UNIVERSITY OF MINNESOTA

This is to certify that I have examined
this copy of a doctoral dissertation by

Lin Zhang

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Yuhong Yang

Name of Faculty Adviser

Signature of Faculty Adviser

Date

GRADUATE SCHOOL

Consistent Cross Validation for Community Detection

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Lin Zhang

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Yuhong Yang, Adviser

December 2019

Acknowledgements

I am delighted to have undertaken my PhD in statistics at the beginning of a new era when the traditional art of statistical inference was merging with the modern art of computing. At the University of Minnesota, I have been inspired to rethink, redefine, and reassess the concept of data: what does data mean to us, how can/does data change our life, and how much theoretical understanding do we need to develop for those computing-based data techniques beyond the empirical and practical level. This last inquiry requires introduction of scientific and mathematical disciplines to the art of computing so that we can begin to fully understand the benefits, limitations and true potential of machine learning.

These questions have driven me through my six-year graduate program and research. My doctoral adviser, Professor Yuhong Yang, encouraged me to investigate application of the classic cross-validation technique to the leading-edge network analyses. This started me on an exciting and enlightening journey, where I not only have had the opportunity to work on a challenging topic, but also to be rewarded by advancing my deeper appreciation of statistics and data science overall. This dissertation marks an extraordinary milestone on my journey, which I will delightedly continue into my data science career.

I am honored to have had many people supporting me on my graduate school journey. My foremost gratitude goes to my adviser Professor Yuhong Yang, who guided me through every part of my research and thesis development, always with profuse enthusiasm and encouragement. His insights and perspectives expanded my understanding of statistics and illuminated me to complete my dissertation. I am also very thankful to be part of the family of the School of Statistics at the University

of Minnesota, where I received tremendous assistance, support, and inspiration from the faulty, staff, and fellow students. I would also like to express my gratitude to my parents for their unwavering encouragement of my academic pursuits. And finally, I would like to give my deepest thanks to my husband, Fan Yang, who I met in the statistics department. Fan has been infinitely generous, supportive, and a great sounding board and mentor throughout my study and I am eternally grateful to him.

Dedication

This thesis is dedicated to my husband Fan Yang and to our beloved son David Alexander Yang.

Abstract

The stochastic block model (Holland et al., 1983) is one of the most popular models for analyzing networks with complicated community structures. While most research in this area focuses on finding robust and efficient algorithms to detect those communities, we are interested in determining how many communities in the network using the cross validation method. In this dissertation, we introduce two new cross validation methods for community detection using different network splitting strategies. We have explored the consistency property for the two new cross-validation methods for community detection. We prove that under some conditions on the network and on the clustering algorithm and with a proper choice of the splitting ratio, our cross validation methods can consistently choose the correct community number in probability. It is known that several prevailing clustering algorithms for network analyses meet these conditions and therefore our consistency conclusion is applicable to those algorithms.

In addition to pursue the theoretical property, we use simulations to show that the two new methods achieve a good success rate when the network contains a small to moderate number of communities. We found out that the success rate depends on two other factors: how sparse the network is and how imbalanced the community sizes are. Regardless of these factors, our new methods are shown to outperform the existing network cross validation method (Chen and Lei, 2018) in simulations under stochastic block model. Furthermore, we have applied our new methods to analyze two real-life networks: the international trade and the U.S. Congress network. We have obtained interesting results that can be easily interpreted from a practical standpoint.

Contents

List of Tables **vii**

List of Figures **viii**

1 Introduction **1**

2 Community Detection under Stochastic Block Model **7**

3 Existing Methods to Determine Community Number K **12**

3.1 Hypothesis Test Methods 12

3.2 Likelihood Methods 14

3.2.1 Likelihood Ratio Test 14

3.2.2 Composite Likelihood BIC 16

3.3 Bayesian Methods 18

3.4 Existing Cross Validation Methods 20

4 Node Split Cross Validation (NS-CV) for Community Detection **23**

4.1 NS-CV Procedure 24

4.2 Consistency of NS-CV 28

4.2.1 Basic Notations and Assumptions 28

4.2.2 Supporting Lemmas 30

4.2.3 Main Consistency Theorem 46

4.3 Simulations 54
4.4 Applications 60
4.4.1 International Trade 60
4.4.2 The U.S. Senate Network 61

5 Edge Split Cross Validation (ES-CV) for Community Detection 68
5.1 ES-CV for Sparse Network 69
5.2 ES-CV for General Network 73
5.3 Consistency of ES-CV 75
5.3.1 Basic Notations and Assumptions 75
5.3.2 Supporting Lemmas 78
5.3.3 Main Consistency Theorem 83
5.4 Simulations 91
5.5 Applications 96
5.5.1 International Trade 96
5.5.2 The U.S. Senate Network 96

6 Conclusion and Future work 101

References 104

List of Tables

4.1 NS-CV Clustering of the U.S. Senate 66
4.2 NS-CV Summary for the U.S. Senate 67

5.1 ES-CV Clustering of the U.S. Senate 100
5.2 ES-CV Summary for the U.S. Senate 100

List of Figures

1.1 Graphical Representation of Network 2

4.1 Simulation: NS-CV Success Rate by Network Size 56
4.2 Simulation: NS-CV Success Rate by Network Sparsity and Imbalance 58
4.3 Simulation: NS-CV versus Chen and Lei's Network Cross Validation . 59
4.4 Application: NS-CV for International Trade Network 62
4.5 Application: NS-CV for the U.S. Senate Network 65

5.1 Simulation: ES-CV Success Rate by Network Size 92
5.2 Simulation: ES-CV Success Rate by Network Sparsity and Imbalance 94
5.3 Simulation: ES-CV vs. NS-CV vs. Chen & Lei's CV 95
5.4 Application: ES-CV for International Trade Network 97
5.5 Application: ES-CV for the U.S. Senate Network 99

Chapter 1

Introduction

Studies of networks have attracted much attention in recent years among computer scientists, social scientists, biologists, psychologists, statisticians and others. In a network analysis, what is of interest is the community structure within the network. It is known that the population of a network is not always uniform, but may consist of small subgroups, referred to as communities. One example is a social network such as Facebook, where participants tend to cluster around several small cliques based on their age, profession, education, common interests, or other demographic or social factors. Another example is a political assembly such as the U.S. Congress, where members who share similar political views group into caucuses. For the sake of simplicity, we assume in this dissertation that the communities (i.e. cliques, caucuses, etc.) are mutually exclusive in the sense that a member can belong to one and only one community. A central problem in a network analysis is to determine the community structure, i.e., who belongs to which community. This process is also known as community detection.

The community structure can be best learned from knowing how its members connect with others from the same community and from different communities. Here, connection can take various forms in the network such as a befriend or unfriend action on social media, or a political endorsement or a legislation support to other Congress

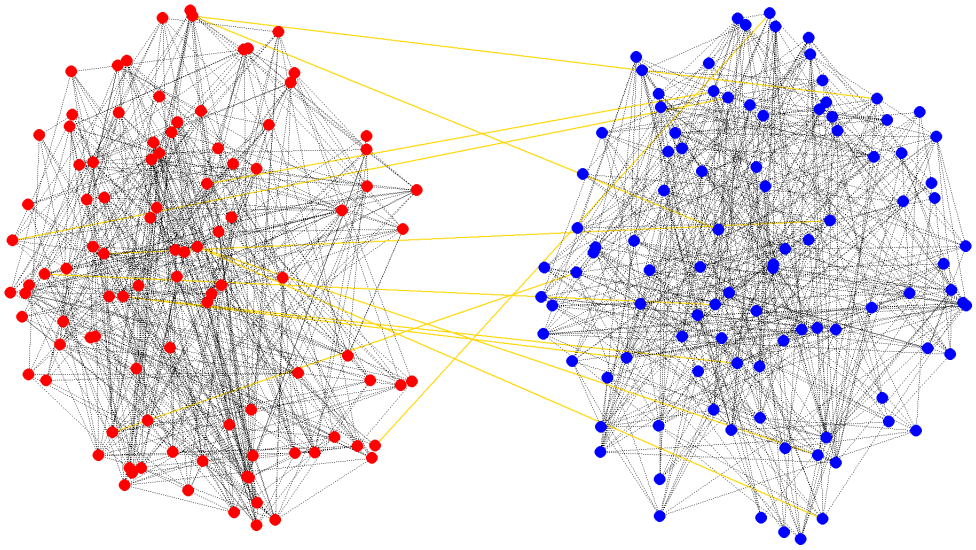


Figure 1.1: Graphical representation of a two-community network with dense within-community connections and sparse between-community connections.

members. To simplify the mathematical model, we assume connections are undirected in this dissertation, meaning that we do not distinguish between who initializes a connection and who receives it. Intuitively, in a network, members belonging to a same community are more likely to be connected with each other than with members from a different community. Thus, a network tends to have dense connections within communities and sparse connections between communities. Another useful way to visualize a network is to use a graph. Members of a network can be represented by a set of nodes in the graph and their connections are represented by the undirected edges between the nodes. Visually, the graph of a network should have a high edge density within communities, and a low density between them. Figure 1.1 illustrates such a typical network with dense within-community edges and spare between-communities edges.

Given a dataset that describes how the members (nodes) of a network connect (edges), a primary task of community detection is to cluster the members into com-

munities. Although this task is quite challenging both theoretically and computationally on a very complex network with many nodes and edges, the result of community detection can bring enormous insights to our understanding of the network and show values in many different applications. These include boosting the targeted advertising effect (Fang, 2014), improving the recommendation system (Li and Chen, 2011), better segmenting or classifying images (Papadopoulos et al., 2010), exploring protein-protein interactions (Airoldi et al., 2006), finding genetically related sub-populations (Greenbaum et al., 2016), or understanding the folding structure of DNA (Cabreros et al., 2016).

Over the past decades, there has been a considerable amount of interdisciplinary research conducted to develop good and efficient algorithms for community detection. Some of the notable works includes minimizing ratio cut (Wei and Cheng, 1989), minimizing normalized cut (Shi and Malik, 2000), maximizing modularity (Newman and Girvan, 2004), hierarchical clustering (Newman, 2004), and edge-removal methods (Newman and Girvan, 2004). Among these research, the spectral clustering method (Donath and Hoffman, 1973) has attracted much attention due to its high computational efficiency and consistent clustering performance (Von Luxburg et al., 2008; Lei et al., 2015). We refer interested readers to Spielmat and Teng (1996) for a comprehensive review on spectral clustering.

In statistics and machine learning research, one of the best-known probabilistic models for community detection is the stochastic block model (SBM), first introduced by Holland et al. (1983). SBM was originally developed for modeling a random graph with a community structure. Under SBM, the random graph of interest consists of nodes from distinct communities and has independent edges following a Bernoulli distribution. The probability for a within-community edge is assumed to be greater than the probability for a between-community edge. The entire model can therefore be fully characterized by an assignment function ϕ that maps each node to its be-

longing community and a probability matrix P that gives the probabilities of edges between and within every communities. A more rigorous discussion of SBM will be given in Section 2. Note that SBM is only a simplified probabilistic model that may not account for the high-level complexity observed in a real-life network. Therefore, assessment of performance for any community detection methods under SBM should only be limited to those representative networks.

There are three general types of community detection methods under SBM: one is based on the greedy algorithm (see e.g., Clauset et al., 2004), one uses spectral clustering (see, e.g., McSherry, 2001; Von Luxburg, 2007), and the third type is based on the maximum likelihood method (see, e.g., Karrer and Newman, 2011; Bickel et al., 2013). While a large variety of community detection methods of these three types have been developed in the past decades, most of these methods rely on a crucial assumption that the number of communities in the network, denoted by K, is known. For some networks, there is a natural choice for K from the community knowledge, for example, $K = 3$ for the political parties in the U.S. Congress, namely Democratic, Republican and Independent. However, in many other cases, either there is no presumable K or even with a default choice of K, the research may want to go beyond it to explore sub-communities. In those cases, K is an unknown parameter for the network.

Unfortunately, besides guessing or some uses of graphical diagnosis, there are few methods developed to help determine how many communities are in a network. For instance, Sarkar (2016) proposed a hypothesis test for $K = 1$ vs. $K > 1$ at each step of a recursive bipartition algorithm. Yet, this hypothesis test is very restrictive since for real application, one needs to use the test multiple times to determine the community number K and the overall error rate is difficult to control. A more general hypothesis test for $K = k$ vs. $K > k$ remains an open problem because of the difficulty of approximating the null distribution.

From a modeling perspective, we may view the number of communities K as a model parameter so that we can use a model selection procedure to determine its optimal value. This is very similar to the problem of determining the number of components for a mixture model, where Daudin et al. (2008) had proposed a method based on Bayesian information criterion (BIC). Saldana et al. (2017) extended this method to the stochastic block model by proposing a composite likelihood BIC (CL-BIC) criteria to determine K. A fully Bayesian approach was developed by McDaid et al. (2013), where a Bayesian hierarchical model was built on SBM with K included as a parameter. The posterior distribution of K could be estimated using a MCMC algorithm.

Besides the Bayesian methods, another commonly used model selection technique is cross-validation. This drives us to explore the opportunity of using the cross-validation method to determine K. Previously, there have already been some existing efforts to apply the cross-validation strategy for community detection. The first notable attempt was made by Wang et al. (2011) who proposed a method called network cross-validation based on some general results from Dietterich (1998). In another attempt, Neville et al. (2012) constructed a network cross-validation method to get a smaller type-I error for a within-class network learning. Hoff (2008) introduced another cross validation method using a random node-wise splitting strategy under a Bayesian framework. Most recently, Chen and Lei (2018) developed a formal network cross validation approach using a block-wise node-pair splitting strategy. They also showed that their method won't underestimate K with a high probability.

In this dissertation, we propose two new cross-validation methods. One method, named as node split cross validation (or NS-CV), follows a different network splitting strategy with that in Chen and Lei (2018). Our method introduces an independent classification step to bridge the training set clustering to the test set, which follows the true idea of cross-validation in network. Our second method takes a brand new

strategy of partitioning the network data. Instead of splitting the node set as done by most of the existing methods, we randomly divide the edge set as represented by the entries in the adjacency matrix. For this reason, we call this method edge splitting cross validation (or ES-CV). We apply the matrix completion technique to impute the "missing" edges in the adjacency matrix that are assigned to the test set so that a regular clustering method (such as Spectral Clustering) can be successfully performed in the training step. On the theoretical side, we explore the consistency property for the two new cross-validation methods following the general framework established by Yang (2007). A set of sufficient conditions are derived under which we prove both methods will choose the true community number K with a high probability.

The rest of this dissertation is organized as follows. Chapter 2 gives a brief overview of stochastic block model and introduces some basic settings and notations for network community detection. Chapter 3 surveys some existing methods to determine the community number K, including the use of hypothesis test, the likelihood method, and the Bayesian method as mentioned above. Chapter 4 and Chapter 5 are for the two new cross-validation methods respectively: node split cross validation (NS-CV) and edge split cross validation (ES-CV). For each method, we will present the formal algorithm, derive its consistency property, and show some simulation results about the performance. We will also include applications of the new cross validation methods to analyze two real-life networks: the international trade network and the United States Senate network. Finally, in Chapter 6, we will conclude this thesis with remarks about some open problems for future research.

Chapter 2

Community Detection under Stochastic Block Model

The Stochastic block model (SBM) is one of the simplest but also most useful models to characterize a random network where the probability of a connection between any two members only depends on whether they belong to the same community or not. Stochastic block model has a simple graphical representation. Let $G = (\mathcal{N}, \mathcal{E})$ be a random graph where $\mathcal{N}$ is the set of n nodes, representing the n members of the network and $\mathcal{E}$ is the set of random edges in the graph. Throughout this dissertation, we assume the edges are undirected, which implies that a connection (such as friendship, marriage) is mutually held between two members. An extension of the stochastic block model to a directed graph can be found in Wang and Wong (1987).

Without loss of generality, we may write $\mathcal{N} = \{1, 2, \ldots, n\} = [n]$ and $\mathcal{E} = \{(i, j) : 1 \leq i < j \leq n\}$. We assume the network contains K disjointed communities, indexed by $\{1, 2, \ldots, K\}$ where $2 \leq K \leq K_{max}$ for some positive integer K_{max}. Let ϕ denote the community labeling function that maps every node into the community it belongs to, or to write $\phi(i) \in \{1, 2, \ldots, K\}$ for $i \in \mathcal{N}$. The graph edge set $\mathcal{E}$ can be represented by an $n \times n$ symmetric matrix $\mathbf{A} = \{A_{ij}\}_{i,j=1}^{n}$, known as the adjacency matrix. Its upper off-diagonal entries A_{ij} $(1 \leq i < j \leq n)$ take a value of either one to indicate

the presence of an edge between node i and node j, or zero for no edge between them. Because the edges are undirected, $A_{ij} = A_{ji}$. In the stochastic block model, a self-loop (an edge that goes from a node to itself) is not allowed. This indicates the diagonal entries of $\mathbf{A}$ should all be equal to zero.

Under the stochastic block model, the off-diagonal entries of $\mathbf{A}$ are independent Bernoulli random variables whose probability of one is dependent on which communities the two nodes belong to. These probabilities can be placed into a $K \times K$ symmetric matrix $\mathbf{P} = \{p_{uv}\}_{u,v=1}^{K}$, known as the connectivity matrix. The entry p_{uv} gives the probability of having an edge between two nodes, one in community u and the other in community v. Note that the diagonal entry p_{uu} is the probability of a within-community edge, while the off-diagonal entry p_{uv} $(u \neq v)$ is the probability of a between-community edge. Because the edges are undirected, $p_{uv} = p_{vu}$. Using the labeling function ϕ, we can write

$$\Pr(A_{ij} = 1) = 1 - \Pr(A_{ij} = 0) = p_{\phi(i)\phi(j)}. \tag{2.1}$$

Given an observed adjacency matrix $\mathbf{A}$ and a known community number K, the goal of community detection is to estimate the labeling function ϕ and the connectivity matrix $\mathbf{P}$. Naturally, this can be done by the maximum likelihood method:

$$\max_{\phi,\ \mathbf{P}} \sum_{1 \leq i < j \leq n} \left[A_{ij} \log\left(p_{\phi(i)\phi(j)}\right) + (1 - A_{ij}) \log\left(1 - p_{\phi(i)\phi(j)}\right) \right]. \tag{2.2}$$

However, it is known that finding an exact solution to the optimization problem (2.2) is NP-hard (Chen et al., 2014). In practice, (2.2) may be solved using some heuristic methods such as the Label Switching Algorithm (Bickel and Chen, 2009) or the Tabu Search Algorithm (Zhao et al., 2012). An approximated solution of (2.2) may also be obtained using Gibbs Sampling (Snijders and Nowicki, 1997).

Besides the maximum likelihood method, a more practical method for community detection is to use Spectral Clustering. This method is computationally more efficient, because it is based on two well-developed algorithms: the spectral decomposition of a symmetric matrix and the k-mean clustering program. Spectral Clustering starts with a construction of the Graph Laplacian $\mathbf{L}$ for the adjacency matrix $\mathbf{A}$ defined as

$$\mathbf{L} := \mathbf{D} - \mathbf{A}, \tag{2.3}$$

where the *degree matrix* $\mathbf{D}$ is a diagonal matrix whose entry D_{ii} is the number of nodes that are connected with node i, i.e.,

$$D_{ii} := \sum_{j=1}^{n} A_{ij}. \tag{2.4}$$

The graph Laplacian $\mathbf{L}$ defined in (2.3) is commonly referred to as the Unnormalized Laplacian Matrix. According to Von Luxburg (2007), $\mathbf{L}$ is a symmetric and positive semi-definite matrix whose smallest eigenvalue equals zero. When the network is known to have K communities, the community detection by Spectral Clustering can be completed in two steps:

1. Compute the first K eigenvectors $u_1, u_2, \ldots, u_k$ of $\mathbf{L}$ that correspond to the K smallest eigenvalues in an ascending order (from the smallest to the largest);

2. Perform the K-mean algorithm on $U = (u_1, u_2, \ldots, u_k) \in \mathbb{R}^{n \times K}$ to partition the n rows into K clusters.

