main research goal is to develop statistical models to discover the underlying community structure in various types of network data, including a snap shot of a network and time-varying network. The word "community" is an intermediate concept between a single node and the whole network, and can refer to a partition, a block structure, etc. Additionally, I desire to make my models be feasible to large size data, so that gigantic networks, e.g. social network, can be analyzed using my con-

tributed methodologies. Spectral clustering type of methods, which usually require less computational resources, are proposed to achieve the research goal.

I first explore the methodologies of discovering community structure under an unobserved latent space by shrinking the latent positions of nodes belonging to the same community. Unlike traditional community detection algorithms, the information of edge covariates are taken into consideration for better estimation. I apply the proposed algorithm on an attorney friendship network to check the correlation between friendship status and office location.

I am also interested in analyzing dynamic network data, where a series of networks are observed. For example, the friendship between the same group of undergraduate students are different in the forth year comparing to the first year. One way to detect communities with dynamic network is to treat network on each time point independently. It is convenient, however, historical information (e.g. the network or community structure in the previous time points), which has potential to improve the estimation accuracy, is ignored. I build an algorithm to borrow the historical information and improve the clustering quality with the help of degree of nodes.

# Table of Contents

Acknowledgments . . . . . iii

Abstract . . . . . v

List of Tables . . . . . ix

List of Figures . . . . . x

Chapter 1 Introduction . . . . . 1

1.1 Erdös-Rényi-Gilbert Model . . . . . 1

1.2 Stochastic Block Model . . . . . 2

1.3 Degree Corrected Block Model . . . . . 3

1.4 Latent Space Model . . . . . 4

Chapter 2 Literature Review and Motivation . . . . . 6

2.1 Review of Community Detection Algorithms . . . . . 6

2.2 Motivation . . . . . 16

2.3 Notations . . . . . 16

Chapter 3 Lasso-type Network Community Detection within Latent Space . . . . . 18

3.1 Introduction and Motivation . . . . . 18

3.2 Setting . . . . . 18

3.3 Our Approach . . . . . 20

3.4 Simulations . . . . . 26

3.5 Data Application: Attorney's Friendship Network . . . . . 29

Chapter 4 Dynamic Network Community Discovery . . . . . 34

4.1 Introduction . . . . . 34

4.2 Model Construction . . . . . 36

4.3 Simulations . . . . . 46

4.4 Data Application: Enron Email Data . . . . . 54

4.5 Discussions . . . . . 56

Chapter 5 Conclusion and Extension . . . . . 61

Bibliography . . . . . 64

# List of Tables

Table 3.1 Spectral Clustering with Normalized Graph Laplacian . . . . . 31

Table 3.2 Spectral Clustering with Age Information . . . . . 32

Table 3.3 Spectral Clustering with Working Year Information . . . . . 32

Table 3.4 Spectral Clustering with Both Age and Working Year Information 32

Table 4.1 Experiment 1(a) (top) and 1(b) (bottom) Mean(Sd) . . . . . 48

# List of Figures


Figure 3.1 Lasso Model Simulation 1 Results . . . . . 28

Figure 3.2 Lasso Model Simulation 2 Results . . . . . 30

Figure 3.3 Attorney Friendship Network and Eigenvalue Plot . . . . . 31

Figure 4.1 Experiment 1(a) (left) and 1(b) (right) Error Rate . . . . . 48

Figure 4.2 Experiment 2(a) (left) and 2(b) (right) Error Rate . . . . . 49

Figure 4.3 Experiment 3(a) (left) and 3(b) (right) Error Rate . . . . . 50

Figure 4.4 Experiment 4(a) (left) and 4(b) (right) Error Rate . . . . . 51

Figure 4.5 Experiment 5(a1,a2) (top) and 5(b1,b2) (bottom) Error Rate . . 53

Figure 4.6 Enron Email Network Eigenvalue Plot . . . . . 55

Figure 4.7 Enron Email Network Estimated Communities . . . . . 56

Figure 4.8 Experiment 1(a) (top) and 5(a) (bottom) with/without More Historical Info. . . . . 58

# Chapter 1

# Introduction

Network analysis has become an ubiquitous topic in statistical researches, especially in the field of social science, computer science, marketing, biology, etc. Survey papers and books [9][1][18] have been published in the last ten years alluding to a fast-growing demand in methodologies for analyzing network data. Community detection is one of the most important and well-discussed sub-branch in the network analysis research field. By observing one snapshot of a network, community detection algorithms aim to cluster nodes in a network into a finite number communities. The word "community" can refer to a partition, a block structure, etc, which is an intermediate concept between a single node and whole network. In the language of statistical learning, network community detection problem is unsupervised. The clustered communities depend on the mechanism of how edges are formed, in which subject knowledge is required to make proper interpretation to the estimated communities.

## 1.1 Erdös-Rényi-Gilbert Model

Detecting communities of nodes, or finding an optimal partition of nodes is considered a difficult task, as well as recognizing the true probabilistic structure of a network. Many models have been proposed in the history. One of which, called Stochastic Network Model, was initially discussed by mathematicians in 1950s. Erdös and Rényi [7] and Gilbert [8] published two similar papers at the same period, whose contribution is now identified as the Erdös-Rényi-Gilbert model. Although written in

different perspective, the Erdös-Rényi-Gilbert model can be described in the following unified framework: the probability of generating an edge between a pair of nodes in an undirected network with $n$ nodes is $p$, and generating edges for two pairs of nodes are mutually independent events. Obviously, the model can be parameterized by two parameters, $n$ and $p$. When $n$ is fixed and $p$ is increasing, edges are expected to appear more likely. The degree of a node is the number of edges connect to it. In the Erdös-Rényi-Gilbert model, the degree distribution is approximately Poisson, because

$$\begin{aligned}\mathbb{P}\{\text{degree of node}(v) = k\} &= \binom{n-1}{k} p^k (1-p)^{n-k-1} \\ &\to \frac{(np)^k e^{-np}}{k!}, \quad \text{as } n \to \infty \text{ and } np = \text{constant}.\end{aligned}$$

Additionally, the product of $n$ and $p$ is a critical indicator. If $np$ is greater than 1, a unique giant component is almost surely guaranteed in the network. The Erdös-Rényi-Gilbert model is the simplest Stochastic Network Model, which is not very useful for researchers to detect the network community structure, unless the desired network has only one community - the whole network itself.

## 1.2 Stochastic Block Model

In 1980s, Holland, Laskey, and Leinhardt [14] proposed a probabilistic network framework favoring block(community) structure, called Stochastic Block Model (SBM). SBM assumes there exists $K$ communities in an undirected network with $n$ nodes, and the probability of forming an edge between a pair of nodes completely depends on the blocks(communities) two nodes belong to. For example, let's consider a two-community network ($K = 2$) with the following linkage probability matrix:

$$P = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.9 \end{bmatrix}.$$

The probability of having an edge between a pair of nodes both belonging to community 1(2) is 0.8(0.9), and the probability of having an edge between a pair of nodes belonging to different communities is 0.2. $P$ is symmetric because edges are undirected. After SBM was discovered, it became one of the most popular probabilistic model for analyzing network data, especially in community detection. One drawback of SBM is that it assumes all the nodes belonging to the same community are stochastically identical and the degree distribution for all nodes are the same. Obviously, it is not very realistic.

### 1.3 Degree Corrected Block Model

The paper of Degree Corrected Block Model (DCBM), an extension of SBM, was published in 2011 by physicists Karrer and Newman [17], in which degree heterogeneity is taken into consideration in the mechanism of edge expansion. DCBM introduces a degree parameter $\theta_i \in [0, 1]$ to node $v_i$, $i = 1, 2, \ldots, n$. For example, let's consider a two-community network again with the following *community affinity matrix*:

$$B = \begin{bmatrix} 0.9 & 0.3 \\ 0.3 & 0.7 \end{bmatrix},$$

and assume the degree parameter for node $v_i/v_j/v_k$ is $\theta_i = 0.6/\theta_j = 0.8/\theta_k = 0.1$, $i \neq j \neq k$. The probability of forming an edge between node $v_i$ and $v_j$ is $0.6 \times 0.8 \times 0.7 = 0.336$ if both of them belong to community 2, and the probability of forming an edge between node $v_i$ and $v_k$ is only $0.6 \times 0.1 \times 0.7 = 0.042$ if both of them belong to community 2. Not only the community affinity, but also the degree of individual node impact the connection probability.

Let's put everything into matrix form. Define 0/1-entry *community membership matrix* $Z \in \mathbb{R}^{n \times K}$ with $Z_{ik} = \mathbb{I}\left(v_i \in \mathcal{V}^{(k)}\right)$ indicating whether node $v_i$ belongs to community $K$. Define $\Theta$ to be a diagonal matrix with $\Theta_{ii} = \theta_i$. It is easy to see that

the $(i,j)$th entry of the matrix

$$\Omega = \Theta Z B Z^T \Theta$$

is the probability of forming an edge between node $v_i$ and $v_j$, $i \neq j$, under DCBM. Equivalently, we can write $\Omega_{ij} = \theta_i \theta_j b_{l_i l_j}$, where $l_i$ is the community node $v_i$ belongs to. DCBM became more popular than Stochastic Block Model after it was proposed. However, degree heterogeneity leads to a challenging problem in estimating the partition with extra $n$ unknown and nuisance degree parameters.

## 1.4 Latent Space Model

In addition to block structure models, Hoff, Raftery, and Handcock [13] proposed the latent space model in 2002. The model assumes that there is an unobserved latent space $\mathscr{W} = \mathbb{R}^q, q = 1, 2, \ldots$, such that node $v_i$ can be represented by a mapped position $w_i$ in the latent space $\mathscr{W}$. Define $\mathcal{W} = \{w_1, w_2, \ldots, w_n\}$, $w_i \in \mathscr{W}$, to be the collection of all the latent positions. In the latent space model, small or zero estimated distance between a pair of nodes in the latent space implies that they come from the same community. The model assumes that edges are independently formed conditionally on the latent positions and covariates. The likelihood is

$$\mathbb{P}(A|\mathcal{W}, \mathcal{X}; \theta) = \prod_{i<j} \mathbb{P}(A_{ij}|w_i, w_j, x_{ij}; \theta),$$

where $A \in \mathbb{S}^n$ is called the *adjacency matrix* and its $(i,j)$th entry, $A_{ij}$, measures the edge between node $i$ and $j$. $\mathbb{S}^n$ is a set of $n \times n$ symmetric matrices. $\mathcal{X} = \{x_{ij} \in \mathbb{R}^p | x_{ij} = x_{ji}\}$ is the set of observed covariates of edges, and $\theta = \{\alpha, \beta, \gamma\}$ are the parameters. A natural way to parameterize the likelihood is using the `logit` function:

$$\log \frac{\mathbb{P}(A_{ij} = 1|w_i, w_j, x_{ij}; \theta)}{1 - \mathbb{P}(A_{ij} = 1|w_i, w_j, x_{ij}; \theta)} = \alpha + x_{ij}^T \beta - \gamma \delta_{ij},$$

where $\delta_{ij}$ is a well-defined distance function between nodes $v_i$ and $v_j$ in the latent space, e.g., $\delta_{ij} = \|w_i - w_j\|_q$, $\delta_{ij} = \|w_i - w_j\|_q^2$, etc. The log-odds transformation gives us a simple interpretation: "for two nodes $v_j$ and $v_k$ equi-distant from $v_i$, the log-odds ratio of $v_i \leftrightarrow v_j$ versus $v_i \leftrightarrow v_k$ is $(x_{ij} - x_{jk})^T\beta$." Moreover, the greater the distance is, the lower the chance of connection. The conditional probability of the appearance of an edge is highest when the distance between two nodes is zero. It is worth mentioning that the Stochastic Block Model is one special case of the latent space model [24], where $\beta = 0$ and nodes belonging to the same community are located at the same position in the latent space $\mathscr{W}$.

Existing stochastic network structures, e.g. SBM, DCBM, and latent space model, have established a solid mathematical framework for statisticians to develop algorithms aiming to discover community structure in an observed network for real life, which happens to be my research goal.

In the next chapter, I will briefly review existing algorithms in detecting communities when a snapshot of a network is assumed to be generated by SBM, DCBM, or latent space model. Additionally, we will illustrate the motivation of introducing new models which can be used to discover network communities in both static and dynamic network cases.

# Chapter 2

# Literature Review and Motivation

## 2.1 Review of Community Detection Algorithms

In the previous chapter, we have briefly introduced the ideas of Stochastic Block Model (SBM), Degree Corrected Block Model (DCBM), and latent space model. Here we would like to review some existing algorithms frequently used in discovering communities based on the foregoing models. For example, we would discuss the spectral clustering algorithm [23][28], which is very efficient in detecting communities under the SBM; and for DCBM, a variation of spectral clustering, called Spectral Clustering On Ratios-of-Eigenvectors (SCORE) [16] is briefly demonstrated. In the end, we explain the idea of model-based clustering algorithm under latent space model [12].

### 2.1.1 Spectral Clustering

#### The Idea of Spectral Clustering

The main purpose of spectral clustering is to identify objects, who share similar characteristics. Define $x_i \in \mathbb{R}^p$, $i = 1, 2, \ldots, n$, to be the observed covariates of the $i$th object, and $X = \{x_1, x_2, \ldots, x_n\}$ is the sets containing all $x_i$'s. We are able to construct a *similarity matrix* $W \in \mathbb{S}^n$ by letting $W_{ij} = \exp\{-\|x_i - x_j\|_p^2/2\sigma^2\}$. Note that the exponential formula used here is called *Gaussian similarity function*, and the positive parameter $\sigma$ controls the range of the neighborhoods. Gaussian similarity function is NOT the only way to construct the similarity matrix. Different definitions

can be employed to different problems. Define $D \in \mathbb{S}^n$ to be a diagonal matrix with $D_{ii} = \sum_{j=1}^{n} W_{ij}$. Define the *graph Laplacian* matrix as

$$L = D - W$$
$$L_{\text{sym}} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}},$$

where $L$ is called the *unnormalized graph Laplacian* and $L_{\text{sym}}$ is called the *symmetric graph Laplacian*. Spectral clustering can be conducted on both graph Laplacian matrices; however, detailed algorithms, clustering results, as well as their convergence conditions are different[28].

The traditional spectral clustering algorithm designed to classify $x_i$'s into $K$ clusters with the unnormalized graph Laplacian is

1. Compute the $K$ eigenvectors $\eta_1, \eta_2, \ldots, \eta_K$ of the unnormalized graph Laplacian matrix $L = D - W$ associated with the smallest $K$ eigenvalues.

2. Stack $K$ eigenvectors together to form a $n$ by $K$ matrix $H$, such that the $i$th column of $H$ matrix is $\eta_i, i = 1, 2, \ldots, K$. Equivalently, $H = (\eta_1, \eta_2, \ldots, \eta_K)$.

3. Use k-means algorithm on the rows of $H$ to find the clusters of objects.

The spectral clustering algorithm with the symmetric graph Laplacian was proposed by Ng, Jordan, and Weiss (2002) [23]. Compare to the previous algorithm, the only difference is that it requires unit normalization after step 2 to re-balance the weight of each node. The algorithm is

1. Compute the $K$ eigenvectors $\eta_1, \eta_2, \ldots, \eta_K$ of the symmetric graph Laplacian matrix $L_{\text{sym}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ associated with the smallest $K$ eigenvalues.

2. Stack $K$ eigenvectors together to form a $n$ by $K$ matrix $H$, such that the $i$th column of $H$ matrix is $\eta_i, i = 1, 2, \ldots, K$. Equivalently, $H = (\eta_1, \eta_2, \ldots, \eta_K)$.

3. Normalize the rows of $H$ to generate $H^*$ matrix.

4. Use k-means algorithm on the rows of $H^*$ to find the clusters of objects.

*Remark:* recently, researchers have shown that it is better to compute the $K$ eigenvectors associated with the $K$ smallest **absolute eigenvalues** in the first step, because negative eigenvalues with large absolute value are able to discover "heterophilic" structure in the network [24], which measures the difference between communities.

### Graph Cut

One reason that the spectral clustering is so successful is due to its relationship to the *graph cut*. Define an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ and edge set $\mathcal{E}$. The edge between nodes $v_i$ and $v_j$ is measured by a non-negative weight $A_{ij}$, $i \neq j$. $A_{ii}$ is defined to be 0 indicating no self-loop. Large positive $A_{ij}$ means that the connection between two nodes is strong, and if $A_{ij} = 0$, two nodes are not linked. As we introduced in the last chapter, the matrix $A = [A_{ij}] \in \mathbb{S}^n$ is called the *adjacency matrix*, and in literature, the matrix $A$ is simplified to be a matrix with 0/1 entries measuring the existence, instead of the size, of observed edges. The adjacency matrix $A$ can be understood as a special case of similarity matrix. Define degree matrix $D$, $D_{ii} = d_i = \sum_{j=1}^{n} A_{ij}$, where $d_i$ is the degree of node $v_i$. Define graph Laplacian matrices $L = D - A$ and $L_{sym} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ the same way as we did in the last section. Note that, for notational convenience, we do not define the degree matrix and graph Laplacian matrices of the similarity matrix $W$ and the adjacency matrix $A$ differently. Readers should be able to identify based on the context.

For any subset of nodes $S \subset \mathcal{V}$, we define its complement $\mathcal{V} \backslash S$ to be $\overline{S}$, and use $i \in S$ as a shorthand notation for the set $\{i | v_i \in S\}$. The observed degree of subset $S$, measuring the summation of the degrees of nodes in subset $S$, is defined by $D(S) = \sum_{i \in S} d_i$. Additionally, for any two arbitrary subsets $S$ and $T$, define

$A(S,T) = \sum_{i\in S, j\in T} A_{ij}$ to be the strength of connection between two subsets $S$ and $T$.

Now let's assume that there exists a partition of $\mathcal{V}$ such that $\mathcal{V} = \mathcal{V}^{(1)} \cup \mathcal{V}^{(2)} \cup \cdots \cup \mathcal{V}^{(K)}$, where the cardinarlity of the set $|\mathcal{V}^{(k)}| > 0, k = 1, 2, \ldots, K$, and $\mathcal{V}^{(k)} \cap \mathcal{V}^{(l)} = \emptyset$ for $k \neq l$. Recall that $Z \in \mathbb{R}^{n\times K}$ is a 0/1-entry matrix with $Z_{ik} = \mathbb{I}\left(i \in \mathcal{V}^{(k)}\right)$ is the community membership matrix. The partition and $Z$ are equivalent in representing the community structure of a network. The *mincut* cost function is defined by

$$\text{mincut}(\mathcal{V}^{(1)}, \mathcal{V}^{(2)}, \ldots, \mathcal{V}^{(K)}) = \frac{1}{2}\sum_{k=1}^{K} A(\mathcal{V}^{(k)}, \overline{\mathcal{V}^{(k)}}),$$

which measures the overall strength of connection between each subset $\mathcal{V}^{(k)}$ to its complement. When $K = 2$, minimizing the mincut cost function returns a trivial solution and it splits the whole vertex set into one subset with only one single vertex, and another subset with the rest of the vertices. In order to avoid this problem, researchers [11][25] modified the mincut cost function to maintain a size balanced optimal partition. Define the cost function of ratio cut and normalized cut as follows,

$$\text{RatioCut}(\mathcal{V}^{(1)}, \mathcal{V}^{(2)}, \ldots, \mathcal{V}^{(K)}) = \frac{1}{2}\sum_{k=1}^{K} \frac{A(\mathcal{V}^{(k)}, \overline{\mathcal{V}^{(k)}})}{|\mathcal{V}^{(k)}|},$$
$$\text{NormalizedCut}(\mathcal{V}^{(1)}, \mathcal{V}^{(2)}, \ldots, \mathcal{V}^{(K)}) = \frac{1}{2}\sum_{k=1}^{K} \frac{A(\mathcal{V}^{(k)}, \overline{\mathcal{V}^{(k)}})}{D(\mathcal{V}^{(k)})},$$

where $|\mathcal{V}^{(k)}|$ is the number of vertices in the $k$th subset, and $D(\mathcal{V}^{(k)})$ is the degree of the $k$th subset. When $|\mathcal{V}^{(k)}|$ or $D(\mathcal{V}^{(k)})$ is small, the cost function of ratio cut or normalized cut would tend to be large, which penalizes the partition and further guarantees more size balanced results.