Spectral Clustering may also use a normalized Graph Laplacian (Shi and Malik, 2000; Ng et al., 2002), defined as

$$\mathbf{L}_{rw} \;:=\; \mathbf{D}^{-1}\mathbf{L}, \text{ or}$$

$$\mathbf{L}_{sym} := \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}. \tag{2.5}$$

In this case, the K-mean algorithm will apply to the first K eigenvectors of $\mathbf{L}_{rw}$ or $\mathbf{L}_{sym}$. In addition, to use $\mathbf{L}_{sym}$, the rows of $U = (u_1, u_2, \ldots, u_k)$ as created in Step 2 need to be normalized to have norm 1 before applying the K-mean algorithm, so that the nodes with different degrees (i.e. D_{ii}) can have a comparable size in the eigenvectors. More detailed discussion about the difference between the unnormalized and normalized Spectral Clustering can be found in Von Luxburg (2007).

A third choice for Spectral Clustering is to use the Singular Value Decomposition of the adjacency matrix $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ (Drineas et al., 2004; Jia, 2013), where the K-mean algorithm is performed on the K left-singular vectors in $\mathbf{U}$ (or the K right-singular vectors in $\mathbf{V}$) that corresponds to the K largest singular values of $\mathbf{A}$. This version of the Spectral Clustering will be used in our edge split cross validation in Chapter 5.

Von Luxburg et al. (2008), Rohe et al. (2014), and Lei et al. (2015) have proved that when the connectivity matrix $\mathbf{P}$, the community size n_i, and the community number K meet certain requirements, the Spectral Clustering is a consistent method for community detection under the stochastic block model. This also confirms the use of Spectral Clustering as an appropriate training method in cross validation.

A powerful extension of the stochastic block model is to introduce an additional set of degree parameters $\{\theta_i\}$ for every members of the network. These parameters characterize the member's degree of popularity in the network. This extended model is known as the degree-corrected block model, or DCBM (Dasgupta et al., 2004; Karrer and Newman, 2011). DCBM accounts for different popularity among the members in the network so that a connection between two members will be dependent on not only their community affiliation, but also their degrees of popularity. Mathematically, this

can be expressed as

$$A_{ij} \sim Bernoulli\left(p_{\phi(i)\phi(j)}\theta_i\theta_j\right). \tag{2.6}$$

Several methods have been developed for community detection under DCBM, which includes modification of Spectral Clustering (Qin and Rohe, 2013; Lei et al., 2015; Gulikers et al., 2017), or the use of modularity or likelihood based algorithms (Karrer and Newman, 2011; Amini et al., 2013). One of the fastest community detection methods under DCBM is known as SCORE developed by Jin (2015). Jin has shown that the effect of the additional degree parameter θ_i can be largely ignored from the model, if we take an entry-wise ratio of each eigenvector to the first one (corresponding to the largest eigenvalue). After that, a simple K-mean algorithm applied to the modified eigenvector ratio matrix will complete clustering of the network. On the theoretical side, the consistency of community detection under DCBM was established by Zhao et al. (2012) and Gao et al. (2018).

This dissertation focuses on the stochastic block model. Yet it is expected that our cross validation methods can be extended to estimate the number of communities under the degree-corrected block model as well.

Chapter 3

Existing Methods to Determine Community Number K

Most community detection applications assume a known community number K for the network of interest. However, it can be a difficult task to make a reasonable and educational assumption on K for some complicated networks. In fact, the community number K itself may be a parameter of interest that needs to be learned from the data along with the community structure. This chapter surveys some existing methods to determine the community number K.

3.1 Hypothesis Test Methods

Zhao et al. (2011) and Bickel and Sarkar (2016) have recommended a use of hypothesis test to determine the community number K for a network. Instead of making a direct estimate, the use of hypothesis test follows a recursive process to identify the K value. The overall network will be first tested on whether it consists of a homogeneous community ($H_0 : K = 1$) or contains two or more distinctive communities ($H_1 : K > 1$). If the null hypothesis is rejected, then the members of the network will be partitioned into two communities using any existing community detection method. The same hypothesis test will then be applied to the two formed sub-communities to

assess the necessity of a further partition. This process will end if the null hypothesis $H_0 : K = 1$ is not rejected for all the sub-communities.

To conduct the hypothesis test $H_0 : K = 1$ vs. $H_1 : K > 1$, Bickel and Sarkar (2016) proposed a use of the largest eigenvalue of properly normalized adjacency matrix as the test statistic. This idea was derived from random matrix theory, because the adjacency matrix A under the null hypothesis falls under a class of random matrices known as Gaussian Orthogonal Ensemble (GOE). The spectral property including the empirical distribution of the eigenvalues and the limiting distribution for the largest/smallest eigenvalue of such random matrices has been well established (Wigner, 1958; Tracy and Widom, 1994; Soshnikov, 1999). Before using the random matrix theory, the adjacency matrix A needs to be normalized by its estimated expectation value as

$$\begin{aligned} \hat{P} &:= \widehat{E(A)} = \hat{p}\mathbf{e}^T\mathbf{e} - \hat{p}\mathbf{I}, \\ \tilde{A} &:= \frac{A - \hat{P}}{\sqrt{(n-1)\hat{p}(1-\hat{p})}}, \end{aligned} \tag{3.1}$$

where $\hat{p} = \sum_{i<j} A_{ij}/n(n-1)$ is the estimated edge probability. Bickel and Sarkar (2016) proved that under the null hypothesis, the largest eigenvalue λ_1 of $\tilde{A}$, after a proper shift and rescale, will converge in distribution to the Tracy-Widom law with index 1.

$$\theta := n^{2/3}\{\lambda_1(\tilde{A}) - 2\} \xrightarrow{d} TW_1. \tag{3.2}$$

For a large network size n, the P-Value for the test statistic θ can be calculated based on the Tracy-Widom distribution. For a network with a small n, Bickel and Sarkar (2016) proposed a small sample correction using a parametric Bootstrap for the limiting Tracy-Widom distribution. Their simulation study showed that after the

correction, the empirical distribution for the test statistic θ has a close match to the Tracy-Widom distribution.

Although the hypothesis test establishes a natural framework to identify the community number K, there is no guarantee in theory that the recursive process would exactly recover the specific partitions of communities (Bickel and Sarkar, 2016). In other words, after $H_0 : K = 1$ is rejected for the first round test, and the network nodes are partitioned into two disjoint subsets, there is no proof that each set would be a disjointed union of the sub-communities. Thus the recursive process of hypothesis test may give some erroneous sub-partitions and eventually lead to a wrong K value.

3.2 Likelihood Methods

The above hypothesis test methods to decide the community number are not involved with the likelihood function. We are introducing the likelihood based hypothesis test in this section, where the community number K is treated as an unknown parameter for the likelihood function of the stochastic block model. In this way, the value of K can be determined by placing a specific optimal criterion on the likelihood function. There are two significant contributions to this type of method.

3.2.1 Likelihood Ratio Test

Wang and Bickel (2017) have studied the behavior of the likelihood ratio statistic for the latent stochastic block model, where the labeling function ϕ is denoted by a latent variable $Z_i = \phi(i) \in \{1, 2, \ldots, K\}$ that is assumed to follow a multinomial distribution with parameters $\pi = \{\pi_1, \pi_2, \ldots, \pi_k\}$. The parameter space for such a latent stochastic block model is $\Theta_K = \{\theta = (\pi, P)\}$ where $P = \{p_{uv}\}_{k\times k}$ is the connectivity matrix that gives the probability of a connection between and within

communities:

$$Pr\left\{A_{ij}=1|(Z_i=u,Z_j=v)\right\}=p_{uv}.$$

Given an observed adjacency matrix A, the likelihood function for the latent stochastic block model has a form:

$$g(A;\theta)=\sum_{z\in[K]^n} f(z,A;\theta), \tag{3.3}$$

where

$$f(z,A;\theta)=\left(\prod_{i=1}^{n}\pi_{z_i}\right)\left(\prod_{i<j}p_{z_iz_j}^{A_{ij}}(1-p_{z_iz_j})^{1-A_{ij}}\right). \tag{3.4}$$

For a mis-specified $K'\neq K$, the log likelihood ratio statistic is defined as:

$$L_{K,K'}=\log\frac{\sup\limits_{\theta\in\Theta_{K'}} g(A;\theta)}{\sup\limits_{\theta\in\Theta_K} g(A;\theta)}, \tag{3.5}$$

Wang and Bickel (2017) have shown that this log likelihood ratio statistic is asymptotically normal when K' is less than the true value K (i.e., the under-specification scenario). Although it is difficult to obtain an explicit form for the limiting distribution for $L_{K,K'}$ when K' is greater than its true value, they derived the order of convergence for this over-specification scenario. Based on these limiting distribution results for $L_{K,K'}$, they constructed a penalized likelihood criterion for selecting the optimal block number:

$$\beta(K')=\sup_{\theta\in\Theta_{K'}} g(A;\theta)-N_{K'}B_n, \tag{3.6}$$

where B_n is the order of the penalty term and $N_{K'}$ is a strictly increasing sequence indexed by K' to describe the complexity of the SBM. The optimal K^* is achieved by

$$K^* = \arg\max_{K'} \beta(K'). \tag{3.7}$$

This criterion is proved to be consistent in the sense that it will asymptotically chooses the correct K value in probability. In practice, because optimization of the likelihood function $g(A;\theta)$ using a traditional EM algorithm can be difficult. They recommend variational methods to approximately value the conditional distribution to simplify the local optimization at each iteration.

3.2.2 Composite Likelihood BIC

Motivated by the inherent relationship between community detection in the stochastic block model and clustering for a mixture model, Saldana and Feng (2017) proposed a composite likelihood BIC (CL-BIC) method to select the number of communities in a network. They identified that one of the difficulties for the use of exact likelihood methods for a stochastic block model is to deal with the possible conditional dependency among the edges A_{ij} within and between communities. In such a case, the composite likelihood method (Lindsay, 1988) can help ease the computation complexity and bring a similar inference for the estimation to the exact likelihood.

The general idea of composite likelihood is to multiply the conditional density functions of the components together, regardless of independent or dependent components. For the stochastic block model, given the adjacency matrix, A and the corresponding univariate density function $p_{ij}(A_{ij};\theta)$ for its entries, the composite

log-likelihood function has a form

$$cl(\theta; A) := \sum_{i<j} \log\Big(p_{ij}(A_{ij}; \theta)\Big). \tag{3.8}$$

Because every summand in $cl(\theta; A)$ is a valid log-likelihood object, the composite score estimating equation $\nabla_\theta cl(\theta; A) = 0$ is unbiased under regularity conditions. The associated composite likelihood estimator $\hat{\theta}_C$ is the solution of $\nabla_\theta cl(\theta; A) = 0$. The corresponding density estimator $\hat{p} = p_{\hat{\theta}_C}$ minimizes the expected composite Kullback-Leibler divergence. Based on this, the BIC criterion to select the community number K is

$$\text{CL-BIC}_K := -2cl(\hat{\theta}_C; A) + d_K^* \log(N(N-1)/2), \tag{3.9}$$

where K is the current model index, $d_K^* := Trace(H_K^{-1} V_K)$, $H_K := E_\theta(-\nabla_\theta^2 cl(\theta; A))$, and $V_K := Var(\nabla_\theta cl(\theta; A))$. Finally, the community number K is chosen as

$$\hat{K}_{CL-BIC} = \arg\min_{K \in \mathcal{K}} \left(\text{CL-BIC}_K\right), \tag{3.10}$$

where $\mathcal{K}$ is a set of the candidate values for K.

Saldana and Feng (2017) argued that the CL-BIC method puts a greater penalty for the increasing number of parameters than the conventional BIC method to pursue the true model for dependent relational data. However, despite its computational convenience, the consistency of this composition likelihood BIC method was not established.

3.3 Bayesian Methods

Several Bayesian methods have also been developed not only to recover the community structure, but also to answer the question about the number of communities (i.e. K) in a network. Some earliest works include Snijders and Nowicki (1997), who constructed a fully Bayesian framework for the stochastic block model. They also used a Gibbs Sampler to simulate a posterior sample for estimating the community structure. However, their method not only required a known K value, but also applies to the case of $K = 2$ only, which limits the usage of their method to a broader application.

McDaid et al. (2013) further extended Snijders and Nowicki's works by adding K as an unknown parameter to the Bayesian stochastic block model so that the estimate of community number K and community structure can be achieved simultaneously. Their hierarchical model starts with treating K as a random variable with a conditional Poisson prior distribution on $K > 0$:

$$\begin{aligned} K &\sim \text{Poisson}(\lambda)|K > 0 \\ Pr(K = k) &= \frac{\lambda^k e^{-\lambda}}{k!(1 - e^{-\lambda})}. \end{aligned} \tag{3.11}$$

With a given K, the next step is to represent the cluster membership as a random variable $z_i \in \{1, 2, \ldots, K\}$ following a multinomial distribution with a symmetric Dirichlet prior:

$$\begin{aligned} z_i|K, \theta &\sim \text{multinomial}(1; \theta_1, \theta_2, \ldots, \theta_K) \\ \theta &\sim \text{Dirichlet}(\alpha, \alpha, \ldots, \alpha), \end{aligned} \tag{3.12}$$

where θ_k gives the probability of a member belonging to the community k and $\sum_{k=1}^{K} \theta_k = 1$. Finally, with both z and K given, the connection (undirected edge

A_{ij}) between two members i and j follows a Bernoulli distribution whose parameter has a conjugated Beta prior:

$$\begin{aligned} A_{ij}|K,z,P &\sim \text{Bernoulli}(p_{z_i z_j}) \\ p_{uv} &\sim \text{Beta}(\beta_1,\beta_2). \end{aligned} \tag{3.13}$$

To improve the MCMC sampling, McDaid et al. (2013) further shrinks the parameter space by integrating out θ and P so that a marginal posterior $f(A,z,K)$ likelihood function has a simpler form as

$$\begin{aligned} L(z,K|A) &\propto \frac{\lambda^K}{K!}\frac{\Gamma(\alpha K)\prod_{k=1}^{k}\Gamma(n_k+\alpha)}{\Gamma(\alpha)^K\Gamma(N+\alpha K)}\prod_{1\le u<v\le K} f(x_{uv}|z) \\ f(x_{uv}|z) &= \frac{B(\beta_1+y_{kl}, n_k n_l - y_{kl}+\beta_2)}{B(\beta_1,\beta_2)}, \end{aligned} \tag{3.14}$$

where n_k is the number of members (nodes) in the community k and y_{kl} is the number of connections (edges) between the community k and l. Both are treated a function of the z value.

The resultant posterior distribution can be best learned using an MCMC algorithm that implements one of the following four types of proposals that give a potential move for (z,K):

- *MK*: A Metropolis move to increase or decrease K by 1, i.e., to add or delete an empty community. The algorithm will first select an existing community at random to remove an empty community. If that community is not empty, this proposal will be abandoned.

- *GS*: A Gibbs sampling moves to update the community assignment on one randomly selected member while keeping all other members assignment unchanged. For the elected member, the new task is achieved proportionally to

updated $L(z', K|A)$. Note that in this proposal, it is acceptable that the selected member is reassigned to its original community.

- *M3*: A Metropolis-Hasting update to reassign the members from two randomly selected communities as introduced by Nobile and Fearnside (2007). The reassignment may follow an arbitrary assignment or improve the choice of a member by considering the posterior probabilities derived from the previously reassigned members. Note that in this proposal the K value will not change.

- *AE*: An abort-eject move that first randomly selected a community and then either split it into two or merged it with another randomly selected community. If we would like to divide the community, we use a uniform probability p_E to decide whether a member is kept in the original community, or ejected to the new one. This update will change both z and K.

According to the simulation conducted in McDaid et al. (2013), the *MK* and *GS* moves are sufficient to have a good sampling coverage of the parameter space for (z, K), but at a low rate. Adding the *M3* and *AE* move will expedite the burn-in stage and make the MCMC less dependent on the initial choice of the parameter values. The simulation showed that this Bayesian method could achieve comparable results with regard to selecting the correct number of communities as an alternative method (Latouche et al., 2012). However, because this method relies on the MCMC simulation to explore the posterior distribution, it may require a substantial number of iterations to obtain a representative posterior sample.

3.4 Existing Cross Validation Methods

Cross validation is a robust model selection method in statistics. It is a technique to evaluate a predictive model by partitioning the data into a training set for model

fitting and a test set for model validation. It is known that cross validation helps prevent over-fitting because it checks the model using a subset that is not part of the fitting process. In practice, cross validation is a handy model selection tool to determine the optimal model parameters such as the regularization parameter in the Ridge or Lasso regression. Because of these benefits, we believe that cross validation may also be an excellent method for the network data analysis. Before introducing our new cross validation method, we give a survey of some existing cross validation methods to determine the community number K for community detection.

Neville et al. (2012) had proposed a network cross validation method that is used together with a Paired-T test to achieve an acceptable Type I error and statistical power for the network analysis. However, the computational burden of this method is considerably high and there was no theoretical justification for its performance. Most recently, Chen and Lei (2018) pointed out that Neville's method excludes a large number of valuable information in the cross validation process and thus can be much improved by a new training and test set splitting strategy that uses all connection information in the adjacency matrix. Under Chen and Lei's cross validation method, the node set $\mathcal{N}$ is randomly partitioned into a training set $\mathcal{N}^{(1)}$ and a test set $\mathcal{N}^{(2)}$. The adjacency matrix A is broken into four blocks accordingly as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^{(11)} & \mathbf{A}^{(12)} \\ \mathbf{A}^{(21)} & \mathbf{A}^{(22)} \end{bmatrix},$$

where $\mathbf{A}^{(jj)}$ is the adjacency matrix for nodes in $\mathcal{N}^{(j)}, j \in \{1, 2\}$ and $\mathbf{A}^{(12)}$, $\mathbf{A}^{(21)}$ are the adjacency matrices between the nodes in the training set $\mathcal{N}^{(1)}$ and the test set $\mathcal{N}^{(2)}$. For every candidate K value, the network is clustered to K communities based on the rectangular matrix block $(\mathbf{A}^{(11)}, \mathbf{A}^{(12)})$ which covers all nodes in the network. Validation of the clustering at each candidate K value is conducted using the matrix block $\mathbf{A}^{(22)}$, which is independent of $(\mathbf{A}^{(11)}, \mathbf{A}^{(12)})$. Chen and Lei (2018) has showed

that their cross validation method will not underestimate the community number K with a high probability. However it is not clear how their method will prevent an overestimate of K. Moreover, we also suspect that running community detection on $(\mathbf{A}^{(11)}, \mathbf{A}^{(12)})$ and estimating the connection probability density at the same time, while exploiting the information in $\mathbf{A}^{(12)}$, may also lead to a bias in estimation that does not hurt much in terms of ruling out smaller K but may hinder the differentiation between the true K and larger values. It is for this reason that we propose two new cross validation methods in this dissertation and we will compare their performances against Chen and Lei's method through simulations.

Finally, it is also worth to mention that Dabbs and Junker (2016) have recently improved Chen and Lei's cross validation method by utilizing a variational EM algorithm that help improve estimation of the network connection probability parameters. However, their work focuses more on the algorithm performance rather than providing a theoretical justification.

Chapter 4

Node Split Cross Validation (NS-CV) for Community Detection

In general, the cross validation procedure will take three steps to determine the community number K for community detection. Step 1 is to randomly partition the network into a training set and a test set. Step 2 is to perform community detection with candidate K values on the training set; Step 3 is to evaluate these candidate K values using the log likelihood or the mean square error (MSE) loss calculated on the test set. Step 1 to Step 3 may be repeated several times through independent samplings or by the use of an F-fold cross validation strategy. The K value that achieves the highest log likelihood or the lowest MSE loss will be the chosen community number for the network.

The biggest challenge arises in Step 2 where a full recovery of the community structure needs to be made on the training set only. In order to use a regular community detection method such as Spectral Clustering in Step 2, we develop two different cross validation strategies, which leads to the two new cross validation methods in this dissertation. This chapter is about the first method.