### Relaxed Graph Cut and Spectral Clustering

The insight that "spectral clustering is a convex relaxation of the graph cut optimization problem" [25] motivates us to understand the bridge between these two concepts. The conclusion is that minimizing the ratio cut cost function is approx-

imately equivalent to executing the spectral clustering algorithm with the unnormalized graph Laplacian matrix, and minimizing the normalized cut cost function is approximately equivalent to executing the spectral clustering algorithm with the normalized graph Laplacian matrix. For the rest of this subsection, we show some details of the equivalence of the ratio cut problem and unnormalized spectral clustering problem.

For simplicity, set $K = 2$ and $\mathcal{V} = S \cup \overline{S}$. Denote $u \in \mathbb{R}^n$ to be a vector with $u_i = \sqrt{\frac{|\overline{S}|}{|S|}}$ when $v_i \in S$, and $u_j = -\sqrt{\frac{|S|}{|\overline{S}|}}$ when $v_j \in \overline{S}$. Therefore, the following equations hold:

$$
\begin{aligned}
u^T L u &= u^T D u - u^T A u \\
&= \sum_{i=1}^{n} d_i u_i^2 - \sum_{i,j=1}^{n} u_i u_j A_{ij} \\
&= \frac{1}{2}\left\{ \sum_{i=1}^{n} d_i u_i^2 - 2\sum_{i,j=1}^{n} u_i u_j A_{ij} + \sum_{j=1}^{n} d_j u_j^2 \right\} \\
&= \frac{1}{2}\left\{ \sum_{i=1}^{n} \left( \sum_{j=1}^{n} A_{ij} \right) u_i^2 - 2\sum_{i,j=1}^{n} u_i u_j A_{ij} + \sum_{j=1}^{n} \left( \sum_{i=1}^{n} A_{ij} \right) u_j^2 \right\} \\
&= \frac{1}{2}\sum_{i,j=1}^{n} A_{ij} \left( u_i - u_j \right)^2, \qquad (L \text{ is positive semi-definate}) \\
&= \frac{1}{2}\left\{ \sum_{v_i \in S, v_j \in \overline{S}} A_{ij} \left( \sqrt{\frac{|\overline{S}|}{|S|}} + \sqrt{\frac{|S|}{|\overline{S}|}} \right)^2 + \sum_{v_i \in \overline{S}, v_j \in S} A_{ij} \left( -\sqrt{\frac{|S|}{|\overline{S}|}} - \sqrt{\frac{|\overline{S}|}{|S|}} \right)^2 \right\} \\
&= \left( \frac{|\overline{S}|}{|S|} + \frac{|S|}{|\overline{S}|} + 2 \right) A(S, \overline{S}) \\
&= \left( \frac{|S| + |\overline{S}|}{|S|} + \frac{|S| + |\overline{S}|}{|\overline{S}|} \right) A(S, \overline{S}) \\
&= \frac{|\mathcal{V}|}{2}\left\{ \frac{A(S, \overline{S})}{|S|} + \frac{A(\overline{S}, S)}{|\overline{S}|} \right\} \\
&= |\mathcal{V}| \times \text{RatioCut}(S, \overline{S}) \\
&= n \times \text{RatioCut}(S, \overline{S})
\end{aligned}
$$

Minimizing $\text{RatioCut}(S, \overline{S})$ is equivalent to minimizing $\frac{u^T L u}{n}$. By the construction of the vector $u$, it can be easily verified that $u^T u = n$ and $u^T \mathbf{1}_n = 0$, where $\mathbf{1} \in \mathbb{R}^n$ is

a vector with all entries to be 1, and it is also the eigenvector associated with the largest eigenvalue, which is 0. Therefore, the problem becomes to

$$\begin{aligned} &\text{minimize} \quad \frac{u^T L u}{u^T u}, \\ &\text{subject to} \quad u^T \mathbf{1}_n = 0, \quad \text{and vector } u \text{ has the defined discrete structure.} \end{aligned}$$

This is a NP hard problem [28]. In order to have an approximate solution, we relax the condition by removing the specific discrete structure of vector $u$, and the problem becomes to

$$\begin{aligned} &\text{minimize} \quad \frac{u^T L u}{u^T u}, \\ &\text{subject to} \quad u^T \mathbf{1}_n = 0, \end{aligned}$$

which can be solved by the Rayleigh-Ritz theorem [21], and one solution is the eigenvector of $L$ associated with the second smallest eigenvalue. (Note that the smallest eigenvalue of $L$ is 0 with the eigenvector $\mathbf{1}_n$ because the matrix $L$ is positive semi-definite, which can be observed by the 5th equation in the derivation of $u^T L u$.) Given the estimated $u$, we can apply clustering methods, e.g. k-means, to detect clusters. In this simple case ($K = 2$), we only need to find a cutoff value, such that $v_i$ belongs to $S$ or $\overline{S}$ depending on whether the estimated $u_i$ is greater or less than the cutoff value.

Recall that in the spectral clustering, we stack the eigenvectors of $L$ associated with the $K$ smallest eigenvalues to form $H$ matrix, and then apply k-means on rows of $H$ to detect clusters. Considering the eigenvector of $L$ associated with the smallest eigenvalue is $\mathbf{1}_n$, the estimating results would be the same compared to using the eigenvector of $L$ associated with the second smallest eigenvalue only.

Now let's discuss the case when $K > 2$. Recall that we have defined matrix $Z \in \mathbb{R}^{n \times K}$, whose $(i,k)$th entry is $Z_{ik} = \mathbb{I}\left(i \in \mathcal{V}^{(k)}\right)$. Let $Z = (z_1, z_2, \ldots, z_K)$, where $z_k$ is the $k$th column of the matrix $Z$. For any cluster $k$, $z_k$ is the vector with entries

1 in the positions where the corresponding vertices belong to $\mathcal{V}^{(k)}$. $z_k$ contains $|\mathcal{V}^{(k)}|$ number of 1's and $n - |\mathcal{V}^{(k)}|$ number of 0's.

Define $y_k$ by $y_k = \frac{z_k}{\sqrt{|\mathcal{V}^{(k)}|}}$, and denote $Y = (y_1, y_2, \ldots, y_K)$ be a matrix whose $i$th column is $y_i, i = 1, 2, \ldots, K$. It is obvious that $Y^T Y = I_n$, indicating columns of the $Y$ matrix are orthonormal to each other. Using the same derivation procedure, we are able to show that $y_k^T L y_k = (Y^T L Y)_{kk} = \frac{A\left(\mathcal{V}^{(k)}, \overline{\mathcal{V}^{(k)}}\right)}{|\mathcal{V}^{(k)}|}$. Thus, adding $K$ components together we have

$$\begin{aligned}
\text{RatioCut}(\mathcal{V}^{(1)}, \mathcal{V}^{(2)}, \ldots, \mathcal{V}^{(K)}) &= \frac{1}{2} \sum_{k=1}^{K} \frac{A(\mathcal{V}^{(k)}, \overline{\mathcal{V}^{(k)}})}{|\mathcal{V}^{(k)}|} \\
&= \frac{1}{2} \sum_{k=1}^{K} (Y^T L Y)_{kk} \\
&= \frac{1}{2} \text{tr}(Y^T L Y),
\end{aligned}$$

and minimizing the ratio cut cost function is equivalent to the following problem:

$$\begin{aligned}
&\text{minimize} \quad \text{tr}(Y^T L Y), \\
&\text{subject to} \quad Y^T Y = I_n, \quad \text{and matrix } Y \text{ has the defined discrete structure.}
\end{aligned}$$

Relax the problem by deleting the discrete structure condition, the problem becomes to

$$\begin{aligned}
&\text{minimize} \quad \text{tr}(Y^T L Y), \\
&\text{subject to} \quad Y^T Y = I_n,
\end{aligned}$$

whose solution, by one of the variation of the Rayleigh-Ritz theorem [21](P. 68), is the stack of $K$ orthonormal eigenvectors of $L$ associated with the $K$ smallest eigenvalues.

### Spectral Clustering in Community Detection

The spectral clustering algorithm has been shown to be consistent under the SBM in terms of $n$, not only under the assumption of fixed number of communities, but also in the situation where the number of communities grow as the number of nodes

increases, which is more practical and realistic for social networks [24][20]. However, because of the existence of degree heterogeneity, the consistency of the spectral clustering algorithm cannot be guaranteed under the DCBM model. A relatively new approach, called Spectral Clustering On Ratios-of-Eigenvectors (SCORE), whose main discovery is that by using the entry-wise ratios between the eigenvector of adjacency matrix $A$ associated with the largest absolute eigenvalue, as well as each of the other $K-1$ eigenvectors to form a matrix similar to $H$, the effect of degree heterogeneity is largely ancillary in terms of the clustering results [16]. In the next section, we briefly discuss the SCORE algorithm.

### 2.1.2 SCORE

Under the DCBM model, the SCORE algorithm designed to cluster nodes into $K$ communities is:

1. Compute the $K$ leading eigenvectors $\eta_1, \eta_2, \ldots \eta_K$ of the adjacency matrix $A$ associated with the largest $K$ absolute eigenvalues.

2. Compute matrix $R \in \mathbb{R}^{n\times(K-1)}$ such that for $1 \leq i \leq n$ and $1 \leq l \leq K-1$,

$$R(i,l) = \frac{\eta_{l+1}(i)}{\eta_1(i)},$$

   which is the coordinate-wise ratio between the $i$th entry of the first leading eigenvector and the $i$th entry of the $l$th leading eigenvector.

3. Use k-means algorithm on the rows of $R$ to find the clusters of objects.

In the second step, the term "$l$th leading eigenvector" represents the eigenvector associated with the $l$th largest absolute eigenvalue. There are three obvious difference between the SCORE and the traditional spectral clustering algorithm. First, the graph Laplacian of the SCORE algorithm is defined by $L = A$, which can be understood as a variation of unnormalized graph Laplacian. Second, the $K$ leading

eigenvectors are computed associated with the largest $K$ **absolute eigenvalues**. Third, in the second step of the algorithm, the $R$ matrix is calculated by the entry-wise ratios between the first leading eigenvector and each of the other $K-1$ leading eigenvectors. The notation $R$, instead of $H$, is used to emphasize the entries of $R$ are ratios. It is worth noticing that the effect of degree heterogeneity is largely removed by using the entry-wise ratio of the leading eigenvectors in the second step.

Define $l \in \mathbb{R}^n$ to be the true label vector, say $l_i = k$ if and only if $v_i \in \mathcal{V}^{(k)}$, and let $\hat{l} \in \mathbb{R}^n$ to be the estimated label vector from the SCORE algorithm. Therefore, the expected number of mismatched labels is $\sum_{i=1}^{n} P(\hat{l}_i \neq l_i)$. Define

$$S_K = \{\pi : \pi \text{ is a permutation of the set}\{1, 2, \ldots, K\}\},$$

and the Hamming distance between the true labels and estimated labels can be defined by

$$\text{Hamm}_n(\hat{l}, l) = \min_{\pi \in S_K} \sum_{i=1}^{n} P(\hat{l}_i \neq \pi(l)_i),$$

which measures the distance between $\hat{l}$ and $l$ in terms of the closest permutation. Define $\Omega$ to be the probability of generating the adjacency matrix $A$, and $\eta_k$ is the $k$th leading eigenvector of $A$ associated with the $k$th largest absolute eigenvalue. Define $\bar{\eta}_i$ to be the $k$th leading eigenvector of $\Omega$ associated with the $k$th largest absolute eigenvalue. When $\|\bar{\eta}_k - \eta_k\|$ and $\|\Theta^{-1}(\bar{\eta}_k - \eta_k)\|$ are bounded, it can be shown that

$$\text{Hamm}_n(\hat{l}, l) \leq C \log^3(n) \text{err}_n,$$

where $C$ is a positive constant, and $\text{err}_n$ is a an error bound depending on the degree matrix $\Theta$.

### 2.1.3 Model-based Clustering under Latent Space Model

In the latent space model, it is assumed that there is a latent space, such that each node can be located into one position in the latent space. With the conditional

independence assumption, the likelihood is

$$\mathbb{P}(A|\mathcal{W},\mathcal{X};\theta)=\prod_{i<j}\mathbb{P}(A_{ij}|w_i,w_j,x_{ij};\theta),$$

where $\mathcal{W}=\{w_1,w_2,\ldots.w_n\}$, $w_i\in\mathbb{R}^q$, is the set of latent positions, $\mathcal{X}=\{x_{ij}\in\mathbb{R}^p|x_{ij}=x_{ji}\}$ is the set of all observed characteristics, and $\theta=\{\alpha,\beta,\gamma\}$ are the parameters. We can parameterize the probability using `logit` function,

$$\mu_{ij}\equiv\log\frac{\mathbb{P}(A_{ij}=1|w_i,w_j,x_{ij};\theta)}{1-\mathbb{P}(A_{ij}=1|w_i,w_j,x_{ij};\theta)}=\alpha+x_{ij}^T\beta-\gamma\delta_{ij}.$$

In order to classify nodes into clusters, it is assumed that each latent position $w_i$ belongs to a $q$-dimensional mixture of $K$ multivariate normal distribution, e.g. $w_i\sim\sum_{k=1}^K\lambda_k\text{MVN}_q(\mu_k,\sigma_k^2\mathbb{I}_q)$, where $\lambda_k$ is the probability a node belongs to the $k$th cluster, and $\sum_{k=1}^K\lambda_k=1$.

Two different estimation methods were proposed to maximize the likelihood [12]. One method is called two-stage maximum likelihood estimation. In the first stage, the MLE of the latent position parameters, $w_i$'s, are computed without consideration of the clustering. Maximizing

$$\mathbb{P}(A|\mathcal{W},\mathcal{X};\theta)=\prod_{i<j}\frac{e^{\mu_{ij}A_{ij}}}{1+e^{\mu_{ij}}},$$

who is convex as a function of distance $\delta_{ij}$ providing with a unique solution. Multidimensional scaling technique can be used to recover latent positions after distances are estimated. In the second stage, MLE of the parameters of the multivariate normal distributions can be computed using EM algorithm conditionally on the previously estimated latent positions. This method is comparatively fast; however, information is lost when the latent positions and distribution parameters are not estimated simultaneously.

The second one is a Bayesian approach equipped with Markov chain Monte Carlo (MCMC) sampling techniques. The prior distribution of the parameters are

$$\beta\sim\text{MVN}_p(\xi,\Psi),$$

$$
\begin{aligned}
&\lambda \sim \text{Dirichlet}(\nu),\\
&\sigma_k^2 \overset{\text{IID}}{\sim} \sigma_0^2 \text{Inv}\chi_\gamma^2 \qquad k = 1, 2, \ldots, K,\\
&\mu_k \overset{\text{IID}}{\sim} \text{MVN}_q(0, \omega^2 \mathbb{I}_q) \qquad k = 1, 2, \ldots, K,
\end{aligned}
$$

where $\xi, \Psi, \nu = (\nu_1, \nu_2, \ldots, \nu_K), \sigma_0^2, \gamma$ and $\omega$ are hyper-parameters which are needed to be specified at the beginning. Detailed sampling methods and conditional posterior distributions can be found in [12], and this method leads to better clustering results comparing to the two-stage MLE; however, the computation is very time consuming with large networks.

## 2.2 Motivation

The main research goal is to build models which can be used to discover network community structure with the adjacency matrices, as well as edge covariates. From the model-based clustering, the Bayesian approach is very slow. Consider that there exists million/billion-user social networks, new approaches are needed to resolve the problem to accommodate the large scale networks. At the same time, with advances in computer technology, it is possible to observe the entire evolution of a network over time. Instead of estimating communities with a snapshot of network, it might be better to estimate in a longitudinal manner, and the key question is how to borrow historical information to improve the estimation for the current network communities. In the next two chapters, we will discuss two potential methods, one is designed to have a fast estimation under the latent space model, and one is designed to handle the dynamic network community discovery under the DCBM setting.

## 2.3 Notations

For any vector $x \in \mathbb{R}^q$, $\|x\|_q = \sqrt{x^T x}$ denotes the Euclidean Norm. For any matrix $W$, $\|W\|_F$ denotes the Frobenius norm with $\|W\|_F^2 = \text{trace}(W^T W) = \text{tr}(W^T W)$. For

any set $\mathcal{V}$, $|\mathcal{V}|$ denotes the set cardinality, e.g. the number of objects in the set $\mathcal{V}$. All eigenvectors mentioned are unit-norm. $\mathbb{I}(\cdot)$ is the indicator function. $\mathbf{1}_n \in \mathbb{R}^n$ is a $n \times 1$ vector with all entries equal to 1. $I_n \in \mathbb{R}^{n \times n}$ is the $n \times n$ identity matrix. $\mathbb{S}^n$ is a set for symmetric $n \times n$ matrices.

# Chapter 3

# Lasso-type Network Community Detection within Latent Space

## 3.1 Introduction and Motivation

Imagine that we have observed a snapshot of network data with the adjacency matrix $A$, as well as covariate(s) for each edge. For example, covariates of an edge from social network could be the age/income/education difference between two users. Traditional spectral clustering algorithm can be used to handle adjacency matrix $A$ only; however, it would be beneficial if we can involve covariate(s) information in the algorithm to possibly boost the estimation results. This is the motivation of this chapter.

## 3.2 Setting

As we stated in the introduction, latent space model [13] assumes there exists an unobserved latent space $\mathscr{W} = \mathbb{R}^q$, such that the linkage probability between any pair of nodes $(v_i, v_j), \forall i \neq j$, can be partially represented by the distance between the mapping of $(v_i, v_j)$ in the latent space. $w_i \in \mathbb{R}^q$ is the mapping (latent position) of node $v_i$.

The probability of edge appearance between a pair $(v_i, v_j)$ depends on the distance between $w_i$ and $w_j$. Closer $w_i$ and $w_j$ stands for stronger connection probability. Additionally, covariates might contribute to the connection probability as well. One

natural way to parameterize the probability is using the `logit` function, say

$$\eta_{ij} \equiv \log \frac{\mathbb{P}(A_{ij}=1|w_i,w_j,x_{ij};\theta)}{1-\mathbb{P}(A_{ij}=1|w_i,w_j,x_{ij};\theta)} = \alpha + x_{ij}^T\beta - \gamma\delta_{ij},$$

where $x_{ij} \in \mathbb{R}^p$ is the covariates of pair $(v_i, v_j)$, $\theta = \{\alpha, \beta, \gamma\} \in \{\mathbb{R}, \mathbb{R}^p, \mathbb{R}\backslash\mathbb{R}^-\}$ is the underlying parameter, and $\delta_{ij} \in \mathbb{R}\backslash\mathbb{R}^-$ is a distance measure between $w_i$ and $w_j$. We assume squared Euclidean distance, $\delta_{ij} = \|w_i - w_j\|_q^2$, for mathematical simplicity. In usual logistic regression, the value of $\mathcal{X} = \{x_{ij} \in \mathbb{R}^p | x_{ij} = x_{ji}\}$ and $\Delta = \{\delta_{ij}\}, i, j = \{1, 2, \ldots, n\}, i \neq j$, are observed and known. Differently, in this problem $\Delta$ is unknown because the latent positions $\mathcal{W} = \{w_1, w_2, \ldots, w_n\}$ are unknown and unobservable.

For any two pairs $(v_i, v_j)$ and $(v_l, v_m)$, $i \neq j$, $l \neq m$, two events $\{A_{ij} = 1|w_i, w_j, x_{ij}\}$ and $\{A_{lm} = 1|w_l, w_m, x_{lm}\}$ are assumed to be independent. Therefore, the likelihood and log-likelihood are

$$\prod_{i<j} \mathbb{P}(A_{ij}=1|w_i,w_j,x_{ij};\theta)^{A_{ij}} \left[1-\mathbb{P}(A_{ij}=1|w_i,w_j,x_{ij};\theta)\right]^{1-A_{ij}} = \prod_{i<j} \frac{e^{\eta_{ij}A_{ij}}}{1+e^{\eta_{ij}}},$$

$$\sum_{i<j} \eta_{ij}A_{ij} - \log(1+e^{\eta_{ij}}) = \sum_{i<j} (\alpha + x_{ij}^T\beta - \gamma\delta_{ij})A_{ij} - \log(1+e^{\alpha+x_{ij}^T\beta-\gamma\delta_{ij}}).$$

Maximizing the log-likelihood would provide us with an estimation of the unknown parameters $\theta$ and unknown latent positions $\mathcal{W}$. It is not easy to find the global optimum because the log-likelihood is not concave as a function of $\mathcal{W}$. Hoff et al. [13] suggested to estimate the latent distances $\Delta$ with constrained positive values satisfying the triangle inequality before estimating $\mathcal{W}$ via multidimensional scaling. Handcock et al. [12] proposed a model-based clustering method by assuming $w_i$ follows a finite mixture of $K$ multivariate normal distributions. In the context of community detection, Handcock's approach is better; however, relative heavy computational intensity makes it not suitable to large networks.

## 3.3 Our Approach

Based on the construction, two close nodes in the latent space have higher connection probability, which further implies that these two nodes might belong to the same community. Therefore, we want to penalize the latent distances between all pairs in maximizing the log-likelihood. The objective function we want to minimize is

$$\begin{aligned} g(\theta, \mathcal{W}) &= -\text{log-likelihood} + \lambda \sum_{i<j} \delta_{ij}, \\ &= \sum_{i<j} \left\{ \log(1 + e^{\alpha + x_{ij}^T \beta - \gamma \delta_{ij}}) - (\alpha + x_{ij}^T \beta - \gamma \delta_{ij}) A_{ij} \right\} + \lambda \sum_{i<j} \delta_{ij}, \\ &= \sum_{i<j} \left\{ \log(1 + e^{\alpha + x_{ij}^T \beta - \gamma \delta_{ij}}) - (\alpha + x_{ij}^T \beta) A_{ij} \right\} + \sum_{i<j} (\lambda + \gamma A_{ij}) \delta_{ij}. \end{aligned} \tag{3.1}$$

Equation 3.1 implies that the latent distance of pair $(v_i, v_j)$ is penalized more when edges are observed, say $A_{ij} = 1$. It is ideal to estimate $(\theta, \mathcal{W})$ using

$$(\hat{\theta}, \hat{\mathcal{W}}) = \arg\min_{\theta, \mathcal{W}} g(\theta, \mathcal{W});$$

however, this is very difficult.

Under the context of profile likelihood, unknown parameters $\theta$ and $\mathcal{W}$ can be partitioned as parameter of interest $\mathcal{W}$ and nuisance parameter $\theta$ because our ultimate goal is to detect network community structure from $\mathcal{W}$. Instead of estimating $\theta$ and $\mathcal{W}$ simultaneously, we can first assume that nuisance parameter $\theta$ is known, then we represent $g(\theta, \mathcal{W}) = g_\theta(\mathcal{W})$, and estimate $\mathcal{W}$ using $\hat{\mathcal{W}} = \arg\min_\theta g(\mathcal{W})$. In reality, $\theta$ is unknown. For each estimate of $\theta$, we can estimate $\mathcal{W}$ using

$$\hat{\mathcal{W}} = \arg\min g_{\hat{\theta}}(\mathcal{W}) = \arg\min g(\hat{\theta}, \mathcal{W}).$$

Unfortunately, estimating $\theta$ along is also a difficult task. Therefore, we use the same idea to estimate $\theta$ using

$$\hat{\theta} = \arg\min g_{\hat{\mathcal{W}}}(\theta) = \arg\min g(\theta, \hat{\mathcal{W}}).$$

Given an initial value of either $\hat{\theta}$ or $\hat{\mathcal{W}}$, let two estimation steps be applied alternatively until convergence. The zig-zag type method, as known as the Gauss-Seidel method, could solve the estimation problem much faster. With the estimate $\widehat{\mathcal{W}}$, apply k-means algorithm on each latent positions would return a partition of nodes with $K$ subsets.

### 3.3.1 Estimate $\mathcal{W}$ with given $\hat{\theta}$

With given $\hat{\theta} = \{\hat{\alpha}, \hat{\beta}, \hat{\gamma}\}$ in the previous iteration step, the optimization problem is $\min_{\mathcal{W}} g_{\hat{\theta}}(\mathcal{W})$, which is equivalent to

$$\min_{\mathcal{W}} \sum_{i<j} \left\{ \log(1 + e^{\hat{\alpha} + x_{ij}^T \hat{\beta} - \hat{\gamma}\delta_{ij}}) - (\hat{\alpha} + x_{ij}^T \hat{\beta}) A_{ij} \right\} + \sum_{i<j} (\lambda + \hat{\gamma} A_{ij}) \delta_{ij}.$$

Conducting first-order Taylor series expansion of the log term at $\delta_{ij} = 0$ gives

$$\log(1 + e^{\hat{\alpha} + x_{ij}^T \hat{\beta} - \hat{\gamma}\delta_{ij}}) = \log(1 + e^{\hat{\alpha} + x_{ij}^T \hat{\beta}}) - \frac{\hat{\gamma} e^{\hat{\alpha} + x_{ij}^T \hat{\beta}}}{1 + e^{\hat{\alpha} + x_{ij}^T \hat{\beta}}} \delta_{ij} + \cdots .$$

Plug the expansion into the objective function yields

$$\begin{aligned}
&\min_{\mathcal{W}} \sum_{i<j} \left\{ \log(1 + e^{\hat{\alpha} + x_{ij}^T \hat{\beta}}) - \frac{\hat{\gamma} e^{\hat{\alpha} + x_{ij}^T \hat{\beta}}}{1 + e^{\hat{\alpha} + x_{ij}^T \hat{\beta}}} \delta_{ij} \right\} - (\hat{\alpha} + x_{ij}^T \hat{\beta}) A_{ij} + (\lambda + \hat{\gamma} A_{ij}) \delta_{ij}, \\
\iff &\min_{\mathcal{W}} \sum_{i<j} \left( \lambda + \hat{\gamma} A_{ij} - \frac{\hat{\gamma} e^{\hat{\alpha} + x_{ij}^T \hat{\beta}}}{1 + e^{\hat{\alpha} + x_{ij}^T \hat{\beta}}} \right) \delta_{ij}, \\
\iff &\min_{\mathcal{W}} \sum_{i<j} \left( \lambda + \hat{\gamma} A_{ij} - \frac{\hat{\gamma} e^{\hat{\alpha} + x_{ij}^T \hat{\beta}}}{1 + e^{\hat{\alpha} + x_{ij}^T \hat{\beta}}} \right) \|w_i - w_j\|_q^2, \\
\iff &\min_{W} \operatorname{tr} \left[ W^T (D^* - A^*) W \right], \qquad (3.2)
\end{aligned}$$

where $W = (w_1, w_2, ..., w_n)^T \in \mathbb{R}^{n \times q}$, $A^* \in S^n$ and

$$A_{ij}^* = \lambda + \hat{\gamma} A_{ij} - \frac{\hat{\gamma} e^{\hat{\alpha} + x_{ij}^T \hat{\beta}}}{1 + e^{\hat{\alpha} + x_{ij}^T \hat{\beta}}} = \lambda + \hat{\gamma} \left( A_{ij} - \frac{e^{\hat{\alpha} + x_{ij}^T \hat{\beta}}}{1 + e^{\hat{\alpha} + x_{ij}^T \hat{\beta}}} \right), \quad i < j, \qquad (3.3)$$

$A_{ii}^* = 0$, and $D^* \in S^n$ is a diagonal matrix with $D_{ii}^* = \sum_{j=1}^n A_{ij}^*$. $D^*$ is the degree matrix of $A^*$. We assume $W^T W = I_q$ for identifiability reason. In equation 3.2, the problem can be solved by one of the variation of the Rayleigh-Ritz theorem [21](p68),

and the solution is the column stack of $q$ orthonormal eigenvectors $\tau_1, \tau_2, \ldots, \tau_q$ corresponding to the smallest eigenvalues $\zeta_1 \leq \zeta_2 \leq \cdots \leq \zeta_q$ of matrix $D^* - A^*$, say

$$\widehat{W} = (\tau_1, \tau_2, \ldots, \tau_q).$$

Define $L^* = D^* - A^*$, and $L^*$ is the unnormalized graph Laplacian of $A^*$. It is worth mentioning that $\text{tr}\left[W^T(D^* - A^*)W\right] = \text{tr}\left(W^T L^* W\right)$ is convex with respect to $W$ if and only if $L^*$ is nonnegative definite. $L^*$ is nonnegative definite when $A^*_{ij} \geq 0$. The condition is not always held unless we pick a $\lambda$ such that

$$\lambda \geq \left| \min_{i<j} \left\{ \hat{\gamma} A_{ij} - \frac{\hat{\gamma} e^{\hat{\alpha} + x_{ij}^T \hat{\beta}}}{1 + e^{\hat{\alpha} + x_{ij}^T \hat{\beta}}} \right\} \right|, \qquad \text{or} \quad \lambda \geq \lambda_{\min} \equiv \max_{i<j} \left\{ \frac{\hat{\gamma} e^{\hat{\alpha} + x_{ij}^T \hat{\beta}}}{1 + e^{\hat{\alpha} + x_{ij}^T \hat{\beta}}} \right\}. \quad (3.4)$$

The second inequality relies on the fact that there always exists at least one pair $(v_i, v_j)$ whose $A_{ij} = 0$, which is equivalent to say the network is not fully connected. $\lambda_{\min}$ is the minimum values of $\lambda$ guaranteeing $A^*_{ij} \geq 0$ and the convexity condition of the optimization problem. We want to check whether we could choose larger value of $\lambda$ to further shrink the distances between pairs of nodes belonging to the same community. The answer is "no need". Based on the theorem 3.1, we can see that eigenvectors of $L^*$ is invariant for any $\lambda$ whose value is greater than the described threshold in equation 3.4.

**Theorem 3.1.** *Define similarity matrix $A \in \mathbb{S}^n$ with $A_{ii} = 0$ and $A_{ij} \geq 0$, $\forall i \neq j$. $D$ is the degree of matrix of $A$, and $L = D - A$ is the unnormalized graph Laplacian of $A$. Assume rank$(L) = n - c$. $0 = \zeta_1 = \cdots = \zeta_c \leq \cdots \leq \zeta_n$ are the ordered eigenvalues of $L$, and $\frac{1}{\sqrt{n}}\mathbf{1}_n = \tau_1 = \cdots = \tau_c, \ldots, \tau_n$ are corresponding orthonormal eigenvectors. Define a new matrix $A^* = aA + b(\mathbf{1}_n\mathbf{1}_n^T - I_n) \in \mathbb{S}^n$, where $a, b > 0$. $D^*$ is the degree of matrix of $A^*$, and $L^* = D^* - A^*$ is the unnormalized graph Laplacian of $A^*$. $\zeta_1^* \leq \zeta_2^* \leq \cdots \leq \zeta_n^*$ are the ordered eigenvalues of $L^*$, and $\tau_1^*, \tau_2^*, \ldots, \tau_n^*$ are corresponding orthonormal eigenvectors. It can be shown that $\zeta_i^* = a\zeta_i + nb$ and $\tau_i^* = \tau_i$, $\forall i > c$.*

*Proof.* It is clear that $D^* = aD + (n-1)bI_n$. Therefore, for $\forall i$

$$\begin{aligned} L^*\tau_i &= (D^* - A^*)\tau_i \\ &= \{[aD + (n-1)bI_n] - [aA + b(\mathbf{1}_n\mathbf{1}_n^T - I_n)]\}\tau_i \\ &= [(aL + nbI_n) - b\mathbf{1}_n\mathbf{1}_n^T]\tau_i \\ &= (a\zeta_i + nb)\tau_i - b\mathbf{1}_n\mathbf{1}_n^T\tau_i. \end{aligned}$$

For $\forall i \in [1, c]$, $\zeta_i = 0$, $\tau_i = \frac{1}{\sqrt{n}}\mathbf{1}_n$ gives $L^*\tau_i = nb\frac{1}{\sqrt{n}}\mathbf{1}_n - b\mathbf{1}_n\mathbf{1}_n^T(\frac{1}{\sqrt{n}}\mathbf{1}_n) = 0$, and for $\forall i \in [c+1, n]$, $L^*\tau_i = (a\zeta_i + nb)\tau_i - b\mathbf{1}_n 0 = (a\zeta_i + nb)\tau_i$. □

Theorem 3.1 implies that eigenvectors of $L^*$ remain invariant for different $\lambda$ under the linear transformation of $A^*$ when the convexity condition is held. $\lambda_{\min}$ is the smallest one satisfies the convexity condition. We can rewrite equation 3.3 as a function of $\lambda$ as

$$\begin{aligned} A_{ij}^* &= \lambda + \hat{\gamma}\left(A_{ij} - \frac{e^{\hat{\alpha}+x_{ij}^T\hat{\beta}}}{1+e^{\hat{\alpha}+x_{ij}^T\hat{\beta}}}\right) \\ &= \left[\lambda_{\min} + \hat{\gamma}\left(A_{ij} - \frac{e^{\hat{\alpha}+x_{ij}^T\hat{\beta}}}{1+e^{\hat{\alpha}+x_{ij}^T\hat{\beta}}}\right)\right] + (\lambda - \lambda_{\min}) \end{aligned}$$

where $\lambda - \lambda_{\min}$ plays the same role as $b$ in the theorem. Therefore, we can choose any $\lambda > \lambda_{\min}$. In order to further simplify the equation, we set

$$\lambda = \hat{\gamma} > \lambda_{\min} = \max_{i<j}\left\{\frac{\hat{\gamma}e^{\hat{\alpha}+x_{ij}^T\hat{\beta}}}{1+e^{\hat{\alpha}+x_{ij}^T\hat{\beta}}}\right\}. \tag{3.5}$$

By doing so, the equation 3.3 becomes

$$A_{ij}^* = \hat{\gamma}\left(1 + A_{ij} - \frac{e^{\hat{\alpha}+x_{ij}^T\hat{\beta}}}{1+e^{\hat{\alpha}+x_{ij}^T\hat{\beta}}}\right).$$

With the results of Theorem 3.1, it is equivalent to define

$$A_{ij}^* = 1 + A_{ij} - \frac{e^{\hat{\alpha}+x_{ij}^T\hat{\beta}}}{1+e^{\hat{\alpha}+x_{ij}^T\hat{\beta}}} = A_{ij} + \frac{1}{1+e^{\hat{\alpha}+x_{ij}^T\hat{\beta}}}.$$

The stated minimization procedure is closely related to unnormalized spectral clustering when $\lambda$ is properly selected as stated in equation 3.4 or equation 3.5, and

$A^*$ is considered as a similarity matrix. Conduct spectral clustering on $A^*$ with unnormalized graph Laplacian is exactly equivalent to the previous analysis steps. The connection to spectral clustering implies that choosing $q = K$ in our algorithm is reasonable.

As we can see, when additional covariates are observed, our algorithm can combine the information of the adjacency matrix $A$ and covariates $\mathcal{X}$ together to form a new similarity matrix $A^*$, which might be able to provide better community estimation. At the meantime time, when there is no observed covariate, equation 3.3 becomes

$$A^*_{ij} = \hat{\gamma} A_{ij} + \left( \lambda - \frac{\hat{\gamma} e^{\hat{\alpha}}}{1 + e^{\hat{\alpha}}} \right),$$

and the clustering results will be identical to the traditional spectral clustering based on the Theorem 3.1.

### 3.3.2 Estimate $\theta$ with given $\widehat{\mathcal{W}}$

With given $\widehat{\mathcal{W}} = \{\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_n\}$ and $q = K$, we can compute $\hat{\Delta} = \{\hat{\delta}_{ij}\} = \{\|\hat{w}_i - \hat{w}_j\|_K^2\}$. The optimization problem is $\min_\theta g_{\widehat{W}}(\theta)$, which is equivalent to

$$\min_{\theta = \{\alpha, \beta, \gamma\}} \sum_{i<j} \log(1 + e^{\alpha + x_{ij}^T \beta - \gamma \hat{\delta}_{ij}}) - (\alpha + x_{ij}^T \beta - \gamma \hat{\delta}_{ij}) A_{ij}.$$

It is the logistic regression problem with $A_{ij}$ as response, and $(x_{ij}, -\delta_{ij})$ as predictors, which can be accomplished by any statistical software (e.g. R) efficiently.

### 3.3.3 The Algorithm

Given the adjacency matrix $A$ and covariates $\mathcal{X}$, our proposed community detection algorithm is

1. Conduct unnormalized spectral clustering on $A$ to have initial estimate of latent positions:

   a) Compute degree matrix of $A$ by letting $D = \text{diag}\left(\sum_{j=1}^n A_{ij}\right)$.

b) Compute the $K$ eigenvectors $\tau_1, \tau_2, \ldots, \tau_K$ corresponding to the $K$ smallest eigenvalues $\zeta_1 \leq \zeta_2 \leq \cdots \leq \zeta_q$ of unnormalized Laplacian matrix $L = D - A$.

c) Let $\widehat{W} \in \mathbb{R}^{n \times K}$ be the matrix containing the vectors $\tau_1, \tau_2, \ldots, \tau_K$ as columns.

d) $\hat{w}_i$ is the $i$th row of $\widehat{W}$, and $\widehat{\mathcal{W}} = \{\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_n\}$.

2. Compute $\hat{\Delta} = \{\hat{\delta}_{ij}\} = \{\|\hat{w}_i - \hat{w}_j\|_K^2\}$.

3. Define $\mathcal{X}^* = \{x_{ij}^*\} = \{(x_{ij}, -\hat{\delta}_{ij})\}$ , and conduct logistic regression with $A_{ij}$ as response and $x_{ij}^*$ as predictor to have $\hat{\theta} = \{\hat{\alpha}, \hat{\beta}, \hat{\gamma}\}$.

4. Define $\hat{A}^*$ with $\hat{A}_{ij}^* = A_{ij} + \frac{1}{1+e^{\hat{\alpha}+x_{ij}^T\hat{\beta}}}$.

5. Define $A = \hat{A}^*$, and repeat step $1-4$ until $\hat{\theta}$ (or $\hat{\Delta}$) converges.

6. Apply k-means algorithm on $\widehat{\mathcal{W}} = \{\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_n\}$ to achieve the partition.

The consistency of spectral clustering has been discussed [29], in which strong evidence has been discovered for the superiority of the spectral clustering with normalized graph Laplacian. More specifically, it converges in more general conditions comparing to using the unnormalized graph Laplacian. Therefore, one reasonable variation of the above algorithm is to conduct spectral clustering algorithm on $A^*$ with symmetric graph Laplacian. The algorithm is

1. Conduct spectral clustering on $A$ with the symmetric graph Laplacian to have initial estimate of latent positions:

   a) Compute degree matrix of $A$ by letting $D = \text{diag}\left(\sum_{j=1}^n A_{ij}\right)$.

   b) Compute the $K$ eigenvectors $\tau_1, \tau_2, \ldots, \tau_K$ corresponding to the $K$ smallest eigenvalues $\zeta_1 \leq \zeta_2 \leq \cdots \leq \zeta_q$ of symmetric Laplacian $L_{\text{sym}} = I_n - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$.

c) Let $\widehat{W} \in \mathbb{R}^{n\times K}$ be the matrix containing the vectors $\tau_1, \tau_2, \ldots, \tau_K$ as columns.

d) Normalized the rows of $\widehat{W}$ to 1 to generate $\widehat{W}^*$ matrix.

e) $\hat{w}_i^*$ is the $i$th row of $\widehat{W}^*$, and $\widehat{\mathcal{W}}^* = \{\hat{w}_1^*, \hat{w}_2^*, \ldots, \hat{w}_n^*\}$.