4.1 NS-CV Procedure

Node Split Cross Validation starts with a random split of the node set $\mathcal{N}$ into a training set $\mathcal{N}^{(1)}$ and a test set $\mathcal{N}^{(2)}$. We denote the splitting ratio as $\tau = |\mathcal{N}^{(1)}|/|\mathcal{N}| \in (0, 1)$. The adjacency matrix $\mathbf{A}$ is then partitioned accordingly into four matrix blocks under a proper re-arrangement of the rows and columns:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^{(11)} & \mathbf{A}^{(12)} \\ \mathbf{A}^{(21)} & \mathbf{A}^{(22)} \end{bmatrix}. \tag{4.1}$$

The two matrix blocks $\mathbf{A}^{(11)}$ and $\mathbf{A}^{(22)}$ contain the connections within the training set $\mathcal{N}^{(1)}$ and within the test set $\mathcal{N}^{(2)}$ respectively. $\mathbf{A}^{(12)} = [\mathbf{A}^{(21)}]^T$ is an $|\mathcal{N}^{(1)}| \times |\mathcal{N}^{(2)}|$ matrix block where the connections between the training and test set are stored. These three matrix blocks are mutually independent and will serve different roles in cross validation:

- Clustering: $\mathbf{A}^{(11)}$ is used to cluster $\mathcal{N}^{(1)}$ into K communities;
- Classification: $\mathbf{A}^{(12)}$ is used to classify nodes in $\mathcal{N}^{(2)}$ to the K communities;
- Validation: $\mathbf{A}^{(22)}$ is used to evaluate the clustered communities for $\mathcal{N}^{(2)}$.

In the Clustering step, given a candidate community number $K \in \{2, \ldots, K_{\max}\}$, we can use any standard community detection method (such as Spectral Clustering) on $\mathbf{A}^{(11)}$ to cluster $\mathcal{N}^{(1)}$ into K disjoint communities, denoted as $\mathcal{N}_1^{(1)}, \mathcal{N}_2^{(1)}, \ldots, \mathcal{N}_K^{(1)}$. This gives us a community labeling function ϕ_K on $\mathcal{N}^{(1)}$, i.e. $\phi_K(i) \in \{1, 2, \ldots, K\}$ for $i \in \mathcal{N}^{(1)}$.

Next in the Classification step, we can use the "One-vs-Other" classifier on $\mathbf{A}^{(12)}$ to classify every node in the test set $\mathcal{N}^{(2)}$ to one of those K communities clustered in

the previous step:

$$\phi_K(j) \;=\; \underset{1\le k\le K}{\arg\max} \left(\sum_{\substack{i\in\mathcal{N}^{(1)} \\ \phi_K(i)=k}} A_{ij}^{(12)}/|\mathcal{N}_k^{(1)}| \right), \qquad \text{for } j\in\mathcal{N}^{(2)}. \tag{4.2}$$

Heuristically this "One-vs-Other" classifier assigns a node in $\mathcal{N}^{(2)}$ to the community with which it has the highest degree of affinity measured by the density of connections in $\mathbf{A}^{(12)}$. Thereafter we obtain the community labeling function ϕ_K for all nodes in $\mathcal{N}$.

In the Validation Step, for every pair of communities indices $u, v \in \{1, 2, \ldots, K\}$, we define

$$E_{uv} \;:=\; \left\{(i,j):\; i,j\in\mathcal{N}^{(2)},\; i<j,\; \phi_K(i)=u,\; \phi_K(j)=v\right\}, \tag{4.3}$$

as the set of all nodes pairs in $\mathcal{N}^{(2)}$ with one node assigned to community u and the other to community v. Given the estimated labeling function ϕ_K, we write the log likelihood as a function of the parameters $\{p_{uv}\}_{u,v=1}^K$ for the connection probability between community u and v and we evaluate this log likelihood function on the test subset $\mathbf{A}^{(22)}$ as

$$\ell\left(\{p_{uv}\}_{u,v=1}^K \,\middle|\, \phi_K\right) = \sum_{u,v\in[K]} \sum_{(i,j)\in E_{uv}} \left[A_{ij}^{(22)} \log\left(p_{uv}\right) + \left(1-A_{ij}^{(22)}\right)\log\left(1-p_{uv}\right)\right]. \tag{4.4}$$

The maximum likelihood estimator for every p_{uv} is

$$\hat{p}_{uv} = \frac{\sum\limits_{(i,j)\in E_{uv}} A_{ij}^{(22)}}{|E_{uv}|}, \tag{4.5}$$

and the maximized log likelihood is

$$\ell\left(\{\hat{p}_{uv}\}_{u,v=1}^{K}\,\middle|\,\phi_K\right)=\sum_{u,v\in[K]}\sum_{(i,j)\in E_{uv}}\left[A_{ij}^{(22)}\log\left(\hat{p}_{uv}\right)+\left(1-A_{ij}^{(22)}\right)\log\left(1-\hat{p}_{uv}\right)\right]. \quad (4.6)$$

Finally, for a single cross validation split, we obtain the optimal community number among all candidate $K\in\{2,\ldots,K_{\max}\}$ as

$$K_{\text{optimal}}=\underset{2\le K\le K_{\max}}{\arg\max}\;\ell\left(\{\hat{p}_{uv}\}_{u,v=1}^{K}\,\middle|\,\phi_K\right). \quad (4.7)$$

In practice, the cross validation is commonly implemented through a F-fold strategy where the node set $\mathcal{N}$ is randomly partitioned into F nearly equal sized subsets at the beginning. The three steps described above are repeated F times. Each time, one subset is retained as the test set, and the remaining $F-1$ subsets are used as the training set. The candidate K that scores the highest average of $\ell\left(\{\hat{p}_{uv}\}_{u,v=1}^{K}\,\middle|\,\phi_K\right)$ across the F repeats will be the chosen community number. We present the formal algorithm for an F-fold Node Split Cross Validation on the next page.

Algorithm 1 Node Split Cross Validation (NS-CV)

Input: The adjacency matrix $\mathbf{A}$, $K_{\max} \geq 2$, $F \geq 2$.

Output: The optimal community number $K_{\text{NS-CV}}$.

1: Randomly partition the node set $\mathcal{N} = \{1, 2, \ldots, n\}$ into F nearly equal sized subsets, denoted as $\mathcal{N}^{(1)}, \mathcal{N}^{(2)}, \ldots, \mathcal{N}^{(F)}$.

2: **for** $f = 1$ to F **do**

3: Partition the adjacency matrix $\mathbf{A}$ into three matrix blocks:

$$\begin{aligned} \mathbf{A}^{(11)} &= (A_{ij})_{i,j \in \mathcal{N} \setminus \mathcal{N}^{(f)}} \\ \mathbf{A}^{(12)} &= (A_{ij})_{i \in \mathcal{N} \setminus \mathcal{N}^{(f)},\ j \in \mathcal{N}^{(f)}} \\ \mathbf{A}^{(22)} &= (A_{ij})_{i,j \in \mathcal{N}^{(f)}} \end{aligned}$$

4: **for** $K = 2$ to K_{max} **do**

5: Run Spectral Clustering on $\mathbf{A}^{(11)}$ to cluster $\mathcal{N} \setminus \mathcal{N}^{(f)}$ into K communities.

6: Follow (4.2) to classify $\mathcal{N}^{(f)}$ into the K communities.

7: Obtain the MLE for the connectivity probabilities $\hat{p}_{uv}$ by (4.5).

8: Compute the maximized log likelihood $\ell^{(f)}\left(\{\hat{p}_{uv}\}_{u,v=1}^{K} \,\middle|\, \phi_K\right)$ by (4.6).

9: **end for**

10: **end for**

11: **return** $K_{\text{NS-CV}} = \underset{2 \leq K \leq K_{\max}}{\arg\max} \sum_{f=1}^{F} \ell^{(f)}\left(\{\hat{p}_{uv}\}_{u,v=1}^{K} \,\middle|\, \phi_K\right) / F$.

Note that Algorithm 1 differs from the Blockwise Node Pair Splitting Network Cross Validation introduced by Chen and Lei (2018) in two aspects. First, Chen and Lei's method clusters all the nodes into K communities in a single step by running a singular value decomposition for the rectangular matrix block $\left(\mathbf{A}^{(11)}, \mathbf{A}^{(12)}\right)$, while our NS-CV method recovers the network's communities in two separate steps: Clustering on $\mathbf{A}^{(11)}$ and Classification on $\mathbf{A}^{(12)}$. Secondly, Chen and Lei's method estimates the connection probabilities $\{p_{uv}\}_{u,v=1}^{K}$ using the same entries of $\left(\mathbf{A}^{(11)}, \mathbf{A}^{(12)}\right)$ as used for the clustering. In contrast, our NS-CV method uses the maximum likelihood estimator on $\mathbf{A}^{(22)}$ whose entries can be considered independent given the community labeling function ϕ_K derived based on $\mathbf{A}^{(11)}$ and $\mathbf{A}^{(12)}$. This independence plays a crucial role in our proof of the NS-CV consistency.

4.2 Consistency of NS-CV

4.2.1 Basic Notations and Assumptions

In this section, we derive the consistency property for NS-CV under a few assumptions about the network and the community detection method. Consider an undirected network $(\mathcal{N}, \mathcal{E})$ generated by the Stochastic Block Model with K^* disjoint communities of sizes $\{n_k\}_{k=1}^{K^*}$ and a true community labeling function ϕ^*. Without further specification, we assume the true community number K^* is fixed and $2 \leq K^* \leq K_{\max}$ for some positive integer $K_{\max}$. Let $n = |\mathcal{N}| = \sum_{k=1}^{K^*} n_k$ be the total size of the network. For simplicity of derivation, we assume that the within-community connection probabilities are all equal to p and the between-community connection probabilities are all equal to q where $0 < q < p < 1$.

We run the Node Split Cross Validation for the network with one random partition of the node set $\mathcal{N}$ into the training set $\mathcal{N}^{(1)}$ and the test set $\mathcal{N}^{(2)}$ according to a

splitting ratio $\tau = |\mathcal{N}^{(1)}|/|\mathcal{N}| \in (0,1)$. For every candidate $K \in \{2,\ldots,K_{\max}\}$, we first apply a clustering method on $\mathbf{A}^{(11)}$ to cluster $\mathcal{N}^{(1)}$ into K communities. Then we classify $\mathcal{N}^{(2)}$ to these K communities according to (4.2). We denote the derived community labeling function as ϕ_K. In the Validation step of NS-CV, under the assumption of an equal within-community connection probability p and equal between-community connection probability q, we define

$$W_{\phi_K} := \{(i,j) : i,j \in \mathcal{N},\ i<j,\ \phi_K(i)=\phi_K(j)\}, \tag{4.8}$$

$$B_{\phi_K} := \{(i,j) : i,j \in \mathcal{N},\ i<j,\ \phi_K(i)\neq\phi_K(j)\}. \tag{4.9}$$

W_{ϕ_K} and B_{ϕ_K} divide the above-diagonal entries of the adjacency matrix $\mathbf{A}$ into two disjoint groups: the within group and the between group. Also write

$$W^{(2)}_{\phi_K} := \big\{(i,j) : i,j \in \mathcal{N}^{(2)},\ i<j,\ \phi_K(i)=\phi_K(j)\big\}, \tag{4.10}$$

$$B^{(2)}_{\phi_K} := \big\{(i,j) : i,j \in \mathcal{N}^{(2)},\ i<j,\ \phi_K(i)\neq\phi_K(j)\big\}, \tag{4.11}$$

be the two subsets of W_{ϕ_K} and B_{ϕ_K} corresponding to the entries in $\mathbf{A}^{(22)}$ only. Then the maximized log likelihood function given by (4.6) can be simplified as

$$\begin{aligned}\ell\left(\hat{p},\hat{q} \mid \phi_K\right) &= \sum_{(i,j)\in W^{(2)}_{\phi_K}} \Big(A^{(22)}_{ij}\log\hat{p} + (1-A^{(22)}_{ij})\log(1-\hat{p})\Big) \\ &\quad + \sum_{(i,j)\in B^{(2)}_{\phi_K}} \Big(A^{(22)}_{ij}\log\hat{q} + (1-A^{(22)}_{ij})\log(1-\hat{q})\Big),\end{aligned} \tag{4.12}$$

where

$$\hat{p} = \frac{\sum_{(i,j)\in W^{(2)}_{\phi_K}} A^{(22)}_{ij}}{|W^{(2)}_{\phi_K}|}, \quad \hat{q} = \frac{\sum_{(i,j)\in B^{(2)}_{\phi_K}} A^{(22)}_{ij}}{|B^{(2)}_{\phi_K}|}, \tag{4.13}$$

are the maximum likelihood estimators for p and q. The optimal K from this single split NS-CV is given by (4.7).

4.2.2 Supporting Lemmas

In this section, we introduce a number of supporting lemmas for the proof of the main consistency theorem for NS-CV. Because the partition of the training and test set in NS-CV is essentially a random sampling without replacement, we start with the following lemma that gives an exponential bound for the Hypergeometric distribution.

Lemma 4.2.1. *(Serfling, 1974) Let $S_n \sim$ Hypergeometric(n, D, N) that represents the sum of a random sample of size n without replacement from a whole set with D 1's and $N - D$ 0's. Then for any $t > 0$,*

$$Pr\left(S_n - n\frac{D}{N} \geq nt\right) \leq \exp\left(-\frac{2nt^2}{1-\frac{n-1}{N}}\right), \tag{4.14}$$

$$Pr\left(S_n - n\frac{D}{N} \leq -nt\right) \leq \exp\left(-\frac{2nt^2}{1-\frac{n-1}{N}}\right). \tag{4.15}$$

The proof of (4.14) was given in Serfling (1974). The inequality (4.15) is a direct result from applying (4.14) to $n - S_n \sim$ Hypergeometric$(n, N - D, N)$.

Our next two lemmas give a lower and upper bound for the cardinalities of W_{ϕ_K} and B_{ϕ_K} defined in (4.8) and (4.9).

Lemma 4.2.2. *For any network of size n with $K \geq 2$ communities given by a community labeling function ϕ_K, the cardinalities of W_{ϕ_K} and B_{ϕ_K} satisfy*

$$|W_{\phi_K}| \geq \frac{n^2 - nK}{2K}, \tag{4.16}$$

$$|B_{\phi_K}| \leq \frac{(K-1)n^2}{2K}, \tag{4.17}$$

where the equality is achieved when every communities have an equal size of n/K.

Proof. Let n_k denote the size of the k^{th} community in the network. Define $\alpha_k = n_k/n$ for $k = 1, 2, \ldots, K$. By Cauchy Inequality, $\sum_{k=1}^{K} \alpha_k^2 \geq 1/K$, where the equality sign is achieved when $\alpha_1 = \alpha_2 = \cdots = \alpha_K = 1/K$. Hence,

$$\begin{aligned} |W_{\phi_K}| &= \sum_{k=1}^{K} \binom{n_k}{2} = \frac{n^2}{2} \sum_{k=1}^{K} \alpha_k^2 - \frac{n}{2} \geq \frac{n^2 - nK}{2K}, \\ |B_{\phi_K}| &= \sum_{\substack{k,l=1 \\ k<l}}^{K} n_k n_l = \frac{n^2 - n}{2} - |W_{\phi}| \leq \frac{(K-1)n^2}{2K}, \end{aligned}$$

where the equality is achieved when every community has an equal size of n/K. □

The lemma above states the lower bound of $|W_{\phi_K}|$ has an order of n^2. This n^2 order does not apply to the lower bound of $|B_{\phi_K}|$ in a general case. An extreme example is when the first $K-1$ communities all have a size of 1 and remaining one has a size of $n - K + 1$, then $|B_{\phi_K}|$ has an order of n. However, if we restrict our discussion to those networks whose communities have comparable sizes, then the same n^2 order can be proved for the lower bound of $|B_{\phi_K}|$.

Definition 4.2.3. *(Balanced Community Structure) A network $(\mathcal{N}, \mathcal{E})$ with size n and $K \geq 2$ disjoint communities is claimed to have a balanced community structure, if its minimum community size $n_{min} = \min_{1 \leq k \leq K}(n_k) \geq n\pi_0/K$ for some positive constant $\pi_0 \in (0, 1)$ independent of n.*

Lemma 4.2.4. *For a network with a balanced community structure, the cardinalities of W_{ϕ_K} and B_{ϕ_K} satisfy*

$$|W_{\phi_K}| \leq \frac{n^2}{2}\left(1 - \frac{K-1}{K}\pi_0(2 - \pi_0)\right) - \frac{n}{2}, \tag{4.18}$$

$$|B_{\phi_K}| \geq \frac{n^2(K-1)\pi_0(2-\pi_0)}{2K}, \tag{4.19}$$

where the equality is achieved when there are $K-1$ communities with their sizes all

equal to the minimum size $n\pi_0/K$, and the remaining one community has a size of $n-(K-1)n\pi_0/K$.

Proof. Define $\alpha_k = n_k/n$ for $k = 1, 2, \ldots, K$. Then $\sum_{k=1}^{K} \alpha_k = 1$ and $\alpha_k \geq \pi_0/K$ by the balanced condition. It can be proved by contradiction that $\sum_{k=1}^{K} \alpha_k^2$ achieves its maximum when there are $K-1$ (α_k) equal to π_0/K, and the remaining one is equal to $1-(K-1)\pi_0/K$, i.e.

$$\sum_{k=1}^{K} \alpha_k^2 \leq (K-1)\left(\frac{\pi_0}{K}\right)^2 + \left(1-(K-1)\frac{\pi_0}{K}\right)^2.$$

This is because otherwise, without losing of generality, assume $\sum_{k=1}^{K} \alpha_k^2$ achieves its maximum when $\alpha_1 = \pi_0/K+\beta_1$ and $\alpha_2 = \pi_0/K+\beta_2$ for two strictly positive constants β_1 and β_2. Then

$$\begin{aligned}\sum_{k=1}^{K} \alpha_k^2 &= \left(\frac{\pi_0}{K}+\beta_1\right)^2 + \left(\frac{\pi_0}{K}+\beta_2\right)^2 + \sum_{k=3}^{K} \alpha_k^2 \\ &< \left(\frac{\pi_0}{K}+\beta_1+\beta_2\right)^2 + \left(\frac{\pi_0}{K}\right)^2 + \sum_{k=3}^{K} \alpha_k^2.\end{aligned}$$

This contradicts to the assumption that $\sum_{k=1}^{K} \alpha_k^2$ achieves the maximum value. Thus,

$$\begin{aligned}|W_{\phi_K}| &= \sum_{k=1}^{K} \binom{n_k}{2} = \frac{n^2}{2}\sum_{k=1}^{K} \alpha_k^2 - \frac{n}{2} \leq \frac{n^2}{2}\left(1-\frac{K-1}{K}\pi_0(2-\pi_0)\right) - \frac{n}{2}, \\ |B_{\phi_K}| &= \sum_{\substack{k,l=1 \\ k<l}}^{K} n_k n_l = \frac{n^2-n}{2} - |W_\phi| \geq \frac{n^2(K-1)\pi_0(2-\pi_0)}{2K},\end{aligned}$$

where the equality is achieved when there are $K-1$ communities with their sizes all equal to the minimum size $n\pi_0/K$, and the remaining one community has a size of $n-(K-1)n\pi_0/K$. □

Next, we present three required conditions for NS-CV consistency:

Condition 1. The true community structure (under ϕ^*) of a network and the recovered community structure (under ϕ_K) from NS-CV for $K \in \{2, \ldots, K_{\max}\}$ are balanced relative to a constant $\pi_0 \in (0, 1)$.

Condition 2. The within and between community probability p and q satisfy $0 < \delta < q < p < 1 - \delta < 1$ for a small some positive δ.

Condition 3. When the candidate community number $K = K^*$, the clustering method used in the Clustering step of NS-CV gives a consistent estimate ϕ_{K^*} from $\mathbf{A}^{(11)}$ so that on $\mathcal{N}^{(1)}$

$$Pr\left(\phi_{K^*} \neq \phi^*\right) \leq g(n\tau), \tag{4.20}$$

where n is the network size, τ is the node splitting ratio for the training subset, and $g(x)$ is a positive function such that $g(x) \to 0$ as $x \to \infty$.