2. Compute $\hat{\Delta} = \{\hat{\delta}_{ij}\} = \{\|\hat{w}_i^* - \hat{w}_j^*\|_K^2\}$.

3. Define $\mathcal{X}^* = \{x_{ij}^*\} = \{(x_{ij}, -\hat{\delta}_{ij})\}$ , and conduct logistic regression with $A_{ij}$ as response and $x_{ij}^*$ as predictor to have $\hat{\theta} = \{\hat{\alpha}, \hat{\beta}, \hat{\gamma}\}$.

4. Define $\hat{A}^*$ with $\hat{A}_{ij}^* = A_{ij} + \frac{1}{1+e^{\hat{\alpha}+x_{ij}^T\hat{\beta}}}$.

5. Define $A = \hat{A}^*$, and repeat step $1-4$ until $\hat{\theta}$ (or $\hat{\Delta}$) converges.

6. Apply k-means algorithm on $\widehat{\mathcal{W}}^* = \{\hat{w}_1^*, \hat{w}_2^*, \ldots, \hat{w}_n^*\}$ to achieve the partition.

## 3.4 Simulations

We use simulated data to compare our algorithms with the original spectral clustering algorithms.

### 3.4.1 Simulation 1

Let's imagine that there are three villages in a map indicating three communities. In total there are 300 villagers in all three villages, and each villager has one-third chance to be assigned to live in one of the three villages. Villagers are connected randomly and connection probability depends on whether two villagers are in the same village, as well as the age difference. We assume that similar age villagers have higher chance to be connected. Motivated by this story, we assume

$$\mathbb{P}(A_{ij} = 1|...) = \frac{\exp(0.5 + \beta x_{ij} - \delta_{ij})}{1 + \exp(0.5 + \beta x_{ij} - \delta_{ij})},$$

where $\delta_{ij} = 0$ if two villagers(nodes) belong to the same village(community), otherwise $\delta_{ij} = 1$. $x_{ij}$ is the age difference covariate. It is generated in the following way: we randomly generate an "age" variable for each node under Uniform$(10, 70)$, and $x_{ij}$ is the absolute age difference between villager $i$ and $j$, $\forall i \neq j$.

When $\beta = 0$, the connection probability is fully determined by the community status. When the absolute value of $\beta$ increases, the age differences contribute more, so that our algorithm should be able to outperform those without taking age difference into consideration.

We compare traditional spectral clustering algorithm with the unnormalized and the normalized (symmetric) graph Laplacian with our proposed algorithms by the misclassification error

$$\frac{\text{number of misclassified villagers}}{300}$$

for various values of $\beta$. For each setting, we iterate 50 times to calculate sample mean, median, and standard deviation.

Simulation 1 results are summarized in Figure 3.1. Blue(red) dashed/solid line represents spectral clustering with unnormalized/normalized graph Laplacian corresponding to $A(A^*)$. When there is no age effect ($\beta = 0$), all four models perform perfectly. When the absolute value of $\beta$ increases from 0 to 0.05, traditional spectral clustering algorithms, as well as the our approach regarding to the unnormalized graph Laplacian start to fail. At the meantime, the performance of our approach regarding to the normalized graph Laplacian gradually drops, which has been shown as the best approach in this simulation even thought the variation is a little higher comparing to other approaches when the age effect is strong.

### 3.4.2 Simulation 2

In simulation 2, we desire to explore the convergence of our algorithms. We set $\beta = -0.3$, which indicates a fairly strong age effect. The number of total villagers

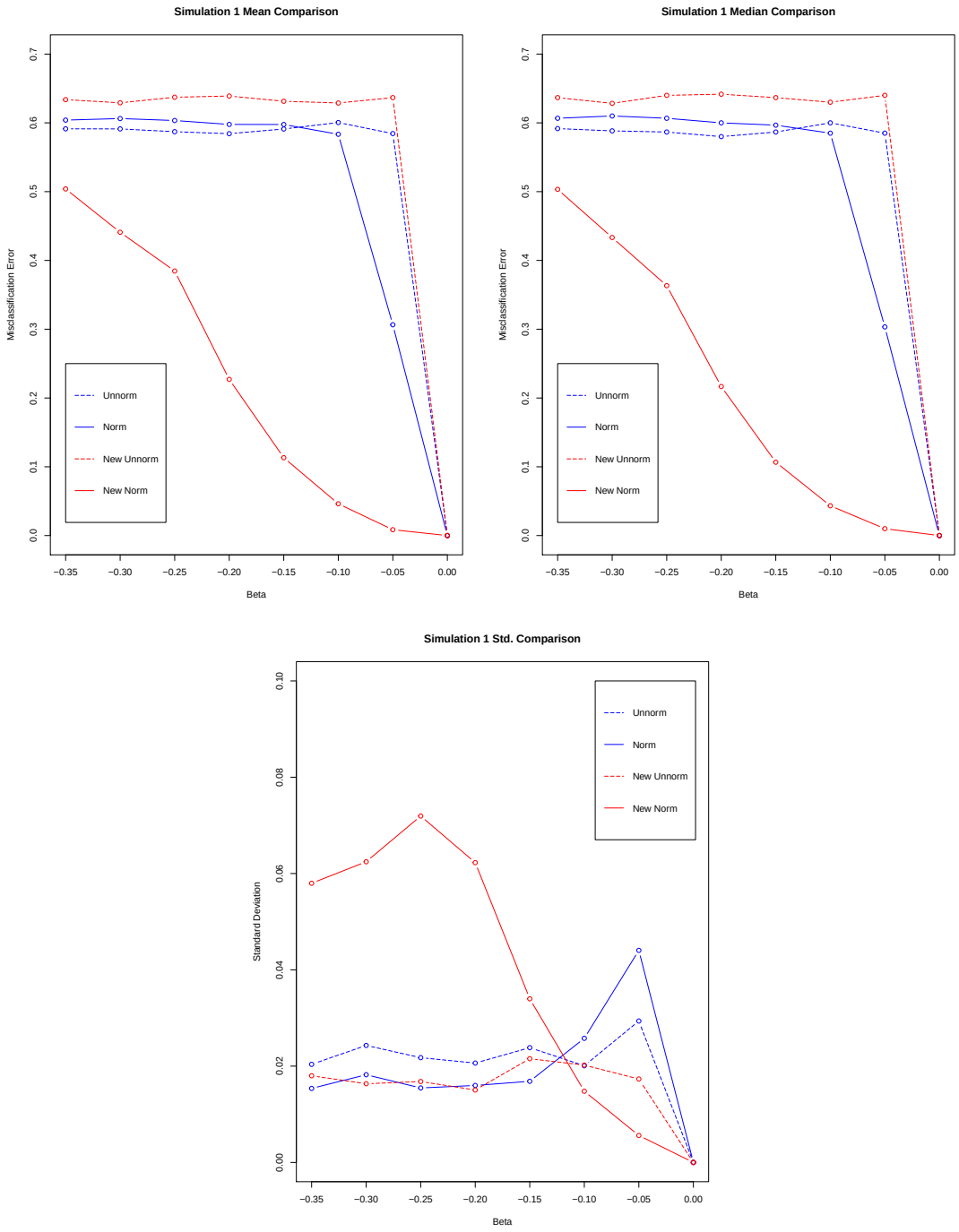


Figure 3.1 Lasso Model Simulation 1 Results

*Blue dashed/solid line represents spectral clustering with unnormalized/normalized graph Laplacian corresponding to $A$. Red dashed/solid line represents spectral clustering with unnormalized/normalized graph Laplacian corresponding to $A^*$.*

are simulated from 300 to 1000. Other conditions remain the same.

Simulation 2 results are summarized in Figure 3.2. The mean and median misclassification error of our approach regarding to the normalized graph Laplacian converges when $n$ increases. Other three algorithms fail completely even with large $n$. The higher variation is negligible when we consider the overall clustering performance.

## 3.5 Data Application: Attorney's Friendship Network

In 1988-1991, friendship data was collected in a Northeastern US corporate law firm in New England [19]. It includes 71 attorneys, and each of them pointed out their friends in the company by answering the following survey question: *"Would you go through this list, and check the names of those you socialize with outside work. You know their family, they know yours, for instance. I do not mean all the people you are simply on a friendly level with, or people you happen to meet at Firm functions."* We are able to portray a social network using the answers from 71 attorneys. Exclude two isolated nodes, and treat all directed edges undirected and unweighted, we are able to plot the network in figure 3.3. The size of node is proportional to the node degree, and color representation illustrates which office attorneys work daily (Boston/yellow, Hartford/green, or Providence/red).

Our goal is to see whether the friendship status has strong correlation with the office location, which can be tested to see if the friendship network clustering result matches the office geographical status. Figure 3.3 also includes the eigenvalue plot of the unnormalized graph Laplacian, in which two big gaps can be seen obviously. The red line separates the first (smallest) eigenvalue and the rest ones implying that the network might contains one community. On the other hand, the blue line separates the first two eigenvalues and the rest ones implying two communities. This method is frequently used to determine the number of communities in unsupervised network community detection problem and clustering problem. From the network plot, we

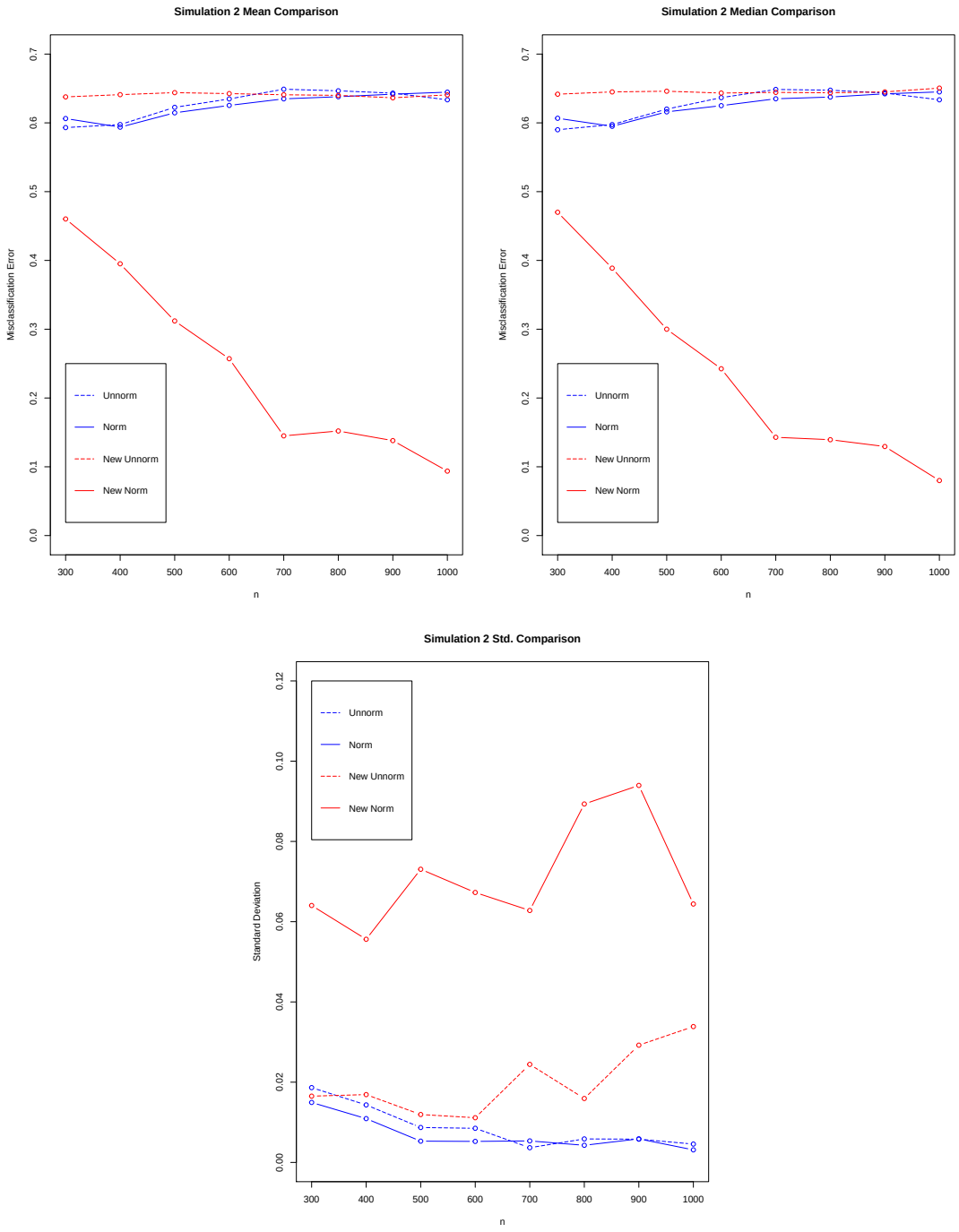


Figure 3.2 Lasso Model Simulation 2 Results

*Blue dashed/solid line represents spectral clustering with unnormalized/normalized graph Laplacian corresponding to $A$. Red dashed/solid line represents spectral clustering with unnormalized/normalized graph Laplacian corresponding to $A^*$.*

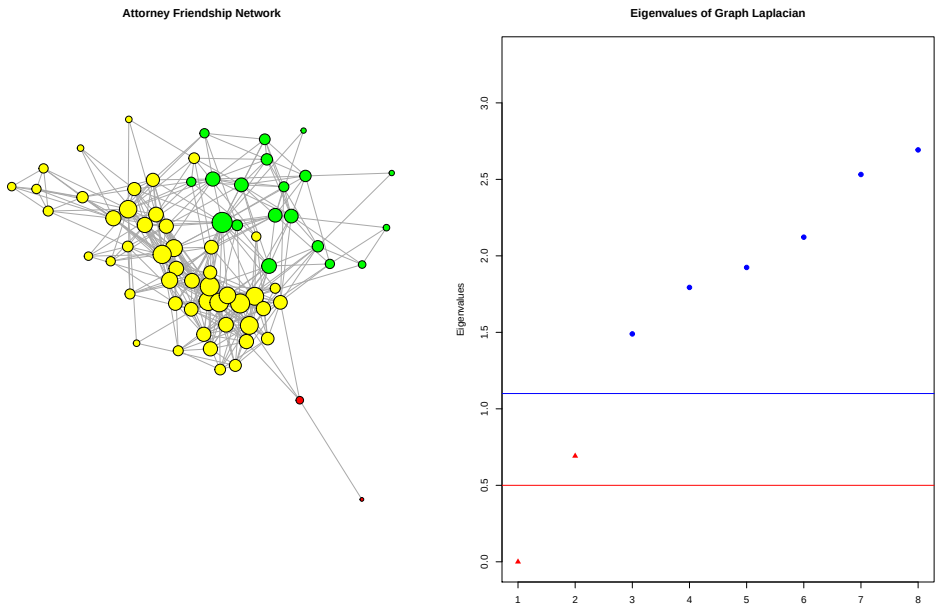


Figure 3.3 Attorney Friendship Network and Eigenvalue Plot

notice that there are only two nodes belonging to the office in Providence (with color red). Therefore, it is fair to assume $K = 2$ for our the analysis.

First, we conduct spectral clustering with normalized(symmetric) graph Laplacian on the adjacency matrix (table 3.1). The misclassification rate is $\frac{1+29+2}{69} = \frac{32}{59}$, indicating the failure of the algorithm.

Table 3.1 Spectral Clustering with Normalized Graph Laplacian

| | Office | | | |
|---|---|---|---|---|
| Estimation | | Boston | Hartford | Providence |
| | Boston | 19 | 1 | 0 |
| | Hartford | 29 | 18 | 2 |

Next, we conduct our algorithm with normalized graph Laplacian on the adjacency matrix, as well as two covariates: age difference and year difference. Age difference is calculated by the absolute difference between two attorneys' ages, and year difference

is calculated by the absolute difference between two attorneys' years with the firm. The misclassification rate depending on adjacency matrix and age difference is $\frac{5}{59}$ (table 3.2), which can be considered as a huge improvement. Borrowing information from working year difference gives misclassification rate $\frac{6}{59}$ (table 3.3), and using both age and year information gives misclassification rate $\frac{7}{69}$ (table 3.4).

Table 3.2 Spectral Clustering with Age Information

| | Office | | | |
|---|---|---|---|---|
| Estimation | | Boston | Hartford | Providence |
| | Boston | 45 | 0 | 0 |
| | Hartford | 3 | 19 | 2 |

Table 3.3 Spectral Clustering with Working Year Information

| | Office | | | |
|---|---|---|---|---|
| Estimation | | Boston | Hartford | Providence |
| | Boston | 45 | 1 | 0 |
| | Hartford | 3 | 18 | 2 |

Table 3.4 Spectral Clustering with Both Age and Working Year Information

| | Office | | | |
|---|---|---|---|---|
| Estimation | | Boston | Hartford | Providence |
| | Boston | 43 | 0 | 0 |
| | Hartford | 5 | 19 | 2 |

We conclude that friendship network along does not imply the office geographical community structure; however, with extra information from the attorneys' age and/or years with the firm, friendship network community structure has very strong relationship with the office location. Another interesting finding is that using both

covariates decreases the model performance a bit, which motivates us to explore more about how to select the best subsets of covariates in the future.

# Chapter 4

# Dynamic Network Community Discovery

## 4.1 Introduction

The traditional spectral clustering, SCORE algorithms, as well as our approaches in the last chapter are designed to discover communities in a static network. No historical information can be used to improve the cluster quality. On the other hand, in the computer science literature, an idea called evolutionary clustering has been discussed to deal with the clustering problem with a series of time-varying similarity matrices, in which the clustering results from the current similarity matrix is adjusted by the historical information. Evolutionary clustering was first discussed in 2006 [4]. One year later, the idea of temporal smoothness is integrated into the evolutionary clustering [5] to provide better clustering results in terms of less short-term noise and more adaptive long-term cluster drifts. The cost function of the evolutionary clustering problem is

$$\text{Cost}_t = \alpha(\text{CS}_t) + \beta(\text{CT}_t), \tag{4.1}$$

where CS stands for snapshot cost, representing the clustering quality of the current snapshot; CT stands for temporal cost, measuring the temporal smoothness guided by the historical information, $\alpha \in [0, 1]$ is the smoothness tuning parameter, and $\beta = 1 - \alpha$. The subscript $t$, $2 \leq t \leq T$, is the time. $t$ starts at 2 because no history can be borrowed from the temporal cost function at $t = 1$, the starting point of the observation.

As we discussed in the chapter 2, SCORE algorithm performs well in detecting

communities of a static network under the DCBM. At the meantime, evolutionary clustering is a newly proposed method to detect clusters over time when temporal smoothness is taken into consideration. In this chapter, we plan to integrate the idea of SCORE and evolutionary clustering together, and discuss three frameworks in terms of three philosophical ways of borrowing historical information,to adaptively discover the network communities when the observed network is evolving over time under the DCBM model assumption.

### 4.1.1 Settings

We assume that at time $t$, $1 \leq t \leq T$, there are $n$ nodes $\mathcal{V}_t = \{v_1, v_2, \ldots, v_n\}$, and $K$ clusters $\mathcal{V}_t = \{\mathcal{V}_t^{(1)}, \mathcal{V}_t^{(2)}, \ldots, \mathcal{V}_t^{(K)}\}$ in the network, where $\mathcal{V}_t^{(l)}$ is the set of nodes belonging to the $l$th community at time $t$. We assume there exists an unobserved partition $\mathcal{V}_t = \mathcal{V}_t^{(1)} \cup \mathcal{V}_t^{(2)} \cup \cdots \cup \mathcal{V}_t^{(K)}$. One 0/1-entry symmetric adjacency matrix $A_t \in \mathbb{S}^n$ is observed.

We denote 0/1-entry matrix $Z_t \in \mathbb{R}^{n \times K}$ to be the community membership matrix at time $t$, and $Z_t(i, l) = 1$ when node $i$ belongs to community $l$ at time $t$. Because a node belongs to one and only one community at each time, it is easy to see that each row of $Z_t$ has only one "1", and the rest of the entries are "0"s. Denote $Z_t = (z_{t1}, z_{t2}, \ldots, z_{tK})$. $\|z_{tl}\|_1 = |\mathcal{V}_t^{(l)}|$ is the size of community $l$ at time $t$. For any $l$ and $m$, $l \neq m$, $z_{tl}$ and $z_{tm}$ are orthogonal vectors, say $z_{tl}^T z_{tm} = 0$. Define $y_{tl} = z_{tl}/\sqrt{z_{tl}^T z_{tl}}$ be the normalized $z_{tl}$, and matrix $Y_t = (y_{t1}, y_{t2}, \ldots, y_{tK})$. It is obvious that $Y_t^T Y_t = I_K$. We define $X_t \in \mathbb{R}^{n \times K}$ such that $X_t^T X_t = I_K$. Columns of $X_t$ are orthonormal to each other (just like $Y_t$); however, we allow multiple non-zero entries in each row of $X_t$. $X_t$ can be considered as a relaxed or perturbed $Y_t$.

Define $\theta_t \in \mathbb{R}^n$ to be the degree vector, whose $i$th entry $\theta_t(i) \in (0, 1)$ is the degree parameter for node $i$ at time $t$. Let $\Theta_t = \mathrm{diag}(\theta_t)$ be the $n$ dimensional diagonal matrix. (*Remark:* degree parameter is **NOT** the node degree in the degree matrix.

Degree parameter $\theta_t$ measures the population popularity of a node, and the node degree is the number of observed edges connected to a node.)

### 4.1.2 Content

In Section 4.2, we introduce three frameworks whose main purpose is to discover communities with a series of observed time-varying adjacency matrices from a network with $n$ nodes under the DCBM. In Section 4.3, five simulated experiments are presented to compare the performance of three frameworks, as well as the SCORE algorithm which servers as a benchmark. In Section 4.4, we apply our proposed algorithm on Enron Email data. In Section 4.5, we discuss three interesting topics, including whether we should borrow more historical information, and how to modify the model to meet the needs when the number of communities $K$, and the number of nodes $n$ are varying over time.

## 4.2 Model Construction

There are two major subsections in this section. First, we introduce the community detection method at $t = 1$. At this point, there is no engaged historical information. Therefore, it is a static network community detection problem. SCORE algorithm [16], and a relaxation for k-means clustering [30] are detailedly described. Second, we discuss the case at $2 \leq t \leq T$. We use the same cost function as previously discussed in equation 4.1. Additionally, three frameworks of temporal smoothness are considered: Preserving Cluster Quality (PCQ), Preserving Cluster Membership (PCM), and Preserving Membership Degree (PMD). The idea of the first two frameworks (PCQ, PCM) was original proposed in [5], and the third one (PMD) is proposed by ourselves to tackled the degree heterogeneity problem under the DCBM assumption. An integrated algorithm is summarized in the end of this section after three frameworks are introduced.

#### 4.2.1 At Time $t = 1$

At time $t = 1$, the adjacency matrix $A_1 \in \mathbb{S}^n$ is observed. Based on the Degree Corrected Block Model (DCBM) [17], $A_1(i, j)$, $i > j$, is a realization of the Bernoulli random variable with linkage probability

$$\mathbb{P}[A_1(i, j) = 1] = \theta_1(i)\theta_1(j)B_1(l, m),$$

where $\theta_1(i)$ is the degree parameters for node $i$ at time $t = 1$, and node $i$ belongs to community $\mathcal{V}_1^{(l)}$, and node $j$ belongs to community $\mathcal{V}_1^{(m)}$. $B_1(l, m)$, the $(l, m)$th entry of the community affinity matrix $B_1 \in \mathbb{R}^{K\times K}$, is the baseline linkage parameter between the nodes in community $l$ and community $m$ at time $t = 1$. We can describe the upper triangular of $A_1$ to be a realization of $n(n - 1)/2$ independent Bernoulli random variables with linkage probabilities located in the corresponding positions in the matrix $\Omega_1 = \Theta_1 Z_1 B_1 Z_1^T \Theta_1$.

Spectral Clustering On Ratios-of-Eigenvectors (SCORE) [16] was designed to detect communities using an observed adjacency matrix under the DCBM. The algorithm contains three steps:

1. Compute the $K$ leading eigenvectors (associated with the largest $K$ absolute eigenvalues) of the adjacency matrix $A_1$: $\eta_{11}, \eta_{12}, \ldots \eta_{1K}$.

2. Compute matrix $R_1 \in \mathbb{R}^{n\times(K-1)}$ such that for $1 \le i \le n$ and $1 \le l \le K - 1$,

$$R_1(i, l) = \frac{\eta_{1(l+1)}(i)}{\eta_{11}(i)},$$

   which is the coordinate-wise ratio between the $i$th entry of the first leading eigenvector and the $i$th entry of the $l$th leading eigenvector.

3. Apply k-means algorithm on the rows of $R_1$ to detect cluster labels.

Define $R_1 = (r_{11}, r_{12}, \ldots, r_{1n})^T$. The objective function of the k-means algorithm in the third step, which should be minimized with respect to $\mathcal{V}_1^{(1)}, \mathcal{V}_1^{(2)}, \ldots, \mathcal{V}_1^{(K)}$ (or

equivalently $Z_1$), is

$$\mathrm{KM}_1 = \sum_{l=1}^{K} \sum_{i \in \mathcal{V}_1^{(l)}} \|r_{1i} - \mu_1^{(l)}\|^2,$$

where $\mu_1^{(l)} = \frac{\sum_{i \in \mathcal{V}_1^{(l)}} r_{1i}}{|V_1^{(l)}|}$ is the centroid (mean) of the $l$th community. A reformulated representation of k-means problem was discovered in 2002 [30]. The reformulated problem is related to the trace maximization associated with the Gram matrix. For the reason which will be specified in the next subsection, it is beneficial to state the reformulated cost function, which is

$$\mathrm{KM}_1 = \mathrm{tr}(R_1 R_1^T) - \mathrm{tr}[Y_1^T (R_1 R_1^T) Y_1],$$

where $Y_1$ is the normalized $Z_1$, and $R_1 R_1^T$ is the Gram matrix corresponding to $R_1^T$. Because $R_1 R_1^T$ is obtained from the observed adjacency matrix $A_1$, minimizing the cost function (with respect to $Y_1$) is equivalent to maximizing $\mathrm{tr}[Y_1^T (R_1 R_1^T) Y_1]$. We further relax the matrix $Y_1$ to $X_1$ with $X_1^T X_1 = I_K$. However, unlike $Y_1$, more than one non-zero entries are permitted in each row of $X_1$. $X_1$ is a relaxed or perturbed $Y_1$. One solution to maximize the relaxed function $\mathrm{tr}[X_1^T (R_1 R_1^T) X_1]$, by one of the variation of the Rayleigh-Ritz theorem in [21](P. 68), is the stack of $K$ eigenvectors associated with the largest $K$ (not absolute) eigenvalues of $R_1 R_1^T$. Note that if $\hat{X}_1$ is a solution, then for any $K$ dimensional orthogonal matrix $U$, $\hat{X}_1 U$ is also a solution. Therefore, the optimal $X_1$ is not unique. However, it is not a problem for us since network communities are exchangeable. We are able to obtain $\hat{Z}_1$, an estimate of $Z_1$, by applying k-means algorithm (or other clustering method) on the rows of $\hat{X}_1$.

In summary, at time $t = 1$, the algorithm to detect communities with $A_1$ is:

1. Compute the $K$ leading eigenvectors (associated with the largest $K$ absolute eigenvalues) of the adjacency matrix $A_1$: $\eta_{11}, \eta_{12}, \ldots \eta_{1K}$.

2. Compute matrix $R_1 \in \mathbb{R}^{n \times (K-1)}$ such that for $1 \le i \le n$ and $1 \le l \le K-1$,

$$R_1(i, l) = \frac{\eta_{1(l+1)}(i)}{\eta_{11}(i)},$$

which is the coordinate-wise ratio between the $i$th entry of the first leading eigenvector and the $i$th entry of the $l$th leading eigenvector.

3. Obtain $\hat{X}_1 \in \mathbb{R}^{n \times K}$ by stacking $K$ eigenvectors associated with the largest $K$ eigenvalues of the Gram matrix $W_1 = R_1 R_1^T$.

4. Apply k-means algorithm on the rows of $\hat{X}_1$ to find $\hat{Z}_1$.

#### 4.2.2 At Time $2 \leq t \leq T$

At time $2 \leq t \leq T$, we consider the cost function $\text{Cost}_t = \alpha(\text{CS}_t) + \beta(\text{CT}_t)$ from equation 4.1, which is commonly used in evolutionary clustering literature [4][5]. The $t = 1$ case can be framed into this cost function as well with $\text{CS}_1 = \text{KM}_1$ and $\alpha = 1$.

The snapshot cost (CS) measures the cluster quality using the current observation $A_t$ at time $t$. Define $\text{CS}_t = \text{KM}_t$. On the other hand, temporal cost (CT) measures the smoothness of the evolution of the network. There is no consensus of the mathematical definition of the smoothness in the literature. Two previously proposed frameworks [5], Preserving Cluster Quality (PCQ) and Preserving Cluster Membership (PCM), are trying to address this problem. In this chapter, we will discuss how to use PCQ and PCM frameworks in the dynamic network community discovery. Moreover, we propose a new framework called Preserving Membership Degree (PMD) to handle the situation with severe degree heterogeneity. By the end of the day, we find PMD outperforms PCQ and PCM in terms of stability and accuracy in complicated network settings by simulations.

##### Preserving Cluster Quality (PCQ)

Under the PCQ framework, the benefit of adding the temporal cost is to find a partition at time $t$ such that both $\text{CS}_t$ and $\text{CS}_{t-1}$ are small. In other words, if several partitions lead to the same $\text{CS}_t$, then the one leading to the smallest $\text{CS}_{t-1}$ wins.

Denote $\mathrm{KM}_t(Z_s)$ to be the k-means objective function at time $t$ evaluated by $Z_s$, the partition at time $s$. The cost function under PCQ framework is

$$\begin{aligned}
\mathrm{Cost(PCQ)}_t &= \alpha \mathrm{CS}_t + \beta \mathrm{CT}_t \\
&= \alpha \mathrm{KM}_t(Z_t) + \beta \mathrm{KM}_{t-1}(Z_t) \\
&= \alpha\{\mathrm{tr}(R_t R_t^T) - \mathrm{tr}[Y_t^T (R_t R_t^T) Y_t]\} \\
&\quad + \beta\{\mathrm{tr}(R_{t-1} R_{t-1}^T) - \mathrm{tr}[Y_t^T (R_{t-1} R_{t-1}^T) Y_t]\} \\
&= \{\alpha \mathrm{tr}(R_t R_t^T) + \beta \mathrm{tr}(R_{t-1} R_{t-1}^T)\} \\
&\quad - \{\alpha \mathrm{tr}[Y_t^T (R_t R_t^T) Y_t] + \beta \mathrm{tr}[Y_t^T (R_{t-1} R_{t-1}^T) Y_t]\} \\
&= \{\alpha \mathrm{tr}(R_t R_t^T) + \beta \mathrm{tr}(R_{t-1} R_{t-1}^T)\} - \{\mathrm{tr}[Y_t^T (\alpha R_t R_t^T + \beta R_{t-1} R_{t-1}^T) Y_t]\} \\
&= \{\alpha \mathrm{tr}(W_t) + \beta \mathrm{tr}(W_{t-1})\} - \{\mathrm{tr}[Y_t^T (\alpha W_t + \beta W_{t-1}) Y_t]\}, \qquad (4.2)
\end{aligned}$$

where $Y_t$ is the normalized $Z_t$, $R_t$ is an $n$ by $K-1$ matrix, whose entries are the coordinate-wise ratio between the first leading eigenvector and other $K-1$ leading eigenvectors computed by the adjacency matrix $A_t$, and $W_t = R_t R_t^T$ is a Gram matrix. In equation 4.2, the first part is known. Minimizing the cost function is equivalent to maximizing the second part $\mathrm{tr}[Y_t^T(\alpha W_t + \beta W_{t-1}) Y_t]$ with respect to $Y_t$. We further relax $Y_t$ to $X_t$ using the same idea of the relaxation from $Y_1$ to $X_1$ (e.g. $X_t^T X_t = I_K$, and more than one non-zero entries in each row of $X_t$). One optimal estimate $\hat{X}_t$, aiming to maximize $\mathrm{tr}[X_t^T(\alpha W_t + \beta W_{t-1}) X_t]$, is the stack of $K$ eigenvectors associated with the largest $K$ eigenvalues of $\alpha W_t + \beta W_{t-1}$. Eventually, apply k-means algorithm on the rows of $\hat{X}_t$ gives an estimate of the community membership matrix $\hat{Z}_t$. Now it is easy to see why we need to reformulate the original k-means cost function to the trace representation. By doing so, two k-means objective functions, $\mathrm{KM}_t(Z_t)$ and $\mathrm{KM}_{t-1}(Z_t)$, can be integrated together to simplify the optimization procedures.

Preserving Cluster Membership (PCM)

Under the PCM framework, unlike using $W_{t-1} = R_{t-1}R_{t-1}^T$ as historical information in PCQ, we borrow the information **only** from historical community structure at time $t-1$. Define the distance function between $X_t$ and $X_{t-1}$ to be

$$\text{dist}(X_t, X_{t-1}) = \frac{1}{2}\|X_tX_t^T - X_{t-1}X_{t-1}^T\|_F^2,$$

which measures the distance between the subspace spanned by the columns of $X_t$ and $X_{t-1}$. Note that the defined distance function is invariant to the $X_tU$ transformation, where $U$ is any $K$ dimensional orthogonal matrix. With the well-defined distance function, the cost function under PCM framework is

$$\begin{aligned}
\text{Cost(PCM)}_t &= \alpha\text{CS}_t + \beta\text{CT}_t \\
&= \alpha\text{KM}_t(Z_t) + \beta\text{dist}(X_t, X_{t-1}) \\
&= \alpha\{\text{tr}(R_tR_t^T) - \text{tr}[Y_t^T(R_tR_t^T)Y_t]\} + \frac{\beta}{2}\|X_tX_t^T - X_{t-1}X_{t-1}^T\|_F^2 \\
&= \alpha\{\text{tr}(R_tR_t^T) - \text{tr}[Y_t^T(R_tR_t^T)Y_t]\} \\
&\quad + \frac{\beta}{2}\text{tr}[(X_tX_t^T - X_{t-1}X_{t-1}^T)^T(X_tX_t^T - X_{t-1}X_{t-1}^T)] \\
&= \alpha\{\text{tr}(R_tR_t^T) - \text{tr}[Y_t^T(R_tR_t^T)Y_t]\} \\
&\quad + \frac{\beta}{2}\left[\text{tr}(X_tX_t^TX_tX_t^T) - 2\text{tr}(X_tX_t^TX_{t-1}X_{t-1}^T) + \text{tr}(X_{t-1}X_{t-1}^TX_{t-1}X_{t-1}^T)\right] \\
&= \left\{\alpha\text{tr}(R_tR_t^T) + \frac{\beta}{2}\left[\text{tr}(X_tX_t^TX_tX_t^T) + \text{tr}(X_{t-1}X_{t-1}^TX_{t-1}X_{t-1}^T)\right]\right\} \\
&\quad - \left\{\alpha\text{tr}[Y_t^T(R_tR_t^T)Y_t] + \beta\text{tr}(X_tX_t^TX_{t-1}X_{t-1}^T)\right\} \\
&= \left\{\alpha\text{tr}(R_tR_t^T) + \frac{\beta}{2}[K+K]\right\} \qquad (4.3) \\
&\quad - \left\{\alpha\text{tr}[Y_t^T(R_tR_t^T)Y_t] + \beta\text{tr}(X_t^TX_{t-1}X_{t-1}^TX_t)\right\} \\
&\overset{Relax}{=} \left\{\alpha\text{tr}(R_tR_t^T) + \beta K\right\} - \text{tr}[X_t^T(\alpha R_tR_t^T + \beta X_{t-1}X_{t-1}^T)X_t] \qquad (4.4) \\
&= [\alpha\text{tr}(W_t) + \beta K] - \text{tr}[X_t^T(\alpha W_t + \beta X_{t-1}X_{t-1}^T)X_t], \qquad (4.5)
\end{aligned}$$

where the fact $\text{tr}(X_tX_t^TX_tX_t^T) = \text{tr}(X_t^TX_tX_t^TX_t) = \text{tr}(I_KI_K) = K$ is used in equation 4.3, and $Y_t$ is relaxed to $X_t$ in equation 4.4. A solution path to minimize the cost

function under the PCM framework is to stack the $K$ eigenvectors associated with the largest $K$ eigenvalues of $\alpha W_t + \beta \hat{X}_{t-1}\hat{X}_{t-1}^T$ to obtain $\hat{X}_t$, followed by applying k-means algorithm on rows of $\hat{X}_t$ to find $\hat{Z}_t$.

Preserving Membership Degree (PMD)

In both PCQ and PCM frameworks, degree heterogeneity is not taken into consideration because the intermediate step of computing $R_t$ largely removes the degree effect. The question here is "Whether degree plays an rule in the evolution of a network?" If the answer is negative, indeed we can simply use PCQ or PCM framework to finalize the community discovery procedure. However, if the answer is positive, it is natural to ask the following question "What mechanism is degree heterogeneity functioning to influence the community memberships in the network development process to the next time step?" **One intuition is that nodes with relatively high degree tend to stay in the same community from $t-1$ to $t$, comparing to those marginal nodes who have fewer within community degree.** Our goal is to construct a framework to implement this intuition.

Recall that $\Theta_t \in \mathbb{S}^n$ is the diagonal matrix with diagonal entries be the degree parameter $\theta_t(i)$, i=1,2,..., n. Therefore, $\Theta_t Z_t \in \mathbb{R}^{n\times K}$ is the matrix whose $(i,l)$th entry is $\theta_t(i)\mathbb{I}\left(v_i \in \mathcal{V}_t^{(l)}\right)$. $\mathbb{I}\left(v_i \in \mathcal{V}_t^{(l)}\right)$ is an indicator function, and its value is 1 when node $i$ belongs to community $l$ at time $t$. Define $\tilde{\Theta}_t Z_t$ to be the normalized $\Theta_t Z_t$ with entries $[\tilde{\Theta}_t Z_t](i,l) = \frac{\theta_t(i)}{\sqrt{\sum_{j:v_j\in\mathcal{V}_t^{(l)}}\theta_t(j)^2}}\mathbb{I}\left(v_i \in \mathcal{V}_t^{(l)}\right)$. Obviously, the $(i,i)$th entry of the diagonal matrix $\tilde{\Theta}_t$ is $\frac{\theta_t(i)}{\sum_{j:v_j\in\mathcal{V}_t^{(l)}}\theta_t(j)^2}$. Denote $\tilde{\Theta}_t\tilde{X}_t$ to be the relaxed version of $\tilde{\Theta}_t Z_t$ such that $(\tilde{\Theta}_t\tilde{X}_t)^T(\tilde{\Theta}_t\tilde{X}_t) = I_K$, and multiple non-zero entries are allowed in each row of $\tilde{\Theta}_t\tilde{X}_t$.