Note that Condition 1 ensures every community has its size grow along with the overall network so that each has sufficiently large number of nodes and connections for community detection. This condition also excludes those networks that contain a very small community (e.g. a community with a single member). Condition 2 ensures that the between-community connections are not too sparse ($0 < \delta < q$) and the within-community connections are not too dense either ($p < 1 - \delta < 1$). Under such a condition, it is possible to fully recover the community structure from the clustering and estimation steps in NS-CV. Moreover, Condition 2 also requires the within-community density be higher than between-community density, which conforms to the common sense for community detection. Condition 3 speaks for the quality of the training method used in NS-CV. Regardless of the cross-validation strategy, the basic training method shall deliver a good community detection result

so that the cross-validation can evaluate the tuning parameter K in this application. It is known that under some technical conditions, Condition 3 is true for the Spectral Clustering method (Lei et al., 2015) used in Algorithm 1 of NS-CV.

Lemma 4.2.5. *Under Condition 1 and 2, the maximum likelihood estimate $\hat{p}$ and $\hat{q}$ as defined in (4.13) for any candidate $K \geq 2$ are bounded away from 0 and 1 in probability so that*

$$Pr\left(\frac{\delta}{2} < \hat{p} < 1 - \frac{\delta}{2}\right) \geq 1 - \exp(-C_p n^2 (1-\tau)^2), \tag{4.21}$$

$$Pr\left(\frac{\delta}{2} < \hat{q} < 1 - \frac{\delta}{2}\right) \geq 1 - \exp(-C_q n^2 (1-\tau)^2), \tag{4.22}$$

where τ is the NS-CV node splitting ratio, $\delta \in (0,1)$ is defined in Condition 2, C_p and C_q are two positive constants only dependent on K, π_0 (from Condition 1), and δ (from Condition 2).

Proof. First consider $\hat{q}$. Recall by (4.13),

$$\hat{q} = \frac{\sum_{(i,j)\in B^{(2)}_{\phi_K}} A^{(22)}_{ij}}{|B^{(2)}_{\phi_K}|}.$$

Because every $A^{(22)}_{ij}$ has an expected value equal to p or q, it is obvious by Condition 2 that

$$0 < \delta < q \leq E(\hat{q}) \leq p < 1 - \delta < 1. \tag{4.23}$$

Note that by Condition 1 the recovered community structure (under ϕ_K) is known balanced with respective to a constant $\pi_0 \in (0,1)$ on the training set $\mathcal{N}^{(1)}$. In the Classification step of NS-CV, we first classify the nodes in the test set $\mathcal{N}^{(2)}$ to the K communities according to (4.2). Let $n^{(2)}_k$ denote the number of nodes

in $\mathcal{N}^{(2)}$ being classified to the k^{th} community for $k = 1, 2, \ldots, K$. If $\min_{1\le k\le K} n_k^{(2)} \ge n(1-\tau)\pi_0/(2K)$, then the recovered network structure for $\mathcal{N}^{(2)}$ is balanced with respect to $\pi_0/2$. Otherwise, let $k_{\min}$ and $k_{\max}$ be the label of the community with the minimum and maximum size in $\mathcal{N}^{(2)}$ respectively. Also write $\mathcal{N}^{(2)}_{\min}$ and $\mathcal{N}^{(2)}_{\max}$ as the corresponding minimum and maximum communities for $\mathcal{N}^{(2)}$. We will re-classify the node j_0 from $\mathcal{N}^{(2)}_{\max}$ to $\mathcal{N}^{(2)}_{\min}$ that satisfies

$$j_0 = \underset{j\in\mathcal{N}^{(2)}_{\max}}{\arg\max} \left(\sum_{\substack{i\in\mathcal{N}^{(1)} \\ \phi_K(i)=k_{\min}}} A^{(12)}_{ij} \right). \tag{4.24}$$

This re-classification can be repeated until $\min_{1\le k\le K} n_k^{(2)} \ge n(1-\tau)\pi_0/(2K)$ and as a result the recovered network structure for $\mathcal{N}^{(2)}$ will be balanced with respect to $\pi_0/2$. Thus by applying Lemma 4.2.2 and Lemma 4.2.4 to the recovered network of $\mathcal{N}^{(2)}$, we have

$$\frac{n^2(1-\tau)^2(K-1)\pi_0(4-\pi_0)}{8K} \le |B^{(2)}_{\phi_K}| \;\le\; \frac{(K-1)n^2(1-\tau)^2}{2K}. \tag{4.25}$$

Next using the Law of Total Probability and Hoeffding Inequality (Hoeffding, 1963), we have

$$\begin{aligned}
& Pr\left(|\hat{q} - E(\hat{q})| \ge \frac{\delta}{2}\right) \\
= & \sum_{h=\frac{n^2(1-\tau)^2(K-1)\pi_0(4-\pi_0)}{8K}}^{\frac{(K-1)n^2(1-\tau)^2}{2K}} Pr\left(|\hat{q} - E(\hat{q})| \ge \frac{\delta}{2} \;\middle|\; |B^{(2)}_{\phi_K}| = h\right) Pr\left(|B^{(2)}_{\phi_K}| = h\right) \\
\le & \sum_{h=\frac{n^2(1-\tau)^2(K-1)\pi_0(4-\pi_0)}{8K}}^{\frac{(K-1)n^2(1-\tau)^2}{2K}} 2\exp\left(-\frac{\delta^2}{2}h\right) Pr\left(|B^{(2)}_{\phi_K}| = h\right)
\end{aligned}$$

$$\begin{aligned} &\leq \exp\left(-\frac{n^2(1-\tau)^2(K-1)\pi_0(4-\pi_0)\delta^2}{16K}\right) \\ &\leq \exp(-C_q n^2(1-\tau)^2), \end{aligned} \tag{4.26}$$

for some constant $C_q > 0$ dependent on K, π_0, and δ. (4.23) and (4.26) together imply

$$Pr\left(\frac{\delta}{2} < \hat{q} < 1 - \frac{\delta}{2}\right) \geq 1 - \exp(-C_q n^2(1-\tau)^2).$$

The probability inequality (4.21) for $\hat{p}$ can be proved in a similar manner. □

Our next lemma shows that under Condition 1 and 3, NS-CV gives a consistent estimate ϕ_{K^*} on the test set $\mathcal{N}^{(2)}$ when the candidate $K = K^*$.

Lemma 4.2.6. *Suppose an undirected network $(\mathcal{N}, \mathcal{E})$ generated from a stochastic block model satisfied Condition 1 and 3. When $K = K^*$, the derived community labeling function ϕ_{K^*} on $\mathcal{N}^{(2)}$ in the Classification step of NS-CV satisfies*

$$\begin{aligned} Pr\left(\phi_{K^*} \neq \phi^*\right) \leq\ & K^* n(1-\tau)\exp\Big(-c_1 n\tau(p-q)^2\Big) \\ &+ K^*\exp\left(-c_2 n(1-\tau)\right) + g(n\tau). \end{aligned} \tag{4.27}$$

where τ is the node splitting ratio, p and q are the within and between communities connection probabilities, $g(x)$ is the function defined in Condition 3, c_1 and c_2 are positive constants dependent on π_0 (from Condition 1) and K^* only.

Proof. By Condition 1, there exists a positive constant $\pi_0 \in (0,1)$ such that $n_{min} = \min\{n_1, n_2, \ldots, n_{K^*}\} \geq n\pi_0/K^*$. Let the notation $n_k^{(1)}$ denote the cardinality of a subset of community k that belongs to the training set $\mathcal{N}^{(1)}$. Thus, each $n_k^{(1)}$ follows a hypergeometric distribution with a parameter $(\tau n, n_k, n)$. By setting $t = n_k/(2n)$

in (4.15) of Lemma 4.2.1, we have for any $n_k^{(1)}$,

$$\begin{aligned} Pr\left(n_k^{(1)} \le \frac{n\pi_0\tau}{2K^*}\right) &\le Pr\left(n_k^{(1)} \le \frac{n_k\tau}{2}\right) \\ &\le Pr\left(n_k^{(1)} - \tau n\frac{n_k}{n} \le -\tau n\frac{n_k}{2n}\right) \\ &\le \exp\left(-\frac{2\tau n(n_k^2)/(4n^2)}{1-(\tau n-1)/n}\right) \\ &\le \exp\left(-\frac{\tau\pi_0^2 n}{2K^{*2}(1-((\tau n-1)/n)}\right) \\ &\le \exp\left(-c_0 n\tau\right), \end{aligned} \tag{4.28}$$

for a positive constant c_0 dependent on K^* and π_0 only. Next we consider the case when $\phi_{K^*} = \phi^*$ on $\mathcal{N}^{(1)}$. In this case, for any node $u \in \mathcal{N}^{(2)}$, let $l_u = \phi^*(u)$ denote its true community. Then according to (4.2), we have

$$\begin{aligned} & Pr\big(\phi_{K^*}(u) \ne \phi^*(u)\big) \\ \le & \sum_{\substack{l=1,2,\ldots,K^* \\ l\ne l_u}} Pr\left(\frac{\sum_{i\in\mathcal{N}^{(1)},\phi^*(i)=l_u} A_{iu}^{(12)}}{n_{l_u}^{(1)}} < \frac{\sum_{i\in\mathcal{N}^{(1)}\phi^*(i)=l} A_{iu}^{(12)}}{n_l^{(1)}}\right) \\ \le & \sum_{\substack{l=1,2,\ldots,K^* \\ l\ne l_u}} \left\{ Pr\left(\frac{\sum_{i\in\mathcal{N}^{(1)},\phi^*(i)=l_u} A_{iu}^{(12)}}{n_{l_u}^{(1)}} - p < -\frac{p-q}{2}\right)\right. \\ & \left. + Pr\left(\frac{\sum_{i\in\mathcal{N}^{(1)},\phi^*(i)=l} A_{iu}^{(12)}}{n_l^{(1)}} - q > \frac{p-q}{2}\right)\right\}. \end{aligned} \tag{4.29}$$

Then we apply the Law of Total Probability and the Hoeffding Inequality to obtain

$$Pr\left(\frac{\sum_{i\in\mathcal{N}^{(1)},\phi^*(i)=l_u} A_{iu}^{(12)}}{n_{l_u}^{(1)}} - p < -\frac{p-q}{2}\right)$$

$$
\begin{aligned}
&\leq \sum_{h=0}^{n\tau} Pr\left(\frac{\sum_{i\in\mathcal{N}^{(1)},\phi^*(i)=l_u} A_{iu}^{(12)}}{n_{l_u}^{(1)}} - p < -\frac{p-q}{2} \;\middle|\; n_{l_u}^{(1)} = h \right) \cdot Pr\left(n_{l_u}^{(1)} = h \right) \\
&\leq \sum_{h=(n\pi_0\tau)/(2K^*)}^{n\tau} Pr\left(\frac{\sum_{i\in\mathcal{N}^{(1)},\phi^*(i)=l_u} A_{iu}^{(12)}}{n_{l_u}^{(1)}} - p < -\frac{p-q}{2} \;\middle|\; n_{l_u}^{(1)} = h \right) \cdot Pr\left(n_{l_u}^{(1)} = h \right) \\
&\quad + Pr\left(n_{l_u}^{(1)} \leq \frac{n\pi_0\tau}{2K^*} \right) \\
&\leq \sum_{h=(n\pi_0\tau)/(2K^*)}^{n\tau} \exp\left(-\frac{h(p-q)^2}{2} \right) \cdot Pr\left(n_{l_u}^{(1)} = h \right) + \exp(-c_0 n\tau) \\
&\leq \exp\left(-\frac{n\tau\pi_0(p-q)^2}{4K^*} \right) + \exp\left(-c_0 n\tau\right) \\
&\leq \exp\Big(- c_1 n\tau (p-q)^2 \Big), \qquad (4.30)
\end{aligned}
$$

for some constant $c_1 > 0$. Similarly, we can prove

$$
Pr\left(\frac{\sum_{i\in\mathcal{N}^{(1)},\phi^*(i)=l} A_{iu}^{(12)}}{n_l^{(1)}} - q > \frac{p-q}{2} \right) \leq \exp\Big(- c_1 n\tau (p-q)^2 \Big). \qquad (4.31)
$$

Therefore, (4.29) can be re-written as

$$
\begin{aligned}
Pr\big(\phi_{K^*}(u) \neq \phi^*(u)\big) &\leq \sum_{\substack{l=1,2,\ldots,K^* \\ l\neq l_u}} 2\exp\Big(- c_1 n\tau (p-q)^2 \Big) \\
&= 2K^* \exp\Big(- c_1 n\tau (p-q)^2 \Big). \qquad (4.32)
\end{aligned}
$$

It is also known by Condition 3 that the probability of $\phi_{K^*} \neq \phi^*$ on $\mathcal{N}^{(1)}$ is bounded by $g(n\tau)$. Thus on $\mathcal{N}^{(2)}$, the ϕ_{K^*} obtained according to (4.2) shall satisfy

$$
\begin{aligned}
Pr\left(\phi_{K^*} \neq \phi^*\right) &\leq \sum_{u\in\mathcal{N}^{(2)}} Pr\left(\phi_{K^*}(u) \neq \phi^*(u)\right) + g(n\tau) \\
&\leq n(1-\tau) 2K^* \exp\Big(- c_1 n\tau (p-q)^2 \Big) + g(n\tau)
\end{aligned}
$$

$$\leq \quad K^* n(1-\tau)\exp\Big(-c_1 n\tau(p-q)^2\Big) + g(n\tau). \tag{4.33}$$

Finally by using a similar argument as (4.28), we can show there exists a positive constant c_2 such that

$$Pr\left(n_k^{(2)} \leq \frac{n\pi_0(1-\tau)}{2K^*}\right) \quad \leq \quad \exp\left(-c_2 n(1-\tau)\right) \text{ for } k = 1, 2, \ldots, K^*. \tag{4.34}$$

Hence,

$$Pr\left(\min_{1\leq k\leq K} n_k^{(2)} \leq \frac{n\pi_0(1-\tau)}{2K^*}\right) \quad \leq \quad K^* \exp\left(-c_2 n(1-\tau)\right). \tag{4.35}$$

Recall that the re-classification process as described in the proof of Lemma 4.2.5 is only needed when the recovered network structure for $\mathcal{N}^{(2)}$ is not balanced with respective to $\pi_0/2$. Thus according to (4.33) and (4.35), when $K = K^*$, the derived labeling function ϕ_{K^*} on $\mathcal{N}^{(2)}$ even with the re-classification shall satisfy

$$\begin{aligned} Pr\left(\phi_{K^*} \neq \phi^*\right) \quad \leq \quad & K^* n(1-\tau)\exp\Big(-c_1 n\tau(p-q)^2\Big) \\ & + K^* \exp\left(-c_2 n(1-\tau)\right) + g(n\tau). \end{aligned}$$

□

Lemma 4.2.6 applies to the case where $K = K^*$. When a candidate $K \neq K^*$ in NS-CV, define

$$\begin{aligned} T_{11} \quad &:= \quad \{(i,j) : \ i,j \in \mathcal{N},\ i<j,\ \phi_K(i) = \phi_K(j),\ \phi^*(i) = \phi^*(j)\}; \\ T_{12} \quad &:= \quad \{(i,j) : \ i,j \in \mathcal{N},\ i<j,\ \phi_K(i) \neq \phi_K(j),\ \phi^*(i) = \phi^*(j)\}; \\ T_{21} \quad &:= \quad \{(i,j) : \ i,j \in \mathcal{N},\ i<j,\ \phi_K(i) = \phi_K(j),\ \phi^*(i) \neq \phi^*(j)\}; \\ T_{22} \quad &:= \quad \{(i,j) : \ i,j \in \mathcal{N},\ i<j,\ \phi_K(i) \neq \phi_K(j),\ \phi^*(i) \neq \phi^*(j)\}. \end{aligned} \tag{4.36}$$

The next two lemmas show that when $K > K^*$ the cardinalities of T_{12} and T_{22} have a lower bound, and when $K < K^*$ the cardinalities of T_{11} and T_{21} have a lower bound.

Lemma 4.2.7. *Under Condition 1, for any $K^+ > K^*$, there exist positive constants $\gamma_1, \gamma_2 \in (0,1)$ depending on K^+, such that*

$$|T_{12}| \geq \frac{\gamma_1 \pi_0^2}{K^{*2}} n^2, \tag{4.37}$$

$$|T_{22}| \geq \frac{\gamma_2 \pi_0^2}{K^{*2}} n^2. \tag{4.38}$$

Proof. First, consider the true community structure in the network as $\mathcal{N} = [n] = \cup_{k=1}^{K^*} C_k^*$ with $|C_k^*| = n_i$. For a candidate $K^+ > K^*$, we have the estimated community structure as $\mathcal{N} = \cup_{k=1}^{K^+} \hat{C}_k$ with $|\hat{C}_i| = \hat{n}_i$. Because the recovered network is assumed to have a balanced community structure under Condition 1, there exists a positive constant $\pi_0 \in (0,1)$ such that $\hat{n}_{min} = \min_{1 \leq i \leq K^+} (\hat{n}_i) \geq \pi_0 n / K^+$ for a $\pi_0 \in (0,1)$.