It is critical to understand the relationship between two $n$ by $K$ orthonormal matrices $\tilde{\Theta}_t\tilde{X}_t$ and $X_t$. As stated, $\tilde{\Theta}_t\tilde{X}_t$ is the relaxed version of $\tilde{\Theta}_t Z_t$, and $X_t$ is the relaxed version of $Y_t$. Note that $Y_t$ can be reformulated as $\bar{\Theta}_t Z_t$, where the $(i,i)$th

entry of the diagonal matrix $\bar{\Theta}_t$ is $\frac{1}{\sqrt{\sum_{j:v_j \in \mathcal{V}_t^{(l)}} 1}}$. It is easy to see that $\bar{\Theta}_t Z_t$ and $\tilde{\Theta}_t Z_t$ are the same when all the population degree parameters are the same, say $\theta_t(1) = \theta_t(2) = \cdots = \theta_t(n)$. Besides, we can show that in more general cases, the relationship between $Y_t$ and $\tilde{\Theta}_t Z_t$ is $Y_t = \bar{\Theta}_t \tilde{\Theta}_t^{-1}(\tilde{\Theta}_t Z_t)$, or equivalently $\tilde{\Theta}_t Z_t = \tilde{\Theta}_t \bar{\Theta}_t^{-1} Y_t$. It is reasonable to assume the relationship between $X_t$ and $\tilde{\Theta}_t \tilde{X}_t$ is $X_t \approx \bar{\Theta}_t \tilde{\Theta}_t^{-1}(\tilde{\Theta}_t \tilde{X}_t)$, or $\tilde{\Theta}_t \tilde{X}_t \approx \tilde{\Theta}_t \bar{\Theta}_t^{-1} X_t$.

The distance function designed for the temporal cost function, aiming to ingratiate our intuition, is

$$\text{dist}(\tilde{\Theta}_t \tilde{X}_t, \tilde{\Theta}_{t-1} \tilde{X}_{t-1}) = \frac{1}{2} \| \tilde{\Theta}_t \tilde{X}_t (\tilde{\Theta}_t \tilde{X}_t)^T - \tilde{\Theta}_{t-1} \tilde{X}_{t-1} (\tilde{\Theta}_{t-1} \tilde{X}_{t-1})^T \|_F^2,$$

where $\tilde{\Theta}_t \tilde{X}_t$ is the relaxed version of $\tilde{\Theta}_t Z_t$, which preserves both community membership and within community degree for each node. With the well-defined distance function, the cost function under PMD framework is

$$\begin{aligned}
\text{Cost(PMD)}_t &= \alpha \text{CS}_t + \beta \text{CT}_t \\
&= \alpha \text{KM}_t(Z_t) + \beta \text{dist}(\tilde{\Theta}_t \tilde{X}_t, \tilde{\Theta}_{t-1} \tilde{X}_{t-1}) \\
&= \alpha \{ \text{tr}(R_t R_t^T) - \text{tr}[Y_t^T (R_t R_t^T) Y_t] \} \\
&\quad + \frac{\beta}{2} \| \tilde{\Theta}_t \tilde{X}_t (\tilde{\Theta}_t \tilde{X}_t)^T - \tilde{\Theta}_{t-1} \tilde{X}_{t-1} (\tilde{\Theta}_{t-1} \tilde{X}_{t-1})^T \|_F^2 \\
&= \alpha \{ \text{tr}(R_t R_t^T) - \text{tr}[Y_t^T (R_t R_t^T) Y_t] \} \\
&\quad + \frac{\beta}{2} \text{tr} \left\{ \left( \tilde{\Theta}_t \tilde{X}_t \tilde{X}_t^T \tilde{\Theta}_t \tilde{\Theta}_t \tilde{X}_t \tilde{X}_t^T \tilde{\Theta}_t \right) \right\} \\
&\quad + \frac{\beta}{2} \text{tr} \left\{ \left( \tilde{\Theta}_{t-1} \tilde{X}_{t-1} \tilde{X}_{t-1}^T \tilde{\Theta}_{t-1} \tilde{\Theta}_{t-1} \tilde{X}_{t-1} \tilde{X}_{t-1}^T \tilde{\Theta}_{t-1} \right) \right\} \\
&\quad - \frac{\beta}{2} \text{tr} \left\{ 2 \tilde{\Theta}_t \tilde{X}_t \tilde{X}_t^T \tilde{\Theta}_t \tilde{\Theta}_{t-1} \tilde{X}_{t-1} \tilde{X}_{t-1}^T \tilde{\Theta}_{t-1} \right\} \\
&= \alpha \{ \text{tr}(R_t R_t^T) - \text{tr}[Y_t^T (R_t R_t^T) Y_t] \} \\
&\quad + \frac{\beta}{2} \text{tr} \left\{ \left( \tilde{\Theta}_t \tilde{X}_t (I_K) \tilde{X}_t^T \tilde{\Theta}_t \right) + \left( \tilde{\Theta}_{t-1} \tilde{X}_{t-1} (I_K) \tilde{X}_{t-1}^T \tilde{\Theta}_{t-1} \right) \right\} \\
&\quad - \beta \text{tr} \left\{ \tilde{X}_t^T \tilde{\Theta}_t \tilde{\Theta}_{t-1} \tilde{X}_{t-1} \tilde{X}_{t-1}^T \tilde{\Theta}_{t-1} \tilde{\Theta}_t \tilde{X}_t \right\} \\
&= \alpha \{ \text{tr}(R_t R_t^T) - \text{tr}[Y_t^T (R_t R_t^T) Y_t] \}
\end{aligned}$$

$$
\begin{aligned}
&+ \frac{\beta}{2}\text{tr}\left\{\left(\tilde{X}_t^T\tilde{\Theta}_t\tilde{\Theta}_t\tilde{X}_t\right) + \left(\tilde{X}_{t-1}^T\tilde{\Theta}_{t-1}\tilde{\Theta}_{t-1}\tilde{X}_{t-1}\right)\right\} \\
&- \beta\text{tr}\left\{\left(\tilde{\Theta}_t\tilde{X}_t\right)^T\left(\tilde{\Theta}_{t-1}\tilde{X}_{t-1}\tilde{X}_{t-1}^T\tilde{\Theta}_{t-1}\right)\left(\tilde{\Theta}_t\tilde{X}_t\right)\right\} \\
&\overset{Relax}{=} \alpha\left\{\text{tr}\left(R_tR_t^T\right) - \text{tr}\left[X_t^T\left(R_tR_t^T\right)X_t\right]\right\} \\
&+ \beta K - \beta\text{tr}\left\{\left(\tilde{\Theta}_t\tilde{X}_t\right)^T\left(\tilde{\Theta}_{t-1}\tilde{X}_{t-1}\tilde{X}_{t-1}^T\tilde{\Theta}_{t-1}\right)\left(\tilde{\Theta}_t\tilde{X}_t\right)\right\} \\
&= \alpha\left\{\text{tr}\left(R_tR_t^T\right) - \text{tr}\left[X_t^T\left(R_tR_t^T\right)X_t\right]\right\} \\
&+ \beta K - \beta\text{tr}\left\{X_t^T\left(\bar{\Theta}_t^{-1}\tilde{\Theta}_t\tilde{\Theta}_{t-1}\bar{\Theta}_{t-1}^{-1}X_{t-1}X_{t-1}^T\bar{\Theta}_{t-1}^{-1}\tilde{\Theta}_{t-1}\tilde{\Theta}_t\bar{\Theta}_t^{-1}\right)X_t\right\}
\end{aligned}
\tag{4.6}
$$

$$
\begin{aligned}
&= \alpha\text{tr}\left(R_tR_t^T\right) + \beta K \\
&- \text{tr}\left[X_t^T\left(\alpha R_tR_t^T + \beta\bar{\Theta}_t^{-1}\tilde{\Theta}_t\tilde{\Theta}_{t-1}\bar{\Theta}_{t-1}^{-1}X_{t-1}X_{t-1}^T\bar{\Theta}_{t-1}^{-1}\tilde{\Theta}_{t-1}\tilde{\Theta}_t\bar{\Theta}_t^{-1}\right)X_t\right]
\end{aligned}
\tag{4.7}
$$

$$
\begin{aligned}
&= \alpha\text{tr}(W_t) + \beta K \\
&- \text{tr}\left[X_t^T\left(\alpha W_t + \beta\bar{\Theta}_t^{-1}\tilde{\Theta}_t\tilde{\Theta}_{t-1}\bar{\Theta}_{t-1}^{-1}X_{t-1}X_{t-1}^T\bar{\Theta}_{t-1}^{-1}\tilde{\Theta}_{t-1}\tilde{\Theta}_t\bar{\Theta}_t^{-1}\right)X_t\right]
\end{aligned}
\tag{4.8}
$$

where the $\tilde{\Theta}_t\tilde{X}_t = \tilde{\Theta}_t\bar{\Theta}_t^{-1}X_t$ is used in equation 4.6. In order to see the difference between PCM and PMD frameworks, let's focus on the second part of equation 4.4 and 4.7.

$$
\begin{aligned}
&\text{PCM:} \quad \text{tr}\left[X_t^T\left(\alpha R_tR_t^T + \beta X_{t-1}X_{t-1}^T\right)X_t\right] \\
&\text{PMD:} \quad \text{tr}\left[X_t^T\left(\alpha R_tR_t^T + \beta\bar{\Theta}_t^{-1}\tilde{\Theta}_t\tilde{\Theta}_{t-1}\bar{\Theta}_{t-1}^{-1}X_{t-1}X_{t-1}^T\bar{\Theta}_{t-1}^{-1}\tilde{\Theta}_{t-1}\tilde{\Theta}_t\bar{\Theta}_t^{-1}\right)X_t\right]
\end{aligned}
$$

There are two sets of products of degree matrices, $\bar{\Theta}_t^{-1}\tilde{\Theta}_t$ and $\bar{\Theta}_{t-1}^{-1}\tilde{\Theta}_{t-1}$, worth mentioning. When $[\bar{\Theta}_t^{-1}\tilde{\Theta}_t](i,i)$ is greater than 1, node $v_i$ has relatively higher degree than others belonging to the same community at time $t$. We call $\bar{\Theta}_t^{-1}\tilde{\Theta}_t$ the *degree contest matrix* at time $t$. The product of two degree contest matrices at time $t-1$ and $t$, $\left(\bar{\Theta}_t^{-1}\tilde{\Theta}_t\right)\left(\tilde{\Theta}_{t-1}\bar{\Theta}_{t-1}^{-1}\right)$, averages the degree contest level between two consecutive time points.

$X_{t-1}$ is the historical information, and $R_t$ is the information from the current adjacency matrix. Nodes with high averaged degree contest level borrow more historical information comparing to PCQ and PCM, which is consistent to our intuition. For example, in social network, core members in a community usually stay longer in the community comparing to the marginal members.

In equation 4.7 and 4.8, $\bar{\Theta}_t^{-1}\tilde{\Theta}_t$ is unknown. We can estimate $\bar{\Theta}_{t-1}^{-1}\tilde{\Theta}_{t-1}$ based on estimated $\hat{Z}_{t-1}$. A reasonable variation of the cost function is to replace $\bar{\Theta}_t^{-1}\tilde{\Theta}_t$ with the estimate of $\bar{\Theta}_{t-1}^{-1}\tilde{\Theta}_{t-1}$. By doing so, we assume the degree contest matrices are approximately the same between two successive time steps. Thus, the working PMD cost function is

$$\text{Cost(PMD)}_t = \alpha\text{tr}(W_t) + \beta K - \text{tr}\left[X_t^T\left(\alpha W_t + \beta\hat{\tilde{\Theta}}_{t-1}^2\hat{\bar{\Theta}}_{t-1}^{-2}\hat{X}_{t-1}\hat{X}_{t-1}^T\hat{\bar{\Theta}}_{t-1}^{-2}\hat{\tilde{\Theta}}_{t-1}^2\right)X_t\right]$$

where $\hat{\tilde{\Theta}}_{t-1}$, $\hat{\bar{\Theta}}_{t-1}$, and $\hat{X}_{t-1}$ are the estimate of $\tilde{\Theta}_{t-1}$, $\bar{\Theta}_{t-1}$, and $X_{t-1}$. Follow the same procedure as we stated before, $\hat{X}_t$ is obtained by stacking $K$ eigenvectors associated with the largest $K$ eigenvalues of $\alpha W_t + \beta\hat{\tilde{\Theta}}_{t-1}^2\hat{\bar{\Theta}}_{t-1}^{-2}\hat{X}_{t-1}\hat{X}_{t-1}^T\hat{\bar{\Theta}}_{t-1}^{-2}\hat{\tilde{\Theta}}_{t-1}^2$, which can be utilized to find $\hat{Z}_t$ with the normal k-means algorithm.

### 4.2.3 Algorithm

The dynamic network community discovery algorithm is

- At $t = 1$, observe the adjacency matrix $A_1$, and

  1. Compute the $K$ leading eigenvectors (associated with the largest $K$ absolute eigenvalues) of the adjacency matrix $A_1$: $\eta_{11}, \eta_{12}, \ldots \eta_{1K}$.
  2. Compute matrix $R_1 \in \mathbb{R}^{n\times(K-1)}$ such that for $1 \le i \le n$ and $1 \le l \le K-1$,

     $$R_1(i,l) = \frac{\eta_{1(l+1)}(i)}{\eta_{11}(i)},$$

     which is the coordinate-wise ratio between the $i$th entry of the first leading eigenvector and the $i$th entry of the $l$th leading eigenvector.

3. Compute the Gram matrix $W_1 = R_1 R_1^T$.

4. Obtain $\hat{X}_1 \in \mathbb{R}^{n \times K}$ by stacking $K$ eigenvectors associated with the largest $K$ eigenvalues of the similarity matrix $W_1$, and apply k-means algorithm on the rows of $\hat{X}_1$ to find $\hat{Z}_1$.

- At $2 \leq t \leq T$, observe the adjacency matrix $A_t$, and

1. Compute the $K$ leading eigenvectors (associated with the largest $K$ absolute eigenvalues) of the adjacency matrix $A_t$: $\eta_{t1}, \eta_{t2}, \ldots \eta_{tK}$.

2. Compute matrix $R_t \in \mathbb{R}^{n \times (K-1)}$ such that for $1 \leq i \leq n$ and $1 \leq l \leq K-1$,

$$R_1(i,l) = \frac{\eta_{1(l+1)}(i)}{\eta_{11}(i)},$$

which is the coordinate-wise ratio between the $i$th entry of the first leading eigenvector and the $i$th entry of the $l$th leading eigenvector.

3. Compute the Gram matrix $W_t = R_t R_t^T$.

4. Obtain $\hat{X}_t \in \mathbb{R}^{n \times K}$ by stacking $K$ eigenvectors associated with the largest $K$ eigenvalues of

   - **PCQ:** $\alpha W_t + \beta W_{t-1}$,
   - **PCM:** $\alpha W_t + \beta \hat{X}_{t-1} \hat{X}_{t-1}^T$,
   - **PMD:** $\alpha W_t + \beta \hat{\tilde{\Theta}}_{t-1}^2 \hat{\tilde{\Theta}}_{t-1}^{-2} \hat{X}_{t-1} \hat{X}_{t-1}^T \hat{\tilde{\Theta}}_{t-1}^{-2} \hat{\tilde{\Theta}}_{t-1}^2$,
     where $\hat{\tilde{\Theta}}_{t-1} = \text{diag}\left[\hat{Z}_{t-1}(\hat{Z}_{t-1}^T \hat{Z}_{t-1})^{-\frac{1}{2}} \mathbf{1}_K\right]$,
     and $\hat{\tilde{\Theta}}_{t-1} = \text{diag}\left[\hat{\Theta}_{t-1} \hat{Z}_{t-1}(Z_{t-1}^T \hat{\Theta}_{t-1}^2 Z_{t-1})^{-\frac{1}{2}} \mathbf{1}_K\right]$
     with $\hat{\Theta}_{t-1} = \text{diag}\,(A_{t-1} \mathbf{1}_n)$,

   and apply k-means algorithm on the rows of $\hat{X}_t$ to find $\hat{Z}_t$.

### 4.3 Simulations

We present the simulation results from five experiments, which compare the performance of PCQ, PCM, and PMD three frameworks in simple and complicated

circumstances.

### 4.3.1 Experiment 1

In experiment 1, we set $(n, K, T, \text{rep}) = (1000, 2, 5, 50)$. The community affinity matrix $B$ has diagonal elements 1 and off-diagonal elements 0.5. Community labels follow $l_1(i) = \text{Ber}(0.5)+1$, and the degrees of nodes are the same, say $\theta_t(i) = 0.2$. The term "rep" indicates the number of times we repeat the experiment. We can describe this situation as observing an SBM-based non-evolutionary network $T$ times. We set $\alpha$ to be 0.5 and 0.9 in experiment 1(a) and 1(b) respectively.

The results of experiment 1 are summarized in Figure 4.1 and Table 4.1. Figure 4.1 displays the mean error rate of benchmark method (BM) and three frameworks over time. Error rate is defined by the number of misclassified nodes divided by $n$. It is the same as the misclassification rate in the last chapter. The benchmark method is the SCORE algorithm using only the "current" adjacency matrix $A_t$. We conclude that (1) PCQ outperforms the rest three methods when there is no degree and evolutionary effect; (2) PMD is slightly better than BM and PCM; (3) the choice of $\alpha$ is quite robust in terms of the clustering accuracy in this simple network setting. The means and standard deviations of the error rate for four methods are recorded in Table 4.1. Similar tables are not provided for later experiments since the comparison can be done by looking at the error rate plot directly.

### 4.3.2 Experiment 2

Compare to experiment 1, we only add the degree effect in experiment 2, by assuming observing a series of adjacency matrices generated from the same DCBM-based network. $(n, K, T, \alpha, \text{rep}) = (1200, 2, 5, 0.8, 50)$. The community affinity matrix $B$ and the mechanism to generate community labels are the same as in experiment 1. Besides, we set $c_0 = 0.5$ and $d_0 = 0.05$, and let the degrees of nodes vary via the

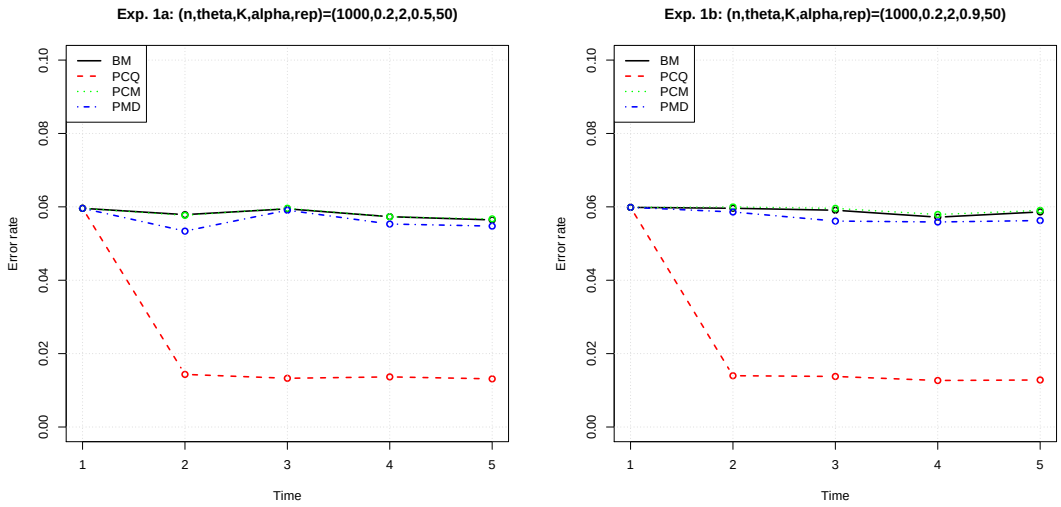


Figure 4.1 Experiment 1(a) (left) and 1(b) (right) Error Rate

Table 4.1 Experiment 1(a) (top) and 1(b) (bottom) Mean(Sd)

| Time | BM | PCQ | PCM | PMD |
|---|---|---|---|---|
| 1 | 0.0596(0.0101) | 0.0596(0.0101) | 0.0596(0.0101) | 0.0596(0.0101) |
| 2 | 0.0579(0.0094) | 0.0143(0.0041) | 0.0577(0.0092) | 0.0534(0.0014) |
| 3 | 0.0595(0.0098) | 0.0133(0.0036) | 0.0596(0.0100) | 0.0591(0.0108) |
| 4 | 0.0573(0.0095) | 0.0137(0.0036) | 0.0573(0.0092) | 0.0553(0.0122) |
| 5 | 0.0564(0.0087) | 0.0013(0.0037) | 0.0567(0.0088) | 0.0548(0.0116) |

| Time | BM | PCQ | PCM | PMD |
|---|---|---|---|---|
| 1 | 0.0599(0.0094) | 0.0599(0.0094) | 0.0599(0.0094) | 0.0599(0.0094) |
| 2 | 0.0596(0.0097) | 0.0140(0.0047) | 0.0600(0.0093) | 0.0586(0.0100) |
| 3 | 0.0591(0.0084) | 0.0138(0.0040) | 0.0596(0.0085) | 0.0561(0.0132) |
| 4 | 0.0572(0.0063) | 0.0127(0.0037) | 0.0580(0.0064) | 0.0559(0.0104) |
| 5 | 0.0586(0.0106) | 0.0128(0.0041) | 0.0590(0.0109) | 0.0563(0.0151) |

following formula:

$$2(\text{a}): \qquad \theta_1(i) = d_0 + (c_0 - d_0)\left(\frac{i}{n}\right),$$
$$2(\text{b}): \qquad \theta_1(i) = d_0 + (c_0 - d_0)\left(\frac{i}{n}\right)^2.$$

From 2(a) to 2(b), the average degree of the nodes in the network is decreasing, which creates more sparse adjacency matrix (less edges) with more hubs. The simulation results in Figure 4.2 indicate that PCQ dominates the all methods and PMD is the second best one again.

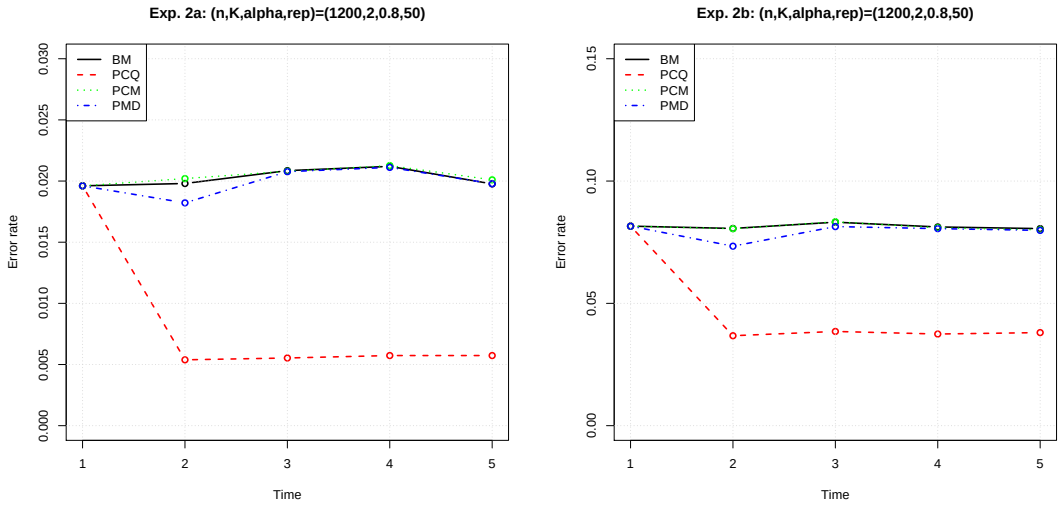


Figure 4.2 Experiment 2(a) (left) and 2(b) (right) Error Rate

### 4.3.3 Experiment 3

Compare to experiment 1, we only add the community evolutionary effect in experiment 3. $(n, K, T, \alpha, \text{rep}) = (1000, 2, 5, 0.8, 50)$. The community affinity matrix $B$ is the same as in experiment 1, and $\theta_t(i) = 0.2$. We generate the initial community label by $l_1(i) = \text{Ber}(0.5) + 1$, and allow $l_t(i)$ to alter over time. In experiment 3(a), we set the probability for a node to change its community to be 0.05, and it is

increased to 0.2 in 3(b). The results of experiment 3 are summarized in Figure 4.3. It is surprising that PCQ fails in faster evolving networks, when more nodes change communities over time. It gives us a hint that PCQ might not be a good choice in the case when nodes frequently change their community labels.

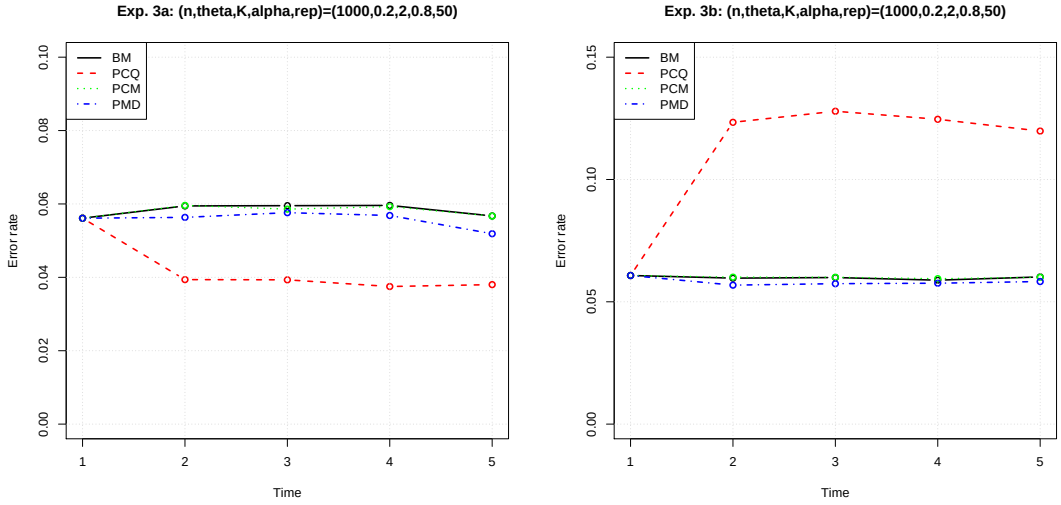


Figure 4.3 Experiment 3(a) (left) and 3(b) (right) Error Rate

### 4.3.4 Experiment 4

In experiment 4, we consider a more complicated situation by combining the degree and evolutionary effects together. $(n, K, T, \alpha, \text{rep}) = (1200, 3, 5, 0.8, 50)$. The community affinity matrix $B$ has diagonal elements 1, $P(1,2) = P(2,3) = 0.5$, and $P(1,3) = 0.1$. The initial community label is generated by $l_1(i) = \text{Multinomial}(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. Let $c_0 = 0.5$, $d_0 = 0.1$, $c_0^* = 0.95$, and $d_0^* = 0.8$. We allow node $i$ to switch to a new community by probability $1 - \text{prob}_i$ between two consecutive time, and $\text{prob}_i$ depends on the degree of the nodes. Based on the intuition discussed in section 4.2.2, the nodes with higher degree has higher $\text{prob}_i$, the probability of keeping the same

community label. Degree and $\text{prob}_i$ are assumed as follows:

$$\text{4(a):} \qquad \theta_1(i) = d_0 + (c_0 - d_0)\left(\frac{i}{n}\right), \quad \text{prob}_i = d_0^* + (c_0^* - d_0^*)\left(\frac{i}{n}\right);$$
$$\text{4(b):} \qquad \theta_1(i) = d_0 + (c_0 - d_0)\left(\frac{i}{n}\right)^2, \quad \text{prob}_i = d_0^* + (c_0^* - d_0^*)\left(\frac{i}{n}\right)^2.$$

The results of experiment 4 are shown in Figure 4.4. Both PCQ and PMD perform significantly better than the SCORE algorithm, and PCQ works the best.

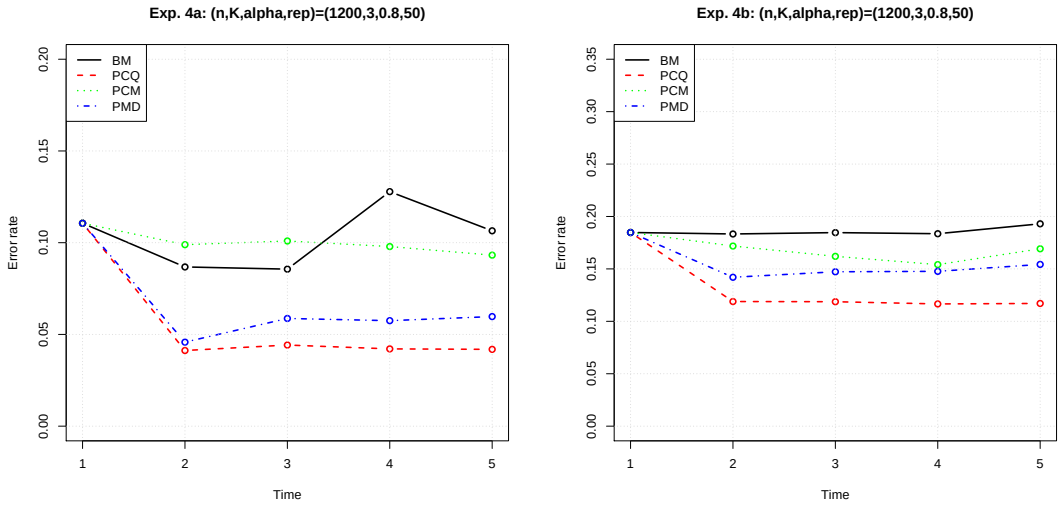


Figure 4.4 Experiment 4(a) (left) and 4(b) (right) Error Rate

### 4.3.5 Experiment 5

In experiment 5, we explore the case with more communities, and simultaneously, exam whether $\alpha$ is still robust in more complicated circumstances comparing to the simple ones in experiment 1. $(K, T, \text{rep}) = (5, 5, 50)$. The community affinity matrix

is

$$B = \begin{bmatrix} 1 & 0.5 & 0.3 & 0.2 & 0.1 \\ 0.5 & 1 & 0.1 & 0.1 & 0.2 \\ 0.3 & 0.1 & 1 & 0.2 & 0.3 \\ 0.2 & 0.1 & 0.2 & 1 & 0.5 \\ 0.1 & 0.2 & 0.3 & 0.5 & 1 \end{bmatrix},$$

and the initial community label is generated by $l_1(i) = \text{Multinomial}(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$. For experiment 5(a) and 5(b), the degree function and $\text{prob}_i$ follows the same ones in experiment 4(a) and 4(b), respectively. Additionally, for each setting we run the simulation with $\alpha = 0.5$ and $\alpha = 0.8$. Results of experiment 5 are summarized in Figure 4.5. It is very impressive to see that PMD framework has accurate and stable performances in different values of $\alpha$ in two network settings. PCQ only performs well when a proper $\alpha$ is chosen, which is $\alpha = 0.8$ in this experiment.

### 4.3.6 Simulation Summary

In the previous five experiments, we compare three frameworks: PCQ, PCM, and PMD, with the SCORE algorithm as the benchmark. The complexity of experiments is controlled by degree effect, community evolutionary effect, and the choice of $\alpha$. Generally speaking, PCQ performs well in simple cases; however, the accuracy cannot be guaranteed with high degree of heterogeneity and more dynamic situations.

We observe that (1) PCQ fails when the evolutionary effect is strong from experiment 3(b); (2) PMD outperforms PCQ when $\alpha = 0.5$ from experiment 5 indicating that we might need to choose a proper $\alpha$ to have a high clustering accuracy under PCQ. Instead, PMD works better than the benchmark and PCM in all simulation experiments. It fits our expectation because PCM is just a special case of PMD assuming all node degrees are the same. Another good property of PMD is that it is not sensitive to the choice of $\alpha$, which is indeed unknown in real world data.

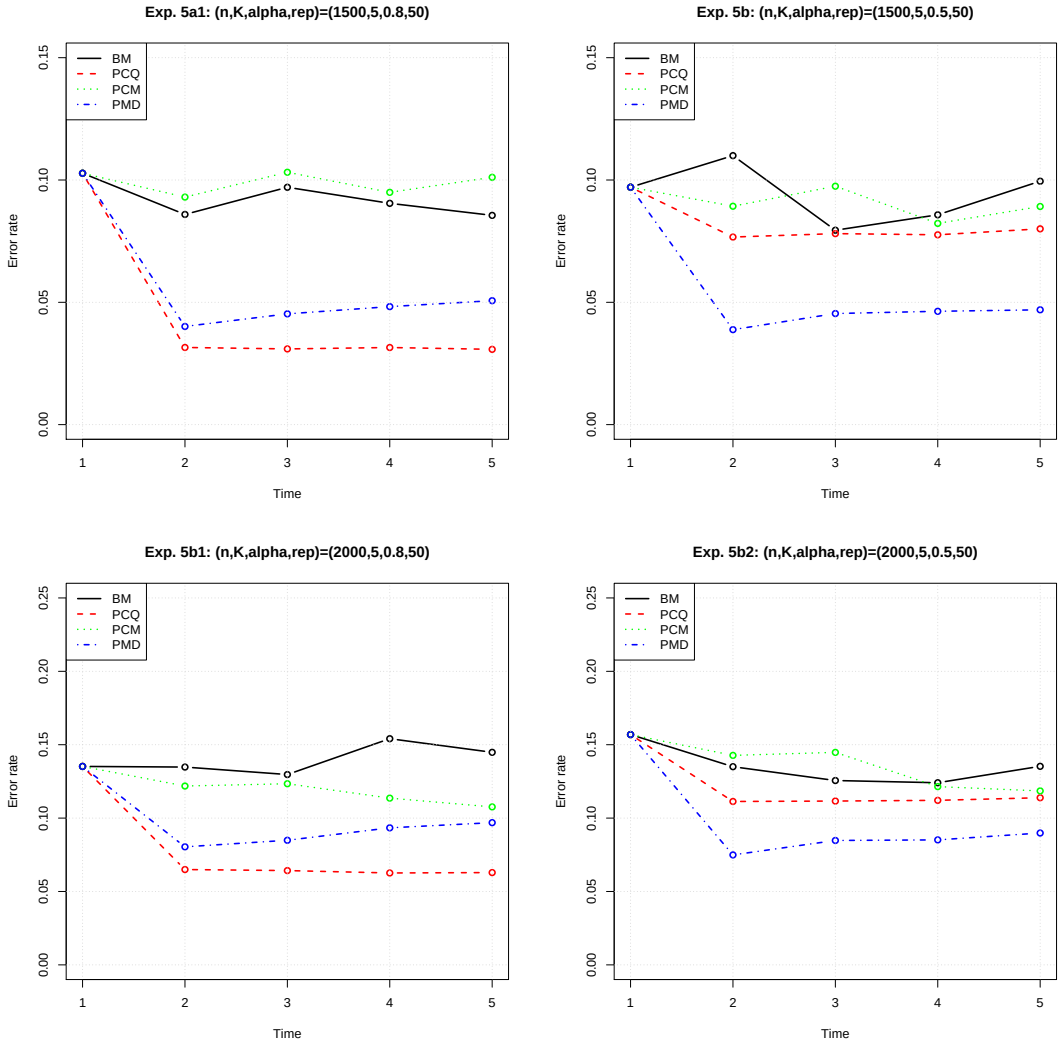


Figure 4.5 Experiment 5(a1,a2) (top) and 5(b1,b2) (bottom) Error Rate

In summary, PCM framework should not be considered to use in reality considering that fact that it performs the worst in five simulated experiments. PCQ framework is the best when the observed adjacency matrices are all coming from a simple network settings, e.g. low degree heterogeneity and low community evolutionary effect. PMD framework is more suitable for complicated cases. Considering that we are not able to know what type of network setting we are facing in the real network data, PMD is always a safe and good choice.

## 4.4 Data Application: Enron Email Data

Enron was an American energy, commodities, and services company based in Houston. As one of the most infamous corporate, it bankrupted due to a financial scandal in December 2001. Emails among management team, including CEO, president, vice president, etc, have been collected from January 1998 to June 2002. We generate three undirected and unweighted network adjacency matrices using the Enron email records in 2000, 2001, and 2002 [15]. We name them pre-scandal, in-scandal, and post-scandal period. In total, 74 users are included in three networks. We are curious to see if some interesting patterns from the results of our dynamic network community discovery algorithm can be found.

First of all, we need to determine the number of communities. Similar to the last chapter, we plot eigenvalues of unnormalized graph Laplacian matrices corresponding to the adjacency matrices in 2000, 2001, and 2002 in Figure 4.6, which suggests two/one/two communities in 2000/2001/2002.

We use Preserving Membership Degree (PMD) framework with $\alpha = 0.9$. Clustering results are summarized in Figure 4.7. Node size is proportional to the observed node degree. Colors indicate the user position. Red/pink/yellow represents CEO/president and vice president/others including director, manager, trader, etc. We observe that (1) more email conversations are generated in in-scandal period

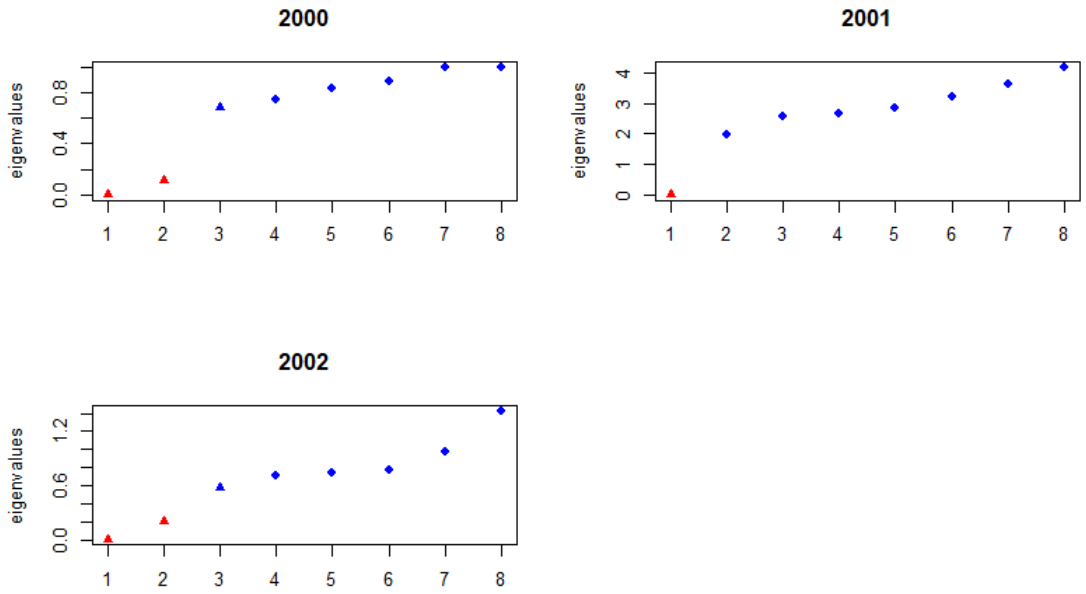


Figure 4.6 Enron Email Network Eigenvalue Plot

comparing to pre- and post-scandal period; (2) degrees of nodes are more similar in pre-scandal period, and high degree hubs emerge in in-scandal period; (3) two obvious communities are formed in post-scandal period, including the core management team and lower level management team.