Since $K^+ > K^*$, there must be at least one community under the true labeling function ϕ^* that has a certain proportion of its nodes being assigned to different communities and the proportion is at least in an order of $\delta_1 n_{min}/n$ for some positive $\delta_1 \in (0,1)$ depending on K^+. Otherwise, we assume for each of the original community, given any small $\epsilon > 0$, the number of nodes that are split out satisfies

$$S_k < n_k \epsilon \quad \text{for} \quad k = 1, \cdots, K^*.$$

Then the nodes that are split out have cardinality at most

$$\sum_{k=1}^{K^*} S_k < n\epsilon.$$

Because ϵ is arbitrarily small, there must be at least one community whose size is smaller than $n\epsilon$. This contradicts the balanced community structure assumption

that $\min_{1\le k\le K^*} n_k \ge n\pi_0/K^*$. Without loss of generality, we may assume the 1st community based on the true labeling function ϕ^* has at least $\delta_1 n_{\min}$ nodes assigned to different communities under the estimated labeling function $\hat{\phi}$ in NS-CV. Thus,

$$|T_{12}| \;\; \ge \;\; (n_1 - \delta_1 n_{min})(\delta_1 n_{min}) \ge \delta_1(1-\delta_1)\frac{\pi_0^2 n^2}{K^{*2}}.$$

This concludes (4.37). To prove (4.38), without losing generality assume that 1st community has the largest size n_1 among all other communities under ϕ^*. When the network is re-clustered to K^+ communities under $\hat{\phi}$, at least one of the K^+ communities (say Community A) will contain a minimum of $n_1^{'}$ nodes from the 1st community for $n_1/K^+ \le n_1^{'} \le n_1$. Consider the following two scenarios: (1) $\limsup(n_1^{'}/n_1) = 1$, (2) $\limsup(n_1^{'}/n_1) < 1$. In the first scenario, we can conclude there exists at least another community under ϕ^* that will have a minimum of ρn_2 nodes assigned to a different Community B under $\hat{\phi}$ for some $\rho \in (0,1)$. This is because otherwise every other communities under ϕ^* must have at least $(1-\epsilon_k)n_k$ nodes assigned to Community A under $\hat{\phi}$ for an arbitrarily small $\epsilon_k = o(1), k = 2, 3, \ldots, K^*$. In consequence, the number of nodes not assigned to Community A will be at most $(n_1 - n_1^{'}) + \sum_{k=2}^{K^*} \epsilon_k n_k = o(n)$. This contradicts to the balanced community structure condition. Therefore, in Scenarios (1),

$$|T_{22}| \ge n_1^{'}\rho n_2 \ge \frac{\gamma_2 \pi_0^2}{K^{*2}} n^2, \tag{4.39}$$

for some positive constant $\gamma_2 \in (0,1)$. In the second scenario, if the 2nd community under ϕ^* has ρn_2 nodes not assigned to Community A under $\hat{\phi}$ for some $\rho \in (0,1)$, then the same inequality (4.39) will hold true for $|T_{22}|$. On the other hand, if the 2nd community under ϕ^* has $(1-\epsilon)n_2$ nodes assigned to Community A under $\hat{\phi}$ for some

$\epsilon = o(n_2)$, then

$$|T_{22}| \geq (n_1 - n_1^{'})(1-\epsilon)n_2 \geq \frac{\gamma_2 \pi_0^2}{K^{*2}} n^2,$$

for some positive constant $\gamma_2 \in (0,1)$. □

Lemma 4.2.8. *Under Condition 1, for any $K^- < K^*$, there exists positive constants $\nu_1, \nu_2 \in (0,1)$ depending on K^-, such that*

$$|T_{21}| \geq \frac{\nu_1 \pi_0^2}{K^{*2}} n^2, \tag{4.40}$$

$$|T_{11}| \geq \frac{\nu_2 \pi_0^2}{K^{*2}} n^2. \tag{4.41}$$

Proof. The inequality (4.40) can be proved in a similar way as the proof of (4.37) in Lemma 4.2.7. To prove (4.41), assume the 1st community under ϕ^* has n_1 nodes. When the network is re-clustered into K^- community, at least one of the K^- communities must contain at least $\lfloor n_1/K^- \rfloor$ nodes from the 1st community. Therefore,

$$|T_{11}| \geq \binom{\lfloor n_1/K^- \rfloor}{2} \geq \frac{\nu_2 \pi_0^2}{K^{*2}} n^2,$$

for some constant $\nu_2 \in (0,1)$. □

Note that T_{11}, T_{12}, T_{21}, and T_{22} are for all pairs of the nodes in $\mathcal{N}$. Since the log likelihood function for NS-CV is evaluated only for the pair of nodes in the test set $\mathcal{N}^{(2)}$, it is beneficial to introduce the following notations:

$$\begin{aligned}
T_{11}^{(2)} &:= \left\{(i,j) : \ i,j \in \mathcal{N}^{(2)}, \ i<j, \ \phi_K(i) = \phi_K(j), \ \phi^*(i) = \phi^*(j)\right\}; \\
T_{12}^{(2)} &:= \left\{(i,j) : \ i,j \in \mathcal{N}^{(2)}, \ i<j, \ \phi_K(i) \neq \phi_K(j), \ \phi^*(i) = \phi^*(j)\right\}; \\
T_{21}^{(2)} &:= \left\{(i,j) : \ i,j \in \mathcal{N}^{(2)}, \ i<j, \ \phi_K(i) = \phi_K(j), \ \phi^*(i) \neq \phi^*(j)\right\}; \\
T_{22}^{(2)} &:= \left\{(i,j) : \ i,j \in \mathcal{N}^{(2)}, \ i<j, \ \phi_K(i) \neq \phi_K(j), \ \phi^*(i) \neq \phi^*(j)\right\}.
\end{aligned} \tag{4.42}$$

Corollary 4.2.9. *Under Condition 1, for any $K^+ > K^*$, there exist positive constants $\gamma_1, \gamma_2 \in (0,1)$ depending on K^+, such that*

$$|T_{12}^{(2)}| \geq \frac{\gamma_1 \pi_0^2}{4K^{*2}} n^2 (1-\tau)^2, \tag{4.43}$$

$$|T_{22}^{(2)}| \geq \frac{\gamma_2 \pi_0^2}{4K^{*2}} n^2 (1-\tau)^2. \tag{4.44}$$

And for any $K^- < K^*$, there exists positive constants $\nu_1, \nu_2 \in (0,1)$ depending on K^-, such that

$$|T_{21}^{(2)}| \geq \frac{\nu_1 \pi_0^2}{4K^{*2}} n^2 (1-\tau)^2, \tag{4.45}$$

$$|T_{11}^{(2)}| \geq \frac{\nu_2 \pi_0^2}{4K^{*2}} n^2 (1-\tau)^2. \tag{4.46}$$

Proof. It was shown in the proof of Lemma 4.2.5 that by applying the re-classification according to (4.24), the recovered network structure for $\mathcal{N}^{(2)}$ is balanced with respect to $\pi_0/2$. Note that $\mathcal{N}^{(2)}$ has a size of $n(1-\tau)$. Thus (4.43), (4.44), (4.45), and (4.46) are direct results from Lemma 4.2.7 and Lemma 4.2.8. □

Lemma 4.2.10. *Under Condition 1, when the candidate community number $K^+ > K^*$ in NS-CV, the estimated between-community probability $\hat{q}$ by equation (4.13) shall diverge from the true parameter q in a sense that there exists a positive constant $d^+ = \gamma_1 \pi_0^2 (p-q)/(2\gamma_1 \pi_0^2 + 8K^{*2})$ such that*

$$Pr\left(\hat{q} - q > d^+\right) \geq 1 - \exp\left(-C_{K^+} n^2 (1-\tau)^2 (p-q)^2\right), \tag{4.47}$$

where C_{K^+} is a positive constant depending on π_0, K^*, and K^+. On the other hand, with a candidate community number $K^- < K^*$ in NS-CV, the estimated within-community probability $\hat{p}$ by equation (4.13) shall diverge from the true parameter p in a sense that there exists a positive constant $d^- = \nu_1 \pi_0^2 (p-q)/(2\nu_1 \pi_0^2 + 8K^{*2})$ such

that

$$Pr\left(\hat{p}-p<-d^{-}\right)\geq 1-\exp\left(-C_{K^{-}}n^{2}(1-\tau)^{2}\left(p-q\right)^{2}\right), \tag{4.48}$$

where C_{K^-} *is a positive constant depending on* π_0, K^*, *and* K^-.

Proof. When $K^+ > K^*$, the expected value of $\hat{q}$ can be re-written as

$$E\left(\hat{q}\right)=\frac{p|T_{12}^{(2)}|+q|T_{22}^{(2)}|}{|T_{12}^{(2)}|+|T_{22}^{(2)}|}=\eta p+(1-\eta)q,$$

where $\eta = |T_{12}^{(2)}|/(|T_{12}^{(2)}| + |T_{22}^{(2)}|)$. Next, define

$$\omega:=\frac{1}{1+\frac{4K^{*2}}{\gamma_1\pi_0^2}}>0. \tag{4.49}$$

Thus by (4.43) and the fact that $|T_{22}^{(2)}| \leq n^2(1-\tau)^2$, we have $\eta \geq \omega$. Hence,

$$\begin{aligned} Pr\left(\hat{q}-q>\frac{\omega(p-q)}{2}\right) &= Pr\left(\hat{q}-E\left(\hat{q}\right)+E\left(\hat{q}\right)-q>\frac{\omega(p-q)}{2}\right) \\ &= Pr\left(\hat{q}-E\left(\hat{q}\right)+\eta(p-q)>\frac{\omega(p-q)}{2}\right) \\ &\geq 1-Pr\left(\hat{q}-E\left(\hat{q}\right)\leq-\frac{\omega(p-q)}{2}\right). \end{aligned} \tag{4.50}$$

Recall $\hat{q}$ has a form given in (4.13). By a similar argument as used in (4.26), we have

$$Pr\left(\hat{q}-E\left(\hat{q}\right)\leq-\frac{w(p-q)}{2}\right) \leq \exp\left(-C_{K^{+}}n^{2}(1-\tau)^{2}(p-q)^{2}\right), \tag{4.51}$$

where C_{K^+} is a positive constant dependent on π_0 and K^+. Finally combining (4.50) and (4.51), we conclude

$$Pr\left(\hat{q}-q>\frac{\omega(p-q)}{2}\right)\geq 1-\exp\left(-C_{K^{+}}n^{2}(1-\tau)^{2}\left(p-q\right)^{2}\right). \tag{4.52}$$

This concludes (4.47). The other inequality (4.48) in the lemma can be proved in a similar way. □

Our last lemma in this section gives a lower and upper bound for the Kullback-Leibler divergence from one Bernoulli random variable to another. This lemma will be used for the proof of the main consistency theorem for NS-CV in the next section.

Lemma 4.2.11. *P_1 and P_2 are two Bernoulli distributions with the parameter $p_1 \in (0,1)$ and $p_2 \in (0,1)$ respectively. The Kullback-Leibler divergence of P_2 from P_1 satisfies*

$$\frac{(p_1-p_2)^2}{(p_1+p_2)(2-p_1-p_2)} \leq D_{KL}(P_1||P_2) \leq \frac{(p_1-p_2)^2}{p_2(1-p_2)}. \tag{4.53}$$

Proof. Recall that the Kullback-Leibler divergence of P_2 from P_1 has a form

$$D_{KL}(P_1||P_2) = p_1 \log \frac{p_1}{p_2} + (1-p_1) \log \frac{1-p_1}{1-p_2}. \tag{4.54}$$

First consider the case of $p_1 \geq p_2$, where we may write

$$D_{KL}(P_1||P_2) = \int_{p_2}^{p_1} \left(\frac{p_1}{x} - \frac{1-p_1}{1-x} \right) dx. \tag{4.55}$$

Since $f(x) = p_1/x - (1-p_1)/(1-x)$ is a non-negative continuous decreasing function on $[p_2, p_1]$, then by the Mean Value Theorem, there exists a constant $c \in (p_2, \frac{p_1+p_2}{2})$ such that

$$\begin{aligned} D_{KL}(P_1||P_2) &\geq \int_{p_2}^{\frac{p_1+p_2}{2}} \left(\frac{p_1}{x} - \frac{1-p_1}{1-x} \right) dx \\ &= \left(\frac{p_1}{c} - \frac{1-p_1}{1-c} \right) \left(\frac{p_1+p_2}{2} - p_2 \right) \\ &\geq \left(\frac{p_1}{(p_1+p_2)/2} - \frac{1-p_1}{1-(p_1+p_2)/2} \right) \left(\frac{p_1-p_2}{2} \right) \end{aligned}$$

$$= \frac{(p_1 - p_2)^2}{(p_1 + p_2)(2 - p_1 - p_2)}. \tag{4.56}$$

Also since $0 \leq f(x) \leq f(p_2) = (p_1 - p_2)/(p_2(1 - p_2))$,

$$\begin{aligned} D_{KL}(P_1||P_2) &= \int_{p_2}^{p_1} \left(\frac{p_1}{x} - \frac{1 - p_1}{1 - x} \right) dx \\ &\leq \frac{p_1 - p_2}{p_2(1 - p_2)}(p_1 - p_2). \end{aligned}$$

When $p_1 < p_2$, write $q_1 = 1 - p_1$ and $q_2 = 1 - p_2$ so that $q_1 > q_2$. Notice that both the Kullback-Leibler divergence and the lower and upper bound in (4.53) are invariant between p_1, p_2 and q_1, q_2. So (4.53) holds true for q_1 and q_2, as well as for p_1 and p_2. □

4.2.3 Main Consistency Theorem

In this section, we present a consistency theorem for NS-CV under a general setting where the within and between community probability p, q, and the node splitting ratio τ are all considered dependent on the network size n.

Theorem 4.2.12 (Consistency of NS-CV)**.** *Consider an undirected network $(\mathcal{N}, \mathcal{E})$ generated from a stochastic block model with a true community number K^* satisfying $2 \leq K^* \leq K_{max}$ for some positive integer K_{max}. The parameter p and q are the within and between community connection probabilities. The NS-CV has a splitting ratio $\tau = |\mathcal{N}^{(1)}|/|\mathcal{N}| \in (0, 1)$. Under Condition 1, 2 and 3, if the following requirements are met for some constant $\alpha > 0$:*

(i) $\tau = \omega\Big(\sqrt{\log n}/n\Big)$, $\tau(p - q)^2 = \omega\Big((\log n)^{(1/2+\alpha)}/n\Big)$,

(ii) $1 - \tau = \omega\Big(\sqrt{\log n}/n\Big)$, $(1 - \tau)(p - q) = \omega\Big(\sqrt{\log n}/n\Big)$,

then the optimal community number $K_{NS\text{-}CV}$ obtained from the NS-CV procedure as

$$K_{NS\text{-}CV} = \underset{2\leq K\leq K_{max}}{\arg\max}\ \ell\left(\hat{p}, \hat{q} \mid \phi_K\right) \tag{4.57}$$

satisfies

$$\lim_{n\to\infty} Pr(K_{NS\text{-}CV} = K^*) = 1. \tag{4.58}$$

Comments: Condition (i) speaks for the cross validation splitting ratio τ. Intuitively, τ can't be too small, because otherwise the training set would be too small to recover the community structure. On the other hand, from condition (ii), the test ratio $1-\tau$ can't be small either, because the test subset needs to be large enough to estimate p and q and to evaluate the the likelihood function. Also notice that the second part of Condition (i) and (ii) states dependency of τ, $1-\tau$ on $p-q$. When the discrepancy $p-q$ of the network is large enough to distinguish different communities, the splitting ratio τ could be close to either 0 or 1. However, when the discrepancy $p-q$ of the network is not large enough to distinguish different communities, τ may be required as a constant. It is obvious that the two conditions are met in a simple case where τ, p, and q are constant. Finally, it is worthwhile to highlight two differences between our main consistency theorem and the consistency result presented by Chen and Lei (2018) for their network cross validation. First, Chen and Lei have only proved a partial version of consistency, because they only showed $Pr(\hat{K} < K^*) \to 0$, but gave no guarantee for $\hat{K} = K^*$ in probability. Secondly, they have only studied the f-fold cross validation strategy, where f is considered a fixed integer. We extend the discussion to cover the case where the cross-validation node splitting ratio τ and the connection probability p and q are treated as variables dependent on the network size n.

Proof. For a candidate $K \in \{2, \ldots, K_{\max}\}$, denote the community labeling function derived from NS-CV as ϕ_K and denote the maximized log likelihood function as $\hat{\ell}_K = \ell(\hat{p}, \hat{q} \mid \phi_K)$ whose form is given in (4.12). In order to prove (4.58), it is equivalent to show when $K \neq K^*$,

$$Pr\left(\hat{\ell}_{K^*} - \hat{\ell}_K > 0\right) \to 1. \tag{4.59}$$

Because $\hat{p}$ and $\hat{q}$ are the maximum likelihood estimator defined in (4.13), it is natural to have

$$\hat{\ell}_{K^*} = \ell(\hat{p}, \hat{q} \mid \phi_{K^*}) \geq \ell(p, q \mid \phi_{K^*}), \tag{4.60}$$

where p and q are the true within and between community connection probabilities for the network. For simplicity, denote $\ell(p, q \mid \phi_{K^*})$ as $\tilde{\ell}_{K^*}$. Thus it is sufficient to show when $K \neq K^*$,

$$Pr\left(\tilde{\ell}_{K^*} - \hat{\ell}_K > 0\right) \to 1. \tag{4.61}$$

Recall that under Condition 1 and 3, Lemma 4.2.6 states on the test set $\mathcal{N}^{(2)}$,

$$\begin{aligned} Pr(\phi_{K^*} \neq \phi^*) \;\leq\; & K^* n(1-\tau) \exp\Big(-c_1 n\tau (p-q)^2\Big) \\ & + K^* \exp(-c_2 n(1-\tau)) + g(n\tau), \end{aligned} \tag{4.62}$$

where c_1 and c_2 are positive constants dependent on K^* and π_0 only. The condition (i) and (ii) in the theorem statement imply $Pr(\phi_{K^*} = \phi^*) \to 1$ as $n \to \infty$. Write

$$\begin{aligned} & Pr\left(\tilde{\ell}_{K^*} - \hat{\ell}_K > 0\right) \\ \geq\; & Pr\left(\tilde{\ell}_{K^*} - \hat{\ell}_K > 0,\ \phi_{K^*} = \phi^*\right) \end{aligned}$$

$$= Pr\left(\tilde{\ell}_{K^*} - \hat{\ell}_K > 0 \mid \phi_{K^*} = \phi^*\right) Pr\left(\phi_{K^*} = \phi^*\right). \tag{4.63}$$

Thus in order to show (4.61), it is sufficient to prove when $K \neq K^*$,

$$Pr\left(\tilde{\ell}_{K^*} - \hat{\ell}_K > 0 \mid \phi_{K^*} = \phi^*\right) \to 1. \tag{4.64}$$

Recall the definitions of $T_{11}^{(2)}$, $T_{12}^{(2)}$, $T_{21}^{(2)}$, and $T_{22}^{(2)}$ in (4.42). The maximized log likelihood $\hat{\ell}_K = \ell\left(\hat{p}, \hat{q} \mid \phi_K\right)$ defined in (4.12) can be re-written as

$$\begin{aligned}
\hat{\ell}_K &= \sum_{(i,j)\in T_{11}^{(2)}} \left(A_{ij}^{(22)} \log \hat{p} + (1 - A_{ij}^{(22)}) \log(1-\hat{p})\right) \\
&+ \sum_{(i,j)\in T_{12}^{(2)}} \left(A_{ij}^{(22)} \log \hat{q} + (1 - A_{ij}^{(22)}) \log(1-\hat{q})\right) \\
&+ \sum_{(i,j)\in T_{21}^{(2)}} \left(A_{ij}^{(22)} \log \hat{p} + (1 - A_{ij}^{(22)}) \log(1-\hat{p})\right) \\
&+ \sum_{(i,j)\in T_{22}^{(2)}} \left(A_{ij}^{(22)} \log \hat{q} + (1 - A_{ij}^{(22)}) \log(1-\hat{q})\right).
\end{aligned} \tag{4.65}$$

Similarly when $\phi_{K^*} = \phi^*$, $\tilde{\ell}_{K^*} = \ell\left(p, q \mid \phi_{K^*}\right) = \ell\left(p, q \mid \phi^*\right)$ can be rewritten as

$$\begin{aligned}
\tilde{\ell}_{K^*} &= \sum_{(i,j)\in T_{11}^{(2)}} \left(A_{ij}^{(22)} \log p + (1 - A_{ij}^{(22)}) \log(1-p)\right) \\
&+ \sum_{(i,j)\in T_{12}^{(2)}} \left(A_{ij}^{(22)} \log p + (1 - A_{ij}^{(22)}) \log(1-p)\right) \\
&+ \sum_{(i,j)\in T_{21}^{(2)}} \left(A_{ij}^{(22)} \log q + (1 - A_{ij}^{(22)}) \log(1-q)\right) \\
&+ \sum_{(i,j)\in T_{22}^{(2)}} \left(A_{ij}^{(22)} \log q + (1 - A_{ij}^{(22)}) \log(1-q)\right).
\end{aligned} \tag{4.66}$$

Furthermore define

$$\begin{aligned}\tilde{\xi}_{K^*} &= \sum_{(i,j)\in T_{11}^{(2)}} \Big(p\log p + (1-p)\log(1-p)\Big) \\ &+ \sum_{(i,j)\in T_{12}^{(2)}} \Big(p\log p + (1-p)\log(1-p)\Big) \\ &+ \sum_{(i,j)\in T_{21}^{(2)}} \Big(q\log q + (1-q)\log(1-q)\Big) \\ &+ \sum_{(i,j)\in T_{22}^{(2)}} \Big(q\log q + (1-q)\log(1-q)\Big); \end{aligned} \tag{4.67}$$

and for $K \neq K^*$ define

$$\begin{aligned}\hat{\xi}_{K} &= \sum_{(i,j)\in T_{11}^{(2)}} \Big(p\log \hat{p} + (1-p)\log(1-\hat{p})\Big) \\ &+ \sum_{(i,j)\in T_{12}^{(2)}} \Big(p\log \hat{q} + (1-p)\log(1-\hat{q})\Big) \\ &+ \sum_{(i,j)\in T_{21}^{(2)}} \Big(q\log \hat{p} + (1-q)\log(1-\hat{p})\Big) \\ &+ \sum_{(i,j)\in T_{22}^{(2)}} \Big(q\log \hat{q} + (1-q)\log(1-\hat{q})\Big). \end{aligned} \tag{4.68}$$

Then we can write

$$\begin{aligned}\tilde{\ell}_{K^*} - \hat{\ell}_K &= \Big(\tilde{\ell}_{K^*} - \tilde{\xi}_{K^*} + \hat{\xi}_K - \hat{\ell}_K\Big) + \Big(\tilde{\xi}_{K^*} - \hat{\xi}_K\Big) \\ &= I + II. \end{aligned} \tag{4.69}$$

For I, we have

$$I = \tilde{\ell}_{K^*} - \tilde{\xi}_{K^*} + \hat{\xi}_K - \hat{\ell}_K$$

$$
\begin{aligned}
&= \sum_{(i,j)\in T_{11}^{(2)}} \Big(A_{ij}^{(22)} - p\Big) \log \frac{p(1-\hat{p})}{(1-p)\hat{p}} + \sum_{(i,j)\in T_{12}^{(2)}} \Big(A_{ij}^{(22)} - p\Big) \log \frac{p(1-\hat{q})}{(1-p)\hat{q}} \\
&+ \sum_{(i,j)\in T_{21}^{(2)}} \Big(A_{ij}^{(22)} - q\Big) \log \frac{q(1-\hat{p})}{(1-q)\hat{p}} + \sum_{(i,j)\in T_{22}^{(2)}} \Big(A_{ij}^{(22)} - q\Big) \log \frac{q(1-\hat{q})}{(1-q)\hat{q}} \\
&= I_{11} + I_{12} + I_{21} + I_{22} \qquad (4.70)
\end{aligned}
$$

By Lemma 4.2.5, there exists an constant $M_{11} > 0$ depending on δ in Condition 2 so that

$$
Pr\left(\Big|\log \frac{p(1-\hat{p})}{(1-p)\hat{p}}\Big| \geq M_{11}\right) \leq \exp\left(-C_p n^2 (1-\tau)^2\right). \qquad (4.71)
$$

Therefore using the Hoeffding Inequality (Hoeffding, 1963), we obtain

$$
\begin{aligned}
& Pr\left(\Big|\sum_{(i,j)\in T_{11}^{(2)}} \Big(A_{ij}^{(22)} - p\Big) \log \frac{p(1-\hat{p})}{(1-p)\hat{p}}\Big| > n(1-\tau)\log\big(n(1-\tau)\big)\right) \\
\leq\ & Pr\left(\Big|\sum_{(i,j)\in T_{11}^{(2)}} \Big(A_{ij}^{(22)} - p\Big)\Big| > \frac{n(1-\tau)\log\big(n(1-\tau)\big)}{M_{11}}\right) \\
& + Pr\left(\Big|\log \frac{p(1-\hat{p})}{(1-p)\hat{p}}\Big| \geq M_{11}\right) \\
\leq\ & \sum_{h=0}^{n^2(1-\tau)^2} Pr\left(\Big|\sum_{(i,j)\in T_{11}^{(2)}} \Big(A_{ij}^{(22)} - p\Big)\Big| > \frac{n(1-\tau)\log\big(n(1-\tau)\big)}{M_{11}} \;\Big|\; \big|T_{11}^{(2)}\big| = h\right) \cdot \\
& Pr\left(\big|T_{11}^{(2)}\big| = h\right) + \exp\left(-C_p n^2(1-\tau)^2\right) \\
\leq\ & 2\sum_{h=0}^{n^2(1-\tau)^2} \exp\left(-\frac{2n^2(1-\tau)^2\log^2\big(n(1-\tau)\big)}{M_{11}^2 h}\right) Pr\left(\big|T_{11}^{(2)}\big| = h\right) \\
+\ & \exp\left(-C_p n^2(1-\tau)^2\right) \\
\leq\ & 2\exp\left(-\frac{2n^2(1-\tau)^2\log^2\big(n(1-\tau)\big)}{M_{11}^2 n^2(1-\tau)^2}\right) \sum_{h=0}^{n^2(1-\tau)^2} Pr\left(\big|T_{11}^{(2)}\big| = h\right)
\end{aligned}
$$

$$
\begin{aligned}
&+ \quad \exp\left(-C_p n^2(1-\tau)^2\right) \\
&\leq \quad \exp\left(-\frac{2\log^2\left(n(1-\tau)\right)}{M_{11}^2}\right) + \exp\left(-C_p n^2(1-\tau)^2\right). \qquad (4.72)
\end{aligned}
$$

Similar inequalities can be derived for the other three terms I_{12}, I_{21}, and I_{22} in (4.70). This concludes that under condition (ii),

$$
I = \tilde{\ell}_{K^*} - \tilde{\xi}_{K^*} + \hat{\xi}_K - \hat{\ell}_K = O_p\Big(n(1-\tau)\log\big(n(1-\tau)\big)\Big), \qquad (4.73)
$$

where the symbol O_p indicates $|\tilde{\ell}_{K^*} - \tilde{\xi}_{K^*} + \hat{\xi}_K - \hat{\ell}_K|$ is bounded by an order of $n(1-\tau)\log\big(n(1-\tau)\big)$ in probability. Next we expand the term II in (4.69) as

$$
\begin{aligned}
II = \tilde{\xi}_{K^*} - \hat{\xi}_K \quad &= \quad |T_{11}^{(2)}| p \log\frac{p}{\hat{p}} + |T_{11}^{(2)}|(1-p)\log\frac{1-p}{1-\hat{p}} \\
&\quad + |T_{12}^{(2)}| p \log\frac{p}{\hat{q}} + |T_{12}^{(2)}|(1-p)\log\frac{1-p}{1-\hat{q}} \\
&\quad + |T_{21}^{(2)}| q \log\frac{q}{\hat{p}} + |T_{21}^{(2)}|(1-q)\log\frac{1-q}{1-\hat{p}} \\
&\quad + |T_{22}^{(2)}| q \log\frac{q}{\hat{q}} + |T_{22}^{(2)}|(1-q)\log\frac{1-q}{1-\hat{q}} \\
&= \quad |T_{11}^{(2)}| \cdot D_{KL}\left(p \,||\, \hat{p}\right) + |T_{12}^{(2)}| \cdot D_{KL}\left(p \,||\, \hat{q}\right) \\
&\quad + |T_{21}^{(2)}| \cdot D_{KL}\left(q \,||\, \hat{p}\right) + |T_{22}^{(2)}| \cdot D_{KL}\left(q \,||\, \hat{q}\right). \qquad (4.74)
\end{aligned}
$$

By definition, the four KL divergences in (4.74) are all positive. When $K > K^*$, only consider the term $|T_{22}^{(2)}| \cdot D_{KL}\left(q \,||\, \hat{q}\right)$. By Lemma 4.2.11,

$$
D_{KL}\left(q \,||\, \hat{q}\right) \geq \frac{(q-\hat{q})^2}{(q+\hat{q})(2-q-\hat{q})} \geq (q-\hat{q})^2. \qquad (4.75)
$$

Recall Corollary 4.2.9 states that when $K > K^*$,

$$
|T_{22}^{(2)}| \quad \geq \quad \frac{\gamma_2 \pi_0^2}{4K^{*2}} n^2(1-\tau)^2, \qquad (4.76)
$$

for some positive constant $\gamma_2 \in (0,1)$. In addition when $K > K^*$, by Lemma 4.2.10,

$$Pr\left(\hat{q} - q > d^+\right) \geq 1 - \exp\left(-C_{K^+} n^2 (1-\tau)^2 (p-q)^2\right), \tag{4.77}$$

where $d^+ = \gamma_1 \pi_0^2 (p-q)/(2\gamma_1 \pi_0^2 + 8K^{*2})$ and C_{K^+} is a positive constant depending on π_0, K^*, and K^+ only. Thus,

$$\begin{aligned} & Pr\left(|T_{22}^{(2)}| \cdot D_{KL}\left(q \,||\, \hat{q}\right) \geq \frac{n^2(1-\tau)^2\gamma_2\pi_0^2}{4K^{*2}} \left(d^+\right)^2\right) \\ \geq \;& Pr\left((q-\hat{q})^2 \geq \left(d^+\right)^2\right) \\ \geq \;& Pr\left(\hat{q} - q \geq d^+\right) \\ \geq \;& 1 - \exp\left(-C_{K^+} n^2 (1-\tau)^2 (p-q)^2\right). \end{aligned} \tag{4.78}$$

Similarly when $K < K^*$, by the inequality (4.46) in Corollary 4.2.9 and the inequality (4.48) in Lemma 4.2.10, we obtain

$$\begin{aligned} & Pr\left(|T_{11}^{(2)}| \cdot D_{KL}\left(p \,||\, \hat{p}\right) \geq \frac{n^2(1-\tau)^2\nu_2\pi_0^2}{4K^{*2}} \left(d^-\right)^2\right) \\ \geq \;& Pr\left((p-\hat{p})^2 \geq \left(d^-\right)^2\right) \\ \geq \;& Pr\left(\hat{p} - p \geq -d^-\right) \\ \geq \;& 1 - \exp\left(-C_{K^-} n^2 (1-\tau)^2 (p-q)^2\right). \end{aligned} \tag{4.79}$$

Combining (4.74), (4.78) and (4.79), we have shown that for any $K \neq K^*$, under condition (ii) in the theorem statement, there exists a constant $L > 0$ dependent on π_0 (from Condition 1), K^*, and K such that

$$Pr\left(\tilde{\xi}_{K^*} - \hat{\xi}_K \geq L n^2 (1-\tau)^2 (p-q)^2\right) \;\geq\; 1 - \exp\left(-C' n^2 (1-\tau)^2 (p-q)^2\right), \tag{4.80}$$

where C' is a positive constant. This indicates that the second term $II = \tilde{\xi}_{K^*} - \hat{\xi}_K$ in (4.69) has a minimum order of $n^2(1-\tau)^2(p-q)^2$ in probability. Finally, putting (4.73) and (4.80) together, under condition (ii), we conclude when $K \neq K^*$,

$$Pr\left(\tilde{\ell}_{K^*} - \hat{\ell}_K > 0 \mid \phi_{K^*} = \phi^*\right) \to 1, \tag{4.81}$$

as $n \to \infty$. This proves (4.64) as well as the main consistency theorem for NS-CV. □

Lei et al. 2015 has shown that under some technical conditions on the Stochastic Block Model (ϕ^*, p, q), when $K = K^*$, the Spectral Clustering used in Algorithm 1 give a consistent community labeling function ϕ_{K^*} on $\mathbf{A}^{(11)}$ with the probability of mis-clustering at most $1/n$. This states that Condition 3 holds true for NS-CV using Spectral Clustering with

$$Pr\left(\phi_{K^*} \neq \phi^*\right) \leq g(n\tau) = \frac{1}{n\tau} \tag{4.82}$$

on the training set $\mathcal{N}^{(1)}$. Therefore, we conclude under the same conditions as Theorem 4.2.12, the NS-CV procedure using Spectral Clustering can consistently choose the correct community number K^* in probability as the network size n goes to infinity.

4.3 Simulations

In this section, we present some simulation results to evaluate the performance of the NS-CV method. Our first simulation is to explore how accurately the NS-CV method can determine the community number under different network sizes. This helps us understand the speed of convergence for (4.58) in our Main Consistency Theorem 4.2.12. We simulate networks under the stochastic block model with a known community number $K^* = 4$. The network size n varies from 100 to 500.

When generating the network connections, we consider six different combinations of the within-community probability p and the between-community probability q. We also run the NS-CV method under four choices of the cross validation fold number: 2, 3, 5, and 8. The success rate of NS-CV is calculated from 100 iterations and plotted against the network size n in Figure 4.1.

Figure 4.1 shows that the NS-CV success rate is generally low at $n = 100$ in 5 out of the 6 scenarios. Only in Scenario 4 where $p = 0.6$ and $q = 0.2$, the NS-CV success rate is greater than 0.8 at $n = 100$. This is expected because it has been shown in the proof of Theorem 4.2.12 that the dominates term II in (4.69) has an order of $n^2(1-\tau)^2(p-q)^2$. Thus the selected K from the NS-CV method will have a higher convergence rate to its true value when $p - q$ is large. Our simulation also shows that in most cases the NS-CV success rate will improve quickly as the network size increases. Only in Scenario 3, the NS-CV rate remains low because the p and q are too close to each other.

In regard to the cross validation fold number, Figure 4.1 shows that 2-fold underperforms the other three choices. This suggests that NS-CV should ask for more samples in the training set than the test set especially when n is small. This is aligned with the two requirements (i) and (ii) in the main body of Theorem 4.2.12, where the order requirement for τ (the training set proportion) is higher than that for $1-\tau$ (the test set proportion). In practice, our simulation may suggest a use of 3-fold cross validation to have a good balance between accuracy and the computing loads.

Next we explore how different levels of network sparsity and the imbalance of community sizes would affect the NS-CV performance. In this simulation, we consider a large network that has $n = 500$ nodes with a varying true community number $K^* \in \{2, 3, 4, 6, 8, 10\}$. To change the network sparsity levels, we choose the between-community connection probability $q \in \{0.02, 0.05, 0.10, 0.20, 0.25, 0.30\}$ and set the within-community probability $p = 3q$. Furthermore, we control the community size

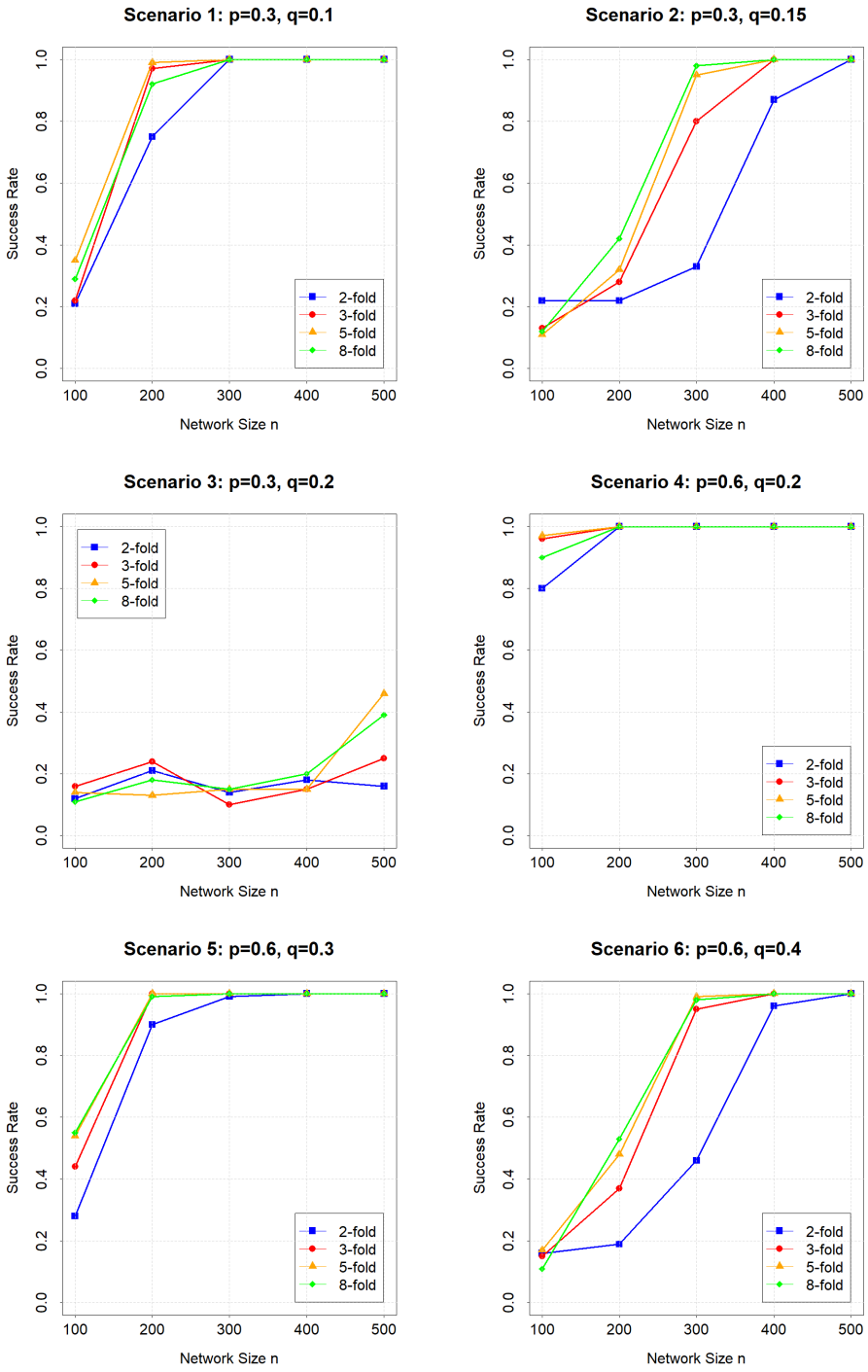


Figure 4.1: NS-CV success rates over the network sizes in six scenarios. The true model has $K^* = 4$. The success rate is calculated from 100 iterations.

imbalance level by changing the smallest community size in the network (denote as n_{min}), while maintaining the remaining communities at nearly equal sizes. In this way, the difference between n_{min} and $(n - n_{min})/(K^* - 1)$ would indicate how imbalanced the community sizes are. At each scenario of a true community number K^*, we compare the success rate of the 3-fold NS-CV calculated from 100 iterations. Our simulation results are summarized in Figure 4.2.

Figure 4.2 shows that in general the NS-CV method achieves a better performance when (1) the community numbers are low (e.g. $K^* = 2$, 3, or 4), (2) the network is not too sparse (e.g. $q \geq 0.1$, $p \geq 0.3$), and (3) the network has relatively balanced community sizes (i.e. $n_{min} \approx n/K^*$). It is also shown when the community number K^* is low, the NS-CV method has more robust performance against imbalanced community sizes and sparse networks. For instance, when $K^* = 2$, the NS-CV success rate is close to 1.0 even if the between-community probability q stays as low as 0.05 and the ratio of the two community sizes is 1:9. On the other hand, at $K^* = 10$ the NS-CV performance becomes low when the network contains a small community whose size $n_{min} = 10$. It is also worth noting that at $K^* = 10$, the NS-CV success rate starts decreasing when the network becomes less sparse (e.g. $q > 0.2$, $p > 0.6$). In other words, to determine the community number K would be a challenging task for either a sparse or a dense network with many underlying communities.

In our last simulation, we compare our NS-CV method with the network cross validation method introduced by Chen and Lei (2018) for community detection. Both cross validation methods are used in a 3-fold setting to analyze the networks of size $n = 500$ and varying $K^* \in \{2, 4, 6\}$. The success rates of the two methods are plotted side-by-side in Figure 4.3 at different network sparsity and unbalanced levels. It is clear that the success rates for NS-CV are higher or equal to Chen and Lei's method in nearly all scenarios. Specifically, the NS-CV method is shown to be robuster for the network with a large degree of unbalance (the blue curve).

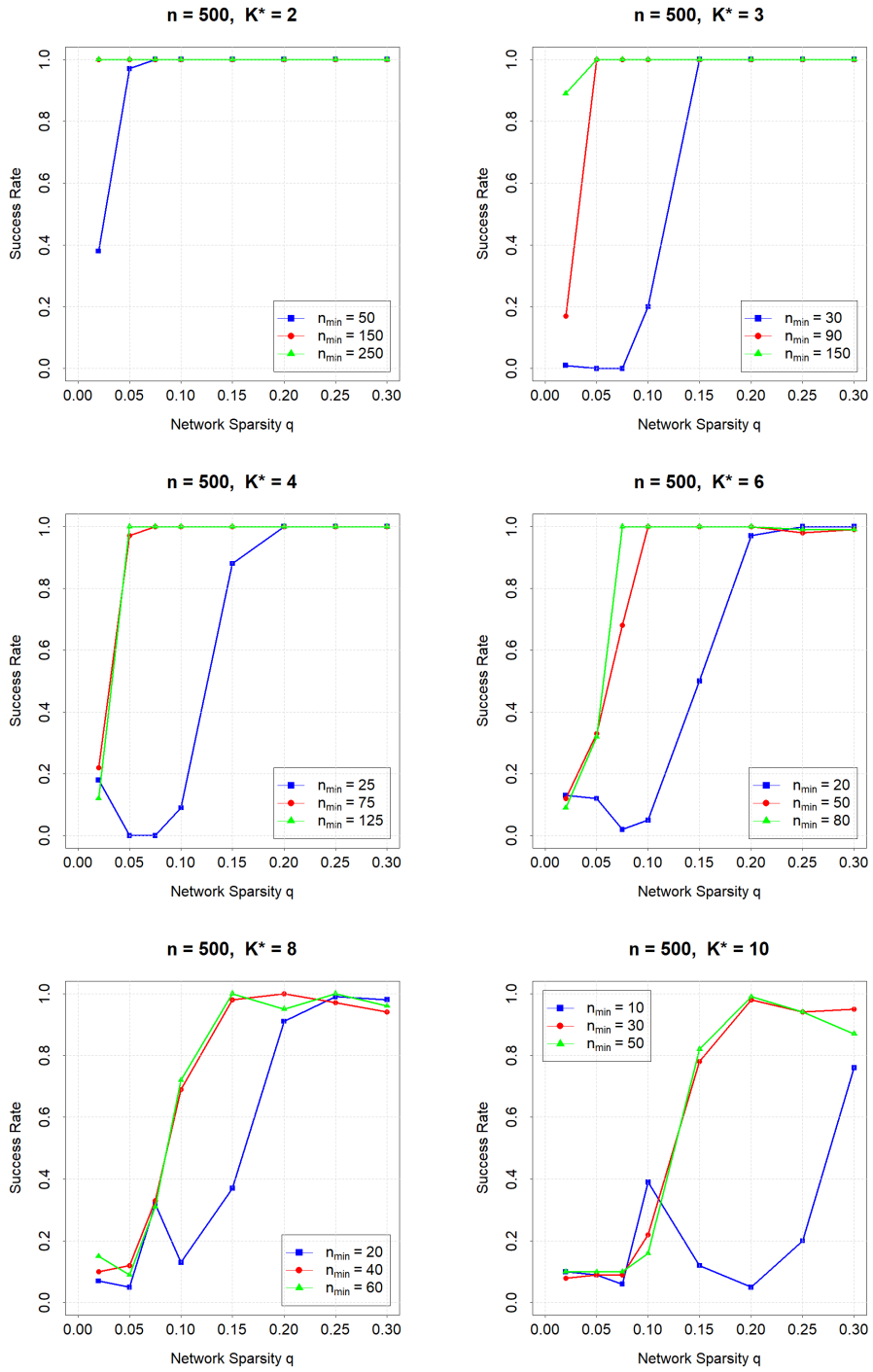


Figure 4.2: 3-fold NS-CV success rates over the network sparsity levels varying by choosing the between-community probability $q \in \{0.02, 0.05, 0.10, 0.20, 0.25, 0.3\}$ and setting the within-community probability $p = 3q$. The community imbalance levels are controlled by the smallest community size n_{min}. The success rates are calculated from 100 iterations.

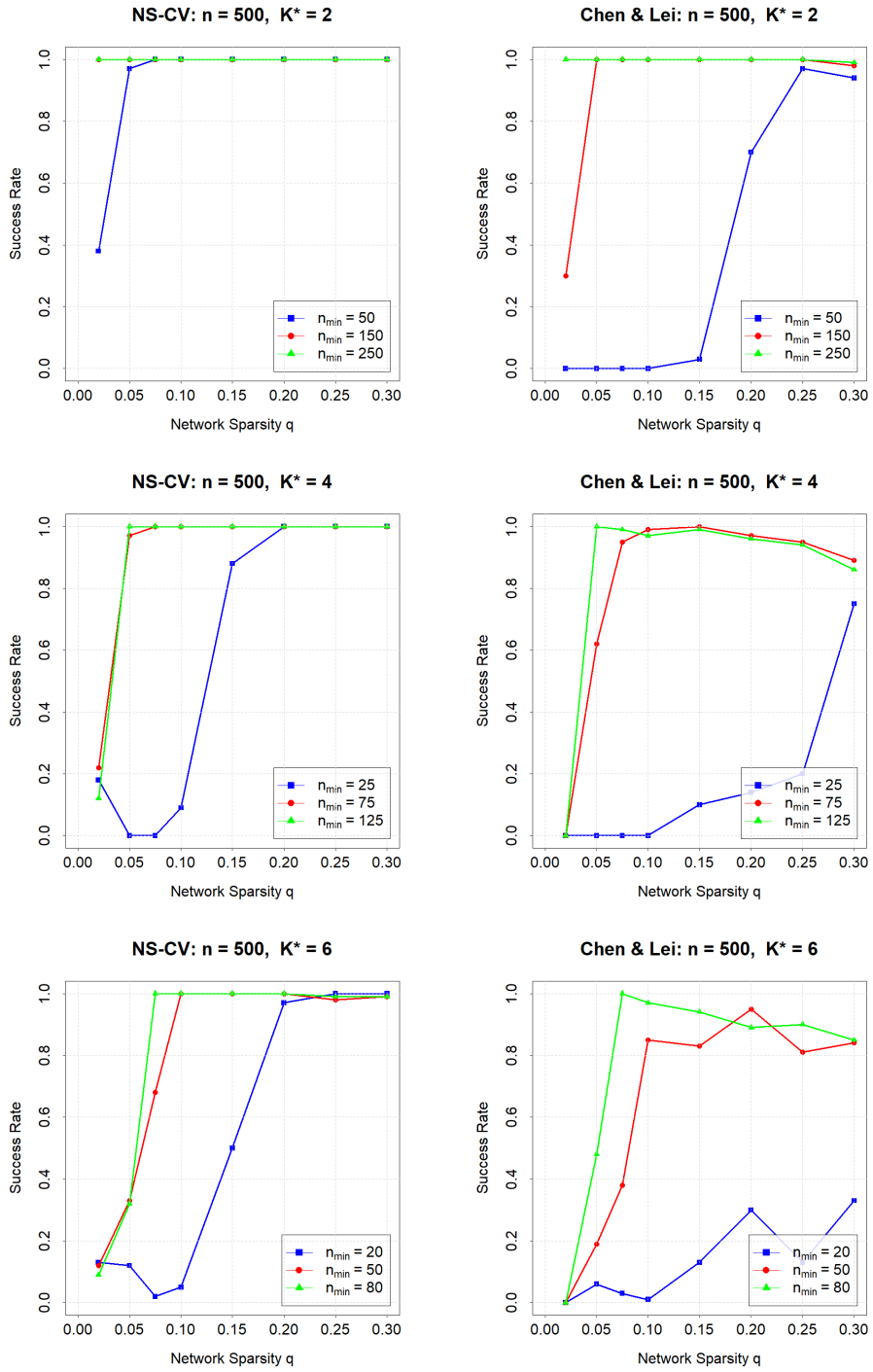


Figure 4.3: Compare NS-CV (left column) with the network cross validation method by Chen and Lei (2018) (right column). Both methods run at a 3-fold setting on the networks with $n = 500$ and $K \in \{2, 4, 6\}$. The cross validation success rates are calculated from 100 iterations at different levels of sparsity and imbalance.

4.4 Applications

In this section, we apply the NS-CV method to study two real-life networks: international trade and the U.S. Senate.

4.4.1 International Trade

The countries around the world export and import goods and services among themselves, creating a complicated network of international trade. It is known that some countries establish a close trading partnership from geographical proximity or by joining a trade agreement such as Trans-Pacific Partnership (TPP) and Transatlantic Trade and Investment Partnership (TTIP). From a global perspective, it is valuable to understand how many distinct trading groups exist in the international trade network and how a country trades with others inside or outside the trading group.

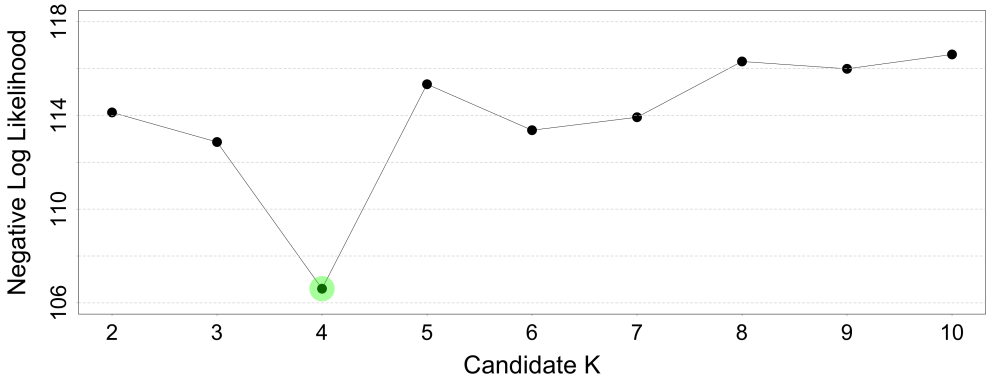
Our international trade data come from Westveld and Hoff (2011) which includes the import and export amount (in US dollars) among 58 countries in the world from 1981 to 2000. To simplify the data processing, our analysis only focuses on the data for year 2000. The first step is to construct the adjacency matrix for these 58 countries. Because connections are considered undirected in the stochastic block model, we add the import and export volume together to obtain the gross trade amount between every pair of the countries. Then a connection ($A_{ij} = 1$) is assigned between country i and country j, if the gross trade amount between the two exceeds the 3rd quartile of the trade amounts for either countries. Note that this definition of connection takes into account the unequal sizes of the two countries. Under this definition, although the gross trade amount between a small country (e.g. Heiti) and a large one (e.g. the United States) may only account for a small fraction of the large, the two countries can still be considered connected if the large country is a major trade partner of the small one. After obtaining the adjacency matrix for the 58 countries,

we apply the 3-fold NS-CV to determine how many distinct communities are in the trade network. We plot the negative likelihood versus the candidate K values from 2 to 10 in Figure 4.4(a), where $K = 4$ is shown to have the lowest negative likelihood value. This concludes four distinct trading groups in the international trade network.

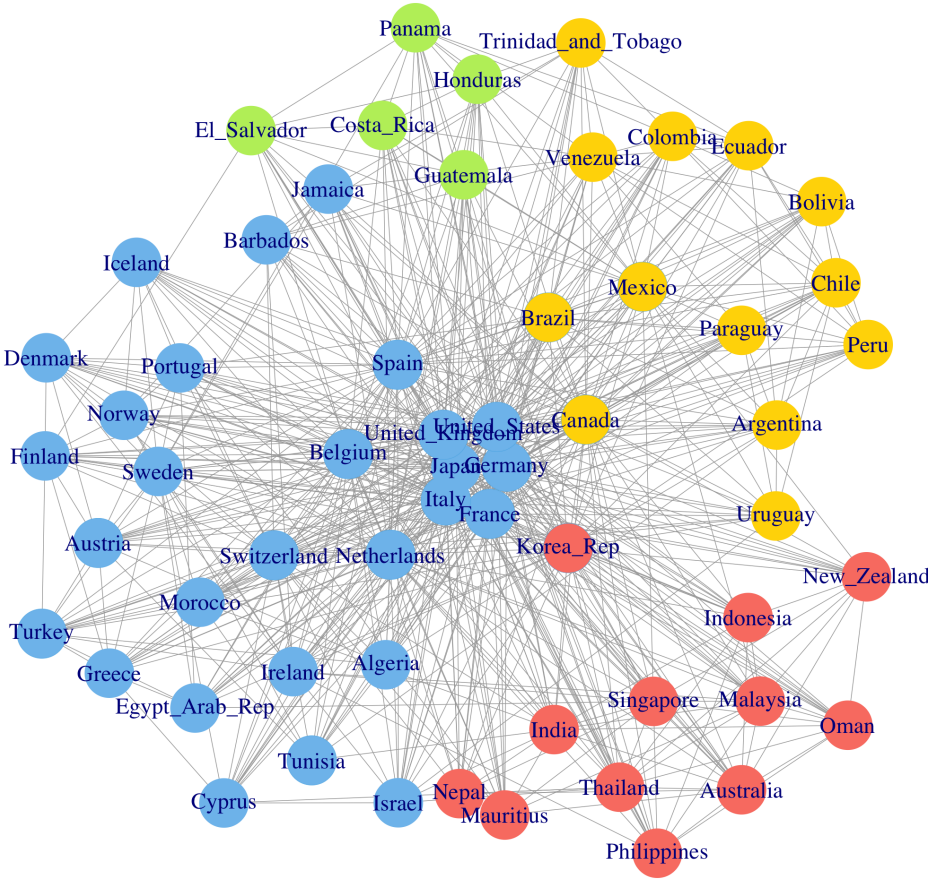
We plot the country names in the four detected trading groups by different colors and also the trade connections among the 58 countries in Figure 4.4(b). The four trading groups have a clear geographical meaning: the red group is mainly for Asian Pacific countries including Australia and New Zealand. The yellow and green groups consist of the countries in North, Central, and South America. It is interesting to see five Central American countries form a separate community in green by themselves. The largest group in the blue color are mainly for the developed countries in Europe or around the Mediterranean Sea plus the United States and Japan. The United States and Japan are not clustered by their geographical region, because as the two largest economies in the world in 2000, they had the strongest ties to the industrialized countries in Europe.

4.4.2 The U.S. Senate Network

In the second application, we apply the NS-CV method to study the political network for the U.S. Senate. Although the same method may be applied to study the U.S. House, we focuses on the U.S. Senate in this dissertation, because it has a more moderate size (100 senators) and the senators' political affiliations and the legislation records can be easily tracked and understood in the political context. The first step of analyzing the U.S. Senate under the stochastic block model is to establish an adjacency matrix that characterizes the connections between every pair of the senators. While many forms of connection such as a name reference in a speech or in a publication, an election campaign endorsement, geographical affinity, or even a casual friendship may be considered, we choose to use a more reliable and objective



(a) NS-CV Optimal K Achieved at 4



(b) Clustering of countries ($K = 4$)

Figure 4.4: NS-CV Community Detection for International Trade Network.

source of evidence, i.e., cosponsored legislation, to determine connections among the senators.

The co-sponsored legislation practice can be traced back to at least 1930 for the U.S. Senate and to 1967 for the U.S. House (Campbell, 1982). Under the current legislative process, a bill can only be introduced to the House or Senate for consideration by one member of Congress. That member is known as the bill's sponsor or patron and his/her name shall appear first on the bill when introduced. Other members of Congress may choose to add their names on the bill after the sponsor to express their support for the legislation. By doing so, they become a co-sponsor or co-patron for the bill. The sponsor and cosponsor are more likely to share a similar political view or agenda regarding the legislation and therefore the sponsor-cosponsor relationship can reflect a kind of connection between the two members. In fact, Campbell (1982) noted that in some legislation, the sponsor might spend a considerable amount of effort recruiting the co-sponsors using personal contacts.

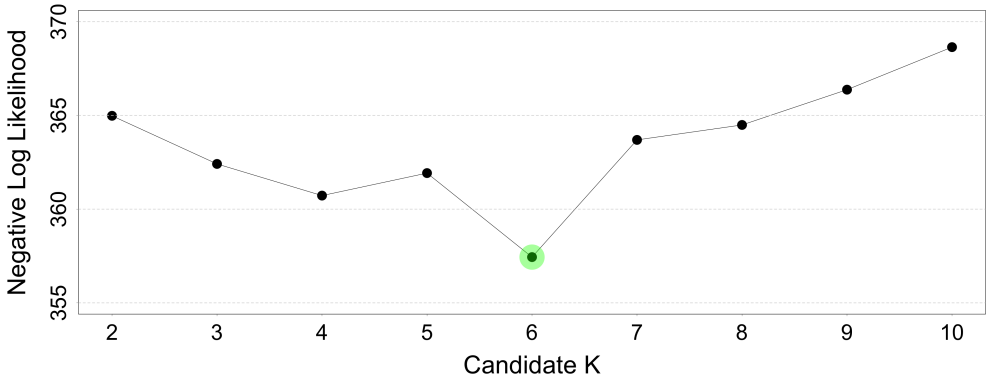
Fowler (2006) had presented a dataset for all the legislation records from 93th to 108th U.S. Congress between 1973 and 2004. Fowler used a number of statistics to study the sponsor-cosponsor relationship which led to some interesting discoveries of the underlying political network within the U.S. Congress. In the current study, we re-analyze the 108th U.S. Senate co-sponsored legislation data for the 108th U.S. Senate using the NS-CV method to determine how many small political clusters existed inside the 108th U.S. Senate.

The 108th U.S. Senate ran from January 3, 2003 to January 3, 2005, under the third and fourth years of the George W. Bush presidency. It was composed of 51 Republicans (Majority Leader: Bill Frist, TN), 48 Democrats (Minority Leader: Tom Daschle, SD), and 1 Independent (Jim Jeffords, VT) who was aligned with Democrats according to the CRS Report for Congress. Forley's data showed that a total of 3,035 Senate bills were introduced with 869 (or 28.6 percent) of them having no co-sponsors

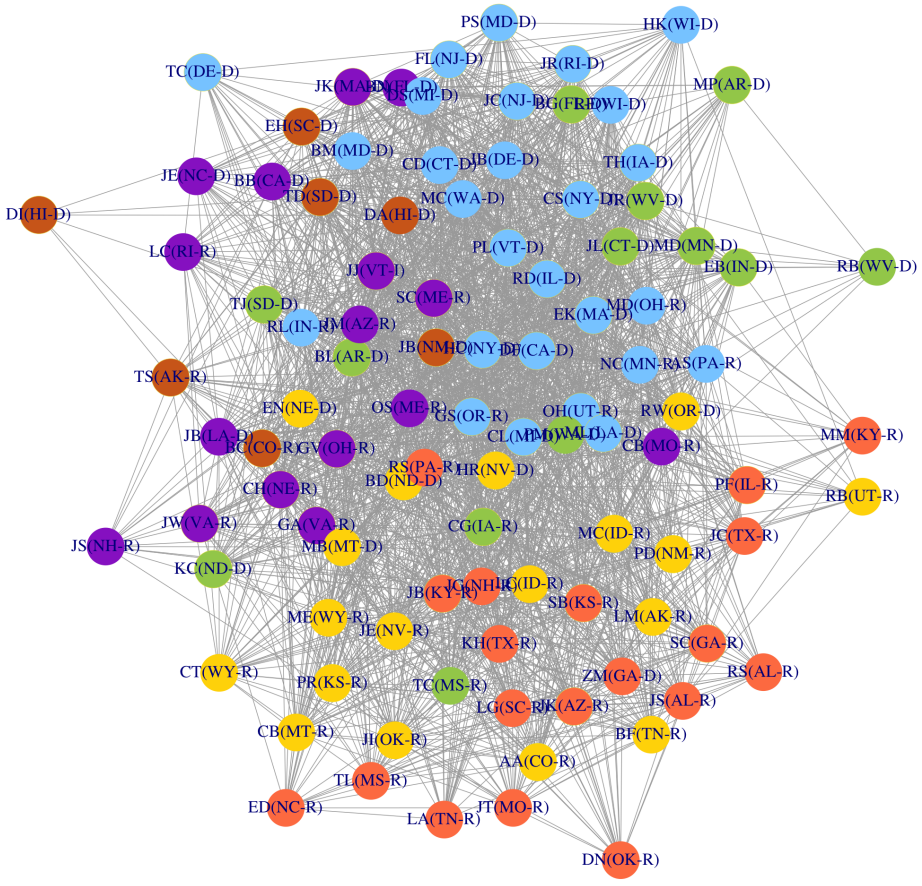
and the remaining 2,166 (or 71.4 percent) bills having at least one co-sponsor. The average number of co-sponsors is equal to 5.3. To construct an adjacency matrix, we may consider the two senators are connected if either one had co-sponsored at least one senate bill introduced by the other. However, there is one caveat for such a definition. We have noticed that several senate bills may have a very large number of co-sponsors (e.g., 81 co-sponsors for S.982, and 75 co-sponsors for S.1379). Such bills might represent a common political view shared by most of the members (e.g., anti-terrorism for S.982), but not reflect the political affiliations between the members. On the technical side, these bills may introduce too many "1"s in the adjacency matrix, making it too saturated to be clustered. For these two reasons, we exclude those bills with many co-sponsors from this analysis. In summary, in our construction of the adjacency matrix, a connection ($A_{ij} = 1$) is assigned between the two Senate members, if either member had co-sponsored at least four legislative bills introduced by the other.

We apply 3-fold NS-CV on the adjacency matrix to analyze the US Senate network and plot the negative log likelihood against the candidate K values from 2 to 10 in Figure 4.5(a). $K = 6$ is shown to achieve the lowest negative log likelihood value, indicating that the 108th U.S. Senate may consist of six separate clusters. We plot these six clusters and the senators' co-sponsorship connections in Figure 4.5(b).

To facilitate further investigation, we tabulated the senators' initials, states and their party affiliation for the six detected communities in Table 4.1. We also summarized the party composition for these 6 clusters in Table 4.2. It is shown that three of the six clusters (Cluster 3, 4 and 5) are mainly comprised of the Republicans, representing conservative caucuses in the Senate. The other three clusters (Cluster 1, 2 and 6) are liberal-oriented caucuses that primarily consist of the Democratic senators. Note that Senator Jim Jeffords from Vermont served as a Republican until 2001. After 2001, he left the party to become an independent and began caucusing with



(a) NS-CV Optimal K Achieved at 6



(b) Clustering of the 108th U.S. Senate ($K = 6$ from NS-CV)

Figure 4.5: NS-CV Community Detection for the 108th U.S. Senate Network.

the Democrat. We defer a further analysis of the composition of these six clusters to political scientists and historians.

Cluster	**Senators** Name Initials (State-Party)
Clusrer 1	JC(NJ-D) MD(OH-R) RD(IL-D) RF(WI-D) DF(CA-D) TH(IA-D) OH(UT-R) EK(MA-D) HK(WI-D) ML(LA-D) FL(NJ-D) CL(MI-D) BM(MD-D) PS(MD-D) CS(NY-D) JB(DE-D) MC(WA-D) TC(DE-D) HC(NY-D) NC(MN-R) GS(OR-R) DS(MI-D) RL(IN-R) AS(PA-R) CD(CT-D) PL(VT-D) JR(RI-D)
Cluster 2	EB(IN-D) RB(WV-D) TC(MS-R) KC(ND-D) MD(MN-D) CG(IA-R) TJ(SD-D) JL(CT-D) BL(AR-D) PM(WA-D) MP(AR-D) BG(FL-D) JR(WV-D)
Cluster 3	GA(VA-R) CB(MO-R) BB(CA-D) LC(RI-R) SC(ME-R) JE(NC-D) CH(NE-R) JJ(VT-I) JM(AZ-R) BN(FL-D) OS(ME-R) JS(NH-R) JW(VA-R) GV(OH-R) B(LA-D) JK(MA-D)
Cluster 4	AA(CO-R) CB(MT-R) MC(ID-R) PD(NM-R) BD(ND-D) JE(NV-R) ME(WY-R) JI(OK-R) LM(AK-R) EN(NE-D) PR(KS-R) CT(WY-R) RW(OR-D) RB(UT-R) LC(ID-R) MB(MT-D) BF(TN-R) HR(NV-D)
Cluster 5	LA(TN-R) SB(KS-R) JB(KY-R) SC(GA-R) JC(TX-R) ED(NC-R) PF(IL-R) LG(SC-R) KH(TX-R) JK(AZ-R) TL(MS-R) ZM(GA-D) DN(OK-R) RS(PA-R) JS(AL-R) RS(AL-R) JT(MO-R) JG(NH-R) MM(KY-R)
Cluster 6	DA(HI-D) JB(NM-D) TS(AK-R) BC(CO-R) EH(SC-D) TD(SD-D) DI(HI-D)

Table 4.1: Composition of the 6 Clusters in the 108th U.S. Senate Determined by NS-CV.

		Party			
		Democrat	Republican	Independent	Total
Cluster	1	21	6	0	27
	2	11	2	0	13
	3	5	10	1	16
	4	5	13	0	18
	5	1	18	0	19
	6	5	2	0	7
	Total	48	51	1	100

Table 4.2: Party Summary for the 6 Clusters in the 108th U.S. Senate.

Chapter 5

Edge Split Cross Validation (ES-CV) for Community Detection

The NS-CV method and the cross validation method developed by Chen and Lei (2018) have two drawbacks: (1) the nodes of the test set are clustered only based on how they connect to the training set but not based on how they connect among themselves. This might introduce some bias in the clustering of the test set and would eventually affect the log likelihood evaluation. (2) Although the node sets are randomly split to the training and test set in these two cross validation procedures, this split is not in a full randomized manner for the overall network. To see this, recall that the network connection data are represented by entries on the upper triangular part of the adjacency matrix $\mathbf{A}$. Under the node split strategy, these entries are not randomly split to a training set and a test set, but imposed to be block-wise partitioned according to (4.1). This drives us to explore a new cross validation strategy under which a random partition of the training and test sets is made for the edges (i.e., the entries of $\mathbf{A}$) instead of the nodes.

To illustrate the edge split cross validation (ES-CV) strategy, recall that $\mathcal{E} = \{(i,j) : 1 \leq i < j \leq n\}$ denotes the network's edge set that corresponds to the upper triangular part of the adjacency matrix $\mathbf{A}$. We randomly partition $\mathcal{E}$ into a training set $\mathcal{E}^{(1)}$ and a test set $\mathcal{E}^{(2)}$. Let $\mathcal{P}_{\mathcal{E}^{(1)}}(\mathbf{A})$ denote the symmetric matrix derived

from $\mathbf{A}$ that preserves the entries corresponding to the edges in $\mathcal{E}^{(1)}$, while setting the remaining entries as "being missing." This $\mathcal{P}_{\mathcal{E}^{(1)}}(\mathbf{A})$ matrix will be used as the training set for ES-CV.

An obvious challenge is to cluster the network based on this incomplete adjacency matrix $\mathcal{P}_{\mathcal{E}^{(1)}}(\mathbf{A})$. One idea is to solve a restricted version of the maximum likelihood estimation similar to (2.2) as

$$\arg\max_{\phi,\ \mathbf{P}} \sum_{\substack{1\le i<j\le n \\ (i,j)\in\mathcal{E}^{(1)}}} \left[A_{ij}\log\left(p_{\phi(i)\phi(j)}\right) + (1-A_{ij})\log\left(1-p_{\phi(i)\phi(j)}\right)\right] \tag{5.1}$$

However this restricted MLE is not simpler, if not more complicated, than the unrestricted version (2.2) which has been shown to be NP-hard (Chen et al., 2014). Moreover, it is also difficult to apply the standard Spectral Clustering method on $\mathcal{P}_{\mathcal{E}^{(1)}}(\mathbf{A})$ due to the "missing" entries. Our solution is illuminated by the recent development of the matrix completion techniques (see, e.g., Candès and Recht, 2009; Candès and Tao, 2010; Mazumder et al., 2010), which provides us a way to impute the "missing" entries so that Spectral Clustering can be performed on the imputed matrix.

5.1 ES-CV for Sparse Network

In a general setting of matrix completion, let $\mathbf{M} = (M_{ij})_{i\in[n_1],j\in[n2]}$ be an $n_1 \times n_2$ matrix whose entries are partially observed on a subset $\Omega \subset [n_1]\times[n_2]$. The observed part of $\mathbf{M}$ can be denoted by $\mathcal{P}_\Omega(\mathbf{M}) = (M_{ij})_{(i,j)\in\Omega}$. The matrix completion problem aims to recover $\mathbf{M}$ from $\mathcal{P}_\Omega(\mathbf{M})$, provided that $\mathbf{M}$ has a lower rank than $\min(n1, n2)$.

The algorithm to recover matrix $\mathbf{M}$ is to solve the convex optimization problem:

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{Z}\|_* \\ \text{subject to} \quad & \sum_{(i,j)\in\Omega} (Z_{ij} - M_{ij})^2 \leq \delta \end{aligned} \tag{5.2}$$

where $\|\cdot\|_*$ is the *nuclear norm* of a matrix that is equal to the sum of its singular values. Candès and Recht had proved that under certain technical conditions for the cardinality of Ω, the rank of $\mathbf{M}$, and $n = max(n_1, n_2)$, the minimizer of problem (5.2) will converge to M in probability (Theorem 1 of Candès and Recht (2009)). Practically, the optimization problem (5.2) can be solved efficiently using several good algorithms, such as Soft-Impute (Mazumder et al., 2010), Alternating Minimization (Jain et al., 2013), Singular Value Thresholding (Cai et al., 2010), or the Augmented Lagrange Multipliers Method (Lin et al., 2010).

In the ES-CV application, we can first apply the matrix completion method to impute those "missing" entries in $\mathcal{P}_{\mathcal{E}^{(1)}}(\mathbf{A})$. Note that since the Stochastic Block Model indicates that the adjacency matrix $\mathbf{A}$ is generated by members belonging to K distinct communities where $K < n$, it can be assumed that the adjacency matrix $\mathbf{A}$ should have a lower rank than n. Thus although a portion of $\mathbf{A}$ is taken out as the test set, a version of $\mathbf{A}$ can be recovered by solving the matrix completion problem with a restriction on symmetry:

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{Z}\|_* \\ \text{subject to} \quad & \sum_{(i,j)\in\mathcal{E}^{(1)}} (Z_{ij} - A_{ij})^2 \leq \delta, \\ & \mathbf{Z} \text{ is symmetric.} \end{aligned} \tag{5.3}$$

Note that there is no guarantee that the solution $\mathbf{Z}$ of (5.3) is an adjacency matrix (i.e., a symmetric matrix with 0 or 1 entries) aside from being symmetric. In fact,

some of the recovered entries of $\mathbf{Z}$ may even be negative. As a result, we cannot directly use the standard Spectral Clustering algorithm on $\mathbf{Z}$. However, because most of the matrix completion methods (e.g. SoftImpute, Singular Value Thresholding) will output the recovered matrix $\mathbf{Z}$ in its Singular Value Decomposition (SVD) form, we may simply apply the SVD-based Spectral Clustering method by running the K-mean algorithm on the K left-singular vectors that correspond to the K largest singular values of the recovered matrix $\mathbf{Z}$ (Drineas et al., 2004; Jia, 2013). This will give us an estimated community labeling function ϕ_K for the nodes in the network. Afterwards, the matrix $\mathbf{P}$ for the within and between community connection probabilities can be easily estimated as:

$$\hat{p}_{uv} = \begin{cases} \left(\sum_{\substack{(i,j)\in\mathcal{E}^{(1)} \\ \phi_K(i)=\phi_K(j)=u}} A_{ij} \right) \Big/ \mid \{(i,j) \in \mathcal{E}^{(1)} : \phi(i) = \phi(j) = u\} \mid, & \text{if } u = v \\ \left(\sum_{\substack{(i,j)\in\mathcal{E}^{(1)} \\ \phi_K(i)=u,\ \phi_K(j)=v}} A_{ij} \right) \Big/ \mid \{(i,j) \in \mathcal{E}^{(1)} : \phi(i) = u,\ \phi(j) = v\} \mid, & \text{if } u \neq v. \end{cases} \tag{5.4}$$

With the estimated ϕ and $\hat{p}_{uv}$, the validation can be simply run on the test set $\mathcal{E}^{(2)}$ by calculating the log likelihood function as:

$$\ell\left(\{\hat{p}_{uv}\}_{u,v=1}^{K} \,\middle|\, \phi_K \right) = \sum_{(i,j)\in\mathcal{E}^{(2)}} \left[A_{ij} \log\left(\hat{p}_{\phi_K(i)\phi_K(j)} \right) + (1 - A_{ij}) \log\left(1 - \hat{p}_{\phi_K(i)\phi_K(j)} \right) \right]. \tag{5.5}$$

Over a set of candidate $K \in \{2, \ldots, K_{\max}\}$, the one that achieves the maximum log

likelihood value is chosen by ES-CV:

$$K_{\text{ES-CV}} = \underset{2\le K\le K_{\max}}{\arg\max}\ \ell\left(\{\hat{p}_{uv}\}_{u,v=1}^{K}\,\middle|\,\phi_K\right). \tag{5.6}$$

The above steps can be refined through an F-fold cross validation strategy to improve the estimate by averaging the log likelihood function over F iterations. We present below a formal version of this algorithm.

Algorithm 2 Edge Split Cross Validation (ES-CV) by Matrix Completion

Input: The adjacency matrix $\mathbf{A}$, $K_{\max} \ge 2$, $F \ge 2$, matrix completion threshold δ.
Output: The optimal community number $K_{\text{ES-CV}}$.

1: Randomly partition the edge set $\mathcal{E} = \{(i,j) : 1 \le i < j \le n\}$ into F subsets with nearly equal sizes, denoted as $\mathcal{E}^{(1)}, \mathcal{E}^{(2)}, \ldots, \mathcal{E}^{(F)}$.
2: **for** $f = 1$ to F **do**
3: Solve (Mazumder et al., 2010; Cai et al., 2010; Lin et al., 2010)

$$\begin{aligned} &\text{minimize} && \|\mathbf{Z}\|_* \\ &\text{subject to} && \sum_{(i,j)\in\mathcal{E}\setminus\mathcal{E}^{(f)}} (Z_{ij} - A_{ij})^2 \le \delta, \\ & && \mathbf{Z} \text{ is symmetric.} \end{aligned}$$

4: Denote SVD for the solution $\mathbf{Z}$ as $\mathbf{U}_f$, $\mathbf{S}_f$, and $\mathbf{V}_f$.
5: **for** $K = 2$ to K_{max} **do**
6: Extract the K singular vectors from $\mathbf{U}_f$ corresponding to the K largest singular values.
7: Perform the K-mean algorithm on the K singular vectors to cluster the network into K communities.
8: Estimate the within-community and between-community connection probabilities $\hat{p}_{uv}$ by (5.4).
9: Compute the log likelihood function $\ell^{(f)}\left(\{\hat{p}_{uv}\}_{u,v=1}^{K}\,\middle|\,\phi_K\right)$ by (5.5).
10: **end for**
11: **end for**
12: **return** $K_{\text{ES-CV}} = \underset{2\le K\le K_{\max}}{\arg\max} \sum_{f=1}^{F} \ell^{(f)}\left(\{\hat{p}_{uv}\}_{u,v=1}^{K}\,\middle|\,\phi_K\right)/F$.

5.2 ES-CV for General Network

Algorithm 2 is oriented to recover a sparse network that has a low-rank adjacency matrix. In a general situation, we may develop a more robust edge split cross validation algorithm based on an improved convex optimization program created by Vinayak et al. (2014):

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{L}\|_* + \lambda||\mathbf{S}||_1 \\ \text{subject to} \quad & 1 \geq L_{ij} \geq S_{ij} \geq 0 \ \text{ for } \ \text{all } i,j \in \{1,2,\cdots,n\}, \\ & L_{ij} = S_{ij} \ \text{ whenever } A_{ij} = 0 \ \text{ for } \ (i,j) \in \mathcal{E}^{(1)}, \\ & \sum_{i,j=1}^{n} L_{ij} \geq |\mathcal{R}| = \sum_{k=1}^{K^*} n_i^2 \end{aligned} \tag{5.7}$$

where $\lambda > 0$ is a regularization parameters and $\mathbf{L}$ is a low-rank matrix that corresponds to the true community structure of the network. In the ideal case, the solution $\mathbf{L}^0$ would be a block diagonal matrix where the entries for within-community connections are all equal to 1, and the entries for between-community connection are all equal to 0. $\mathbf{S}$ is an error matrix whose entries are equal 1 for a pair of nodes in the training set $\mathcal{E}^{(1)}$ that are from the same community but not connected (i.e. $A_{ij} = 0$). The rest of the entries of $\mathbf{S}$ are set to zero. Note that the restriction in the above program seeks an approximation of $\mathbf{A}$ by $\mathbf{L}-\mathbf{S}$ on $\mathcal{E}^{(1)}$. The parameter $|\mathcal{R}|$ is the sum of squares for the communities' sizes, or the total number of within-communities pairs of nodes in the network. In practice, the value of $|\mathcal{R}|$ is generally unknown. But according to Vinayak et al. (2014), we may solve the program (5.7) by a trial-and-error of several $|\mathcal{R}|$ values until a desirable solution is obtained.

After a solution $\mathbf{L}^0$ for the program (5.7) is obtained, a simple K-mean method can be used on $\mathbf{L}^0$ to cluster the network into K communities. Then the within-community and between-community connection probability can be estimated by (5.4). The final

validation can run on the test set $\mathcal{E}^{(2)}$ using the log likelihood function given by (5.5). In summary, we present below a formal algorithm for an F-fold ES-CV using the Improved Convex Optimization.

Algorithm 3 Edge Split Cross Validation by Improved Convex Optimization

Input: The adjacency matrix $\mathbf{A}$, $K_{\max} \geq 2$, $F \geq 2$, $|\mathcal{R}|$, and regularization parameter $\lambda > 0$.

Output: The optimal community number $K_{\text{ES-CV}}$.

1: Randomly split the edge index set $\mathcal{E} = \{(i,j) : 1 \leq i < j \leq n\}$ into F subsets with nearly equal sizes, denoted as $\mathcal{E}^{(1)}, \mathcal{E}^{(2)}, \ldots, \mathcal{E}^{(F)}$.

2: **for** $f = 1$ to F **do**

3: Solve (Vinayak et al., 2014)

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{L}\|_* + \lambda ||\mathbf{S}||_1 \\ \text{subject to} \quad & 1 \geq L_{ij} \geq S_{ij} \geq 0 \ \text{ for all } i,j \in \{1,2,\cdots,n\}, \\ & L_{ij} = S_{ij} \ \text{ whenever } A_{ij} = 0 \text{ and } (i,j) \notin \mathcal{E}^{(f)} \\ & \sum_{i,j=1}^{n} L_{ij} \geq \mathcal{R} \end{aligned}$$

4: Denote the solution as $\mathbf{L}_f$.

5: **for** $K = 2$ to K_{max} **do**

6: Perform the K-mean algorithm on the rows of $\mathbf{L}_f$ to cluster the network into K communities.

7: Estimate the within-community and between-community connection probabilities $\hat{p}_{uv}$ by (5.4).

8: Compute the log likelihood function $\ell^{(f)}\left(\{\hat{p}_{uv}\}_{u,v=1}^{K} \,\middle|\, \phi_K\right)$ by (5.5).

9: **end for**

10: **end for**

11: **return** $K_{\text{ES-CV}} = \underset{2 \leq K \leq K_{\max}}{\arg\max} \sum_{f=1}^{F} \ell^{(f)}\left(\{\hat{p}_{uv}\}_{u,v=1}^{K} \,\middle|\, \phi_K\right) / F$.

5.3 Consistency of ES-CV

In this section, we derive consistency for the ES-CV method for community detection under a similar framework as we studied NS-CV in Section 4.2. We will leverage some notations and lemmas proved in Section 4.2, while introducing others specific to ES-CV.

5.3.1 Basic Notations and Assumptions

Consider an undirected network $(\mathcal{N}, \mathcal{E})$ of size n that is generated from the Stochastic Block Model with K^* disjoint communities of sizes $\{n_k\}_{k=1}^{K^*}$, where $2 \leq K^* \leq K_{\max}$ for some positive integer $K_{\max}$. Let ϕ^* denote the true community labeling function that maps every member of the network to its belonging community. For simplicity, we assume the within-community and between-community connection probabilities are equal to p and q respectively with $0 < q < p < 1$.

Suppose $\mathbf{A}$ is an observed $n \times n$ adjacency matrix for the network. According to the ES-CV procedure illustrated in Section 5.1, we assume every edge $(i, j) \in \mathcal{E} = \{(i, j) : 1 \leq i < j \leq n\}$ is split either to the training set $\mathcal{E}^{(1)}$ with a probability $r \in (0, 1)$, or to the test set $\mathcal{E}^{(2)}$ with a probability $1 - r$. This generates the random partition for $\mathcal{E} = \mathcal{E}^{(1)} \cup \mathcal{E}^{(2)}$. For the adjacency matrix $\mathbf{A}$, we accordingly preserve the values for the entries A_{ij} and A_{ji} where $(i, j) \in \mathcal{E}^{(1)}$, while setting the remaining A_{ij} and A_{ji} values to be "missing" (or "NA") for $(i, j) \in \mathcal{E}^{(2)}$. We use the same notation $\mathcal{P}_{\mathcal{E}^{(1)}}(\mathbf{A})$ as before to denote the resultant incomplete