We have the following hypotheses only based on our observation: (1) the entire management team generated more conversations in 2001 before the scandal to try to avoid the scandal happens; (2) core management team (e.g. CEO, president, vice president) should take main responsibility for the scandal since they are more involved in conversations with all parties when the scandal was happening; (3) separation into two communities in post-scandal period might be caused by the difference of group interests. More subject knowledge in business is required to conduct a deeper analysis regarding to how the management teams was manipulating the event in the pre-scandal and in-scandal periods, and what are possible group interests for two

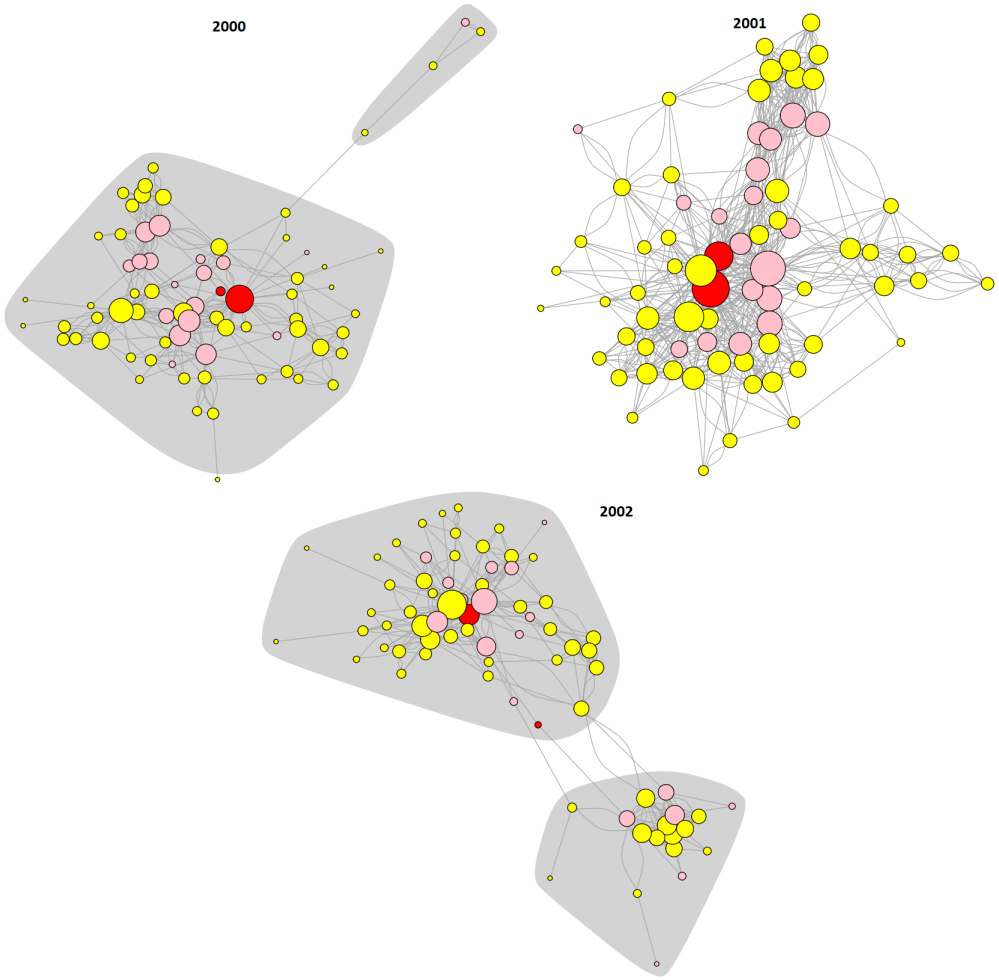


Figure 4.7 Enron Email Network Estimated Communities

communities. We leave it to readers who are interested in.

## 4.5 Discussions

### 4.5.1 More Historical Information

So far, we have only used historical information from time $t-1$ in the community discovery at time $t$. By doing so, both accuracy and stability of the clustering results are improved in most simulation experiments. It is natural to think whether involving more historical information would be helpful, and in which scale would the mean error

rate drop. Let's now consider the following total cost function:

$$\text{Cost}_t = \alpha(\text{CS}_t) + \beta(\text{CT}_t) + \gamma(\text{CT}_{t-1}), \tag{4.9}$$

where $t = 3, 4, \ldots, T$, and $\alpha + \beta + \gamma = 1$. Let $\alpha = 0.6$, and $\beta = \gamma = 0.2$, we run the simulation using the same setting as experiment 1(a) and 5(a) (except for the choice of $\alpha$) to generate the Figure 4.8. From top two plots, we can conclude that the mean error rate is significantly decreased after using $t-2$ information for PCQ framework, in which the network community labels do not change over time (simple setting). From the bottom two plots of Figure 4.8, we observe that the mean error rate for the PCQ/PMD framework is slightly increased/decreased. Both increase and decrease are not significant. Simulation results show that when the network is complicated enough, utilizing more historical information might have both positive and negative effects. It is beneficial to the nodes whose community labels are unchanged, and it also brings deceptive information to other nodes. Without knowing the true structure of the observed network, it is hard to determine the level of history to be borrowed. This topic needs further investigation in the future.

### 4.5.2 Time-varying $K$

It is more realistic to assume that number of communities changes over time. The authors who propose PCQ and PCM framework [5] pointed out that the dimension of the $W_t$ in PCQ and the distance function $\frac{1}{2}\|X_t X_t^T - X_{t-1} X_{t-1}^T\|_F^2$ in PCM do not depend on the value of $K$. Therefore, PCQ and PCM algorithms can be used directly without any modification when $K$ is varying in different time steps. The argument is true for PMD as well, because we measure the distance between the subspace spanned by the columns of $\tilde{\Theta}_t \tilde{X}_t$ and $\tilde{\Theta}_{t-1} \tilde{X}_{t-1}$, in which the effect of the number of columns, $K$, is ancillary.

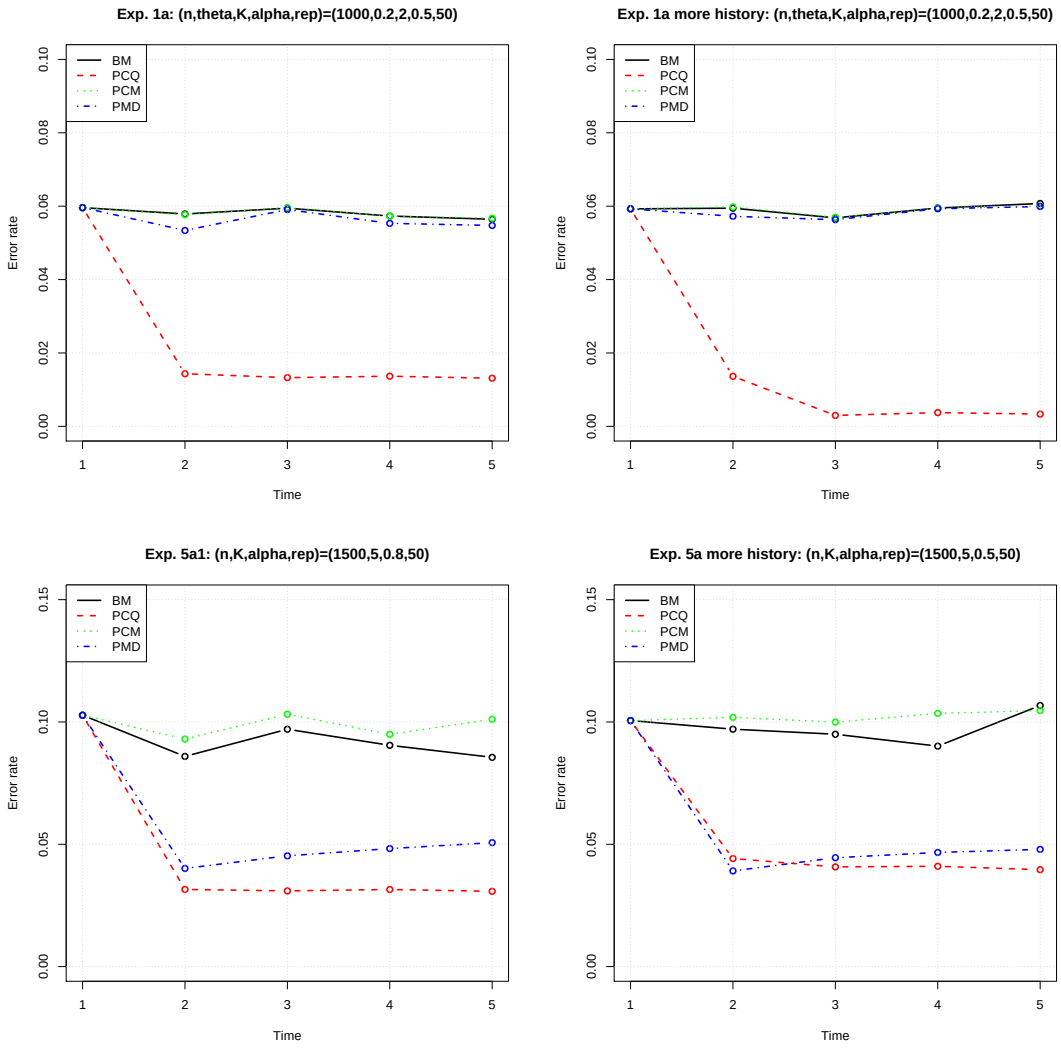


Figure 4.8 Experiment 1(a) (top) and 5(a) (bottom) with/without More Historical Info.

### 4.5.3 Time-varying $n$

Another impracticable assumption we made in previous sections is fixing the number of observed nodes over time. It is natural to have observations with different $n$ at two consecutive time step. Define $n_t$ and $n_{t-1}$ to be the number of observed nodes at time $t$ and $t-1$, respectively.

First, let's consider the case when $n_t = n_{t-1} - 1$, and assume it is the node $v_i$ at $t-1$ disappears at $t$. One way mentioned in [5] for PCQ framework is to remove the $i$th row/column from $W_{t-1}$ to match the dimension of $W_t$. In PCM or PMD framework, we can remove the $i$th row from $X_{t-1}$ or $\tilde{\Theta}_{t-1}\tilde{X}_{t-1}$, respectively, and then re-normalize the modified matrix. Similar modification can be conducted when more than one node disappears from at time $t$ and more rows/columns need to be removed.

When new nodes are added at time $t$ comparing to $t-1$, we need to find a way to increase the dimension of $W_{t-1}$, $X_{t-1}$ and $\tilde{\Theta}_{t-1}\tilde{X}_{t-1}$ in three frameworks, respectively. Let's assume $n_t = n_{t-1} + m_t$, and the first $n_{t-1}$th nodes at time $t$ are the old ones at time $t-1$. In PCQ framework, the modification formula [5] is

$$\text{modified } W_{t-1} = \begin{bmatrix} W_{t-1} & E_{t-1} \\ E_{t-1}^T & F_{t-1} \end{bmatrix},$$

where $E_{t-1} = \frac{1}{n_{t-1}} W_{t-1}\mathbf{1}_{n_{t-1}}\mathbf{1}_{m_t}^T$ and $F_{t-1} = \frac{1}{n_{t-1}^2}\mathbf{1}_{n_{t-1}}^T W_{t-1}\mathbf{1}_{n_{t-1}}\mathbf{1}_{m_t}\mathbf{1}_{m_t}^T$. The modified $W_{t-1}$ has three good characteristics: (1) It is positive semi-definite if $W_{t-1}$ is; (2) the relationship between each newly added node and an existing node is the same as the average relationship between two existing nodes; (3) the relationship between two newly added nodes is the same as the average relationship between two existing nodes. In PCM framework, the modification formula is

$$\text{modified } X_{t-1} = \begin{bmatrix} X_{t-1} \\ G_{t-1} \end{bmatrix},$$

where $G_{t-1} = \frac{1}{n_{t-1}}\mathbf{1}_{m_t}\mathbf{1}_{n_{t-1}}^T X_{t-1}$. By doing so, the probability of a newly added node belonging to the $k$th community is proportional to the estimated size of the $k$th

community. Re-normalization is needed after the modification. In PMD framework, we propose a similar modification formula

$$\text{modified } \tilde{\Theta}_{t-1}\tilde{X}_{t-1} = \begin{bmatrix} \hat{\Theta}_{t-1}\tilde{X}_{t-1} \\ H_{t-1} \end{bmatrix},$$

where $H_{t-1} = \frac{1}{n_{t-1}}\mathbf{1}_{m_t}\mathbf{1}_{n_{t-1}}^T\hat{\Theta}_{t-1}\tilde{X}_{t-1}$, and $\hat{\Theta}_{t-1} = \text{diag}\,(A_{t-1}\mathbf{1}_n)$. The probability of a newly added node belonging to the $k$th community is proportional to the total number of edges connecting to the estimated $k$th community, which can be understood as the estimated degree of the $k$th community. modified $\tilde{\Theta}_{t-1}\tilde{X}_{t-1}$ need to be re-normalized after the modification as well.

# Chapter 5

# Conclusion and Extension

So far we have discussed two algorithms to reinforce the ability of discovering network community structure based on spectral clustering type of method by borrowing extra information.

In Chapter 3, we assume both the adjacency matrix $A$ and edge covariates $\mathcal{X}$ are observed, and we propose an algorithm based on $A^*$, a combination of $A$ and $\mathcal{X}$, to have better community estimation. The intuition is that by mapping nodes into a latent space, the probability of forming an edge between any pair of nodes can be fully determined by a linear combination of the distance between two nodes in the latent space and edge covariates. We can simplify the objective function by applying Taylor series expansion, so that the original complicated problem can be reformulated to a fast-to-solve trace maximization problem. There is no doubt that the algorithm can be generalized. For example, we can assume adjacency matrix $A$ measures edge strength. Instead of `logit`, we use `log` link function to mimic the analysis procedure in previous chapter. The parameterization is

$$\eta_{ij} \equiv \log\{\mathrm{E}[A_{ij}|w_i, w_j, x_{ij}; \theta]\} = \alpha + x_{ij}^T\beta - \gamma\delta_{ij},$$

and the log-likelihood is

$$\sum_{i<j}\{\eta_{ij}A_{ij} - e^{\eta_{ij}} - \log(A_{ij}!)\}.$$

The objective function with distance regularization is

$$g^*(\theta, \mathcal{W}) = -\text{log-likelihood} + \lambda\sum_{i<j}\delta$$

$$= \sum_{i<j} \{e^{\eta_{ij}} + \log(A_{ij}!) - \eta_{ij} A_{ij}\} + \lambda \sum_{i<j} \delta$$
$$= \sum_{i<j} \left\{ \log(A_{ij}!) + e^{\alpha + x_{ij}^T \beta - \gamma \delta_{ij}} - (\alpha + x_{ij}^T \beta) A_{ij} + (\lambda + \gamma A_{ij}) \delta_{ij} \right\},$$

which is very similar to the equation 3.1. We can apply first-order Taylor series expansion to the exponential term at $\delta_{ij} = 0$, and further simply the objective function (as a function of $\mathcal{W}$) as follows

$$\sum_{i<j} \left[ \lambda + \gamma \left( A_{ij} - e^{\alpha + x_{ij}^T \beta} \right) \right] \delta_{ij} = \text{tr} \left[ W^T (D^{**} - A^{**}) W \right],$$

where

$$A_{ij}^{**} = \lambda + \gamma \left( A_{ij} - e^{\alpha + x_{ij}^T \beta} \right).$$

It serves as the same function as equation 3.3. We observe that with different distribution assumption, the combination of the adjacency matrix and edge covariates varies; however, there always exists a way to transfer the original problem into a trace maximization form. Therefore, our approach has potential to generate a complete set of algorithms in terms of different distribution assumptions of $A$ (and link functions) to provide community discovery solutions to all type of network data. Besides, it can be generalized in other ways, including adding another regularization term to penalize the covariates, and/or modifying the linearity assumption to be nonlinear, etc.

In Chapter 4, we assume to observe a series of adjacency matrices from the same network. Historical information is borrowed to provide with more accurate community estimation for current network. More specifically, in the proposed Preserving Membership Degree (PMD) algorithm, both previous partition and the relative degree are taken into consideration to address the intuition that nodes with relatively high degree tend to stay in the same community from $t-1$ to $t$, comparing to marginal nodes who have fewer within community degree. The intuition seems to make sense in most social networks, but it is not necessary true in others. There is a need to invent a method to test the intuition itself.

Network analysis and community discovery is relatively new to statisticians. In this dissertation, we have only discussed two types of approaches to analyze static and dynamic network data. There exists a large territory of network problems that have been seldom touched, like how to improve the prediction of people's survival time using social network information, how to predict the next terrorist attack or political election results by analyzing the interaction of users in a social network, etc, which motivates us to continue studying and exploring.

# Bibliography


[1] Emmanuel Abbe. Community detection and stochastic block models: recent developments. *arXiv preprint arXiv:1703.10146*, 2017.

[2] Peter J Bickel and Aiyou Chen. A nonparametric view of network models and newman–girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106(50):21068–21073, 2009.

[3] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[4] Deepayan Chakrabarti, Ravi Kumar, and Andrew Tomkins. Evolutionary clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 554–560. ACM, 2006.

[5] Yun Chi, Xiaodan Song, Dengyong Zhou, Koji Hino, and Belle L Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 153–162. ACM, 2007.

[6] Anne Condon and Richard M Karp. Algorithms for graph partitioning on the planted partition model. *Random Structures and Algorithms*, 18(2):116–140, 2001.

[7] Paul Erdos and Alfréd Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

[8] Edgar N Gilbert. Random graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144, 1959.

[9] Anna Goldenberg, Alice X Zheng, Stephen E Fienberg, Edoardo M Airoldi, et al. A survey of statistical network models. *Foundations and Trends® in Machine Learning*, 2(2):129–233, 2010.

[10] Gene H Golub and Charles F Van Loan. Matrix computations. 1996. *Johns Hopkins University, Press, Baltimore, MD, USA*, pages 374–426, 1996.

[11] Lars Hagen and Andrew B Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE transactions on computer-aided design of integrated circuits and systems*, 11(9):1074–1085, 1992.

[12] Mark S Handcock, Adrian E Raftery, and Jeremy M Tantrum. Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 170(2):301–354, 2007.

[13] Peter D Hoff, Adrian E Raftery, and Mark S Handcock. Latent space approaches to social network analysis. *Journal of the american Statistical association*, 97(460):1090–1098, 2002.

[14] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

[15] SRI International. Calo (cognitive assistant that learns and organizes). `http://www.ai.sri.com/project/CALO`, 2004.

[16] Jiashun Jin et al. Fast community detection by score. *The Annals of Statistics*, 43(1):57–89, 2015.

[17] Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

[18] Eric D Kolaczyk and Gábor Csárdi. *Statistical analysis of network data with R*, volume 65. Springer, 2014.

[19] Emmanuel Lazega et al. *The collegial phenomenon: The social mechanisms of cooperation among peers in a corporate law partnership*. Oxford University Press on Demand, 2001.

[20] Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th international conference on World Wide Web*, pages 695–704. ACM, 2008.

[21] Helmut Lutkepohl. Handbook of matrices. *Computational Statistics and Data Analysis*, 2(25):243, 1997.

[22] Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006.

[23] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[24] Karl Rohe, Sourav Chatterjee, Bin Yu, et al. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, 2011.

[25] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[26] Tom AB Snijders and Krzysztof Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of classification*, 14(1):75–100, 1997.

[27] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[28] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[29] Ulrike Von Luxburg, Mikhail Belkin, and Olivier Bousquet. Consistency of spectral clustering. *The Annals of Statistics*, pages 555–586, 2008.

[30] Hongyuan Zha, Xiaofeng He, Chris Ding, Ming Gu, and Horst D Simon. Spectral relaxation for k-means clustering. In *Advances in neural information processing systems*, pages 1057–1064, 2002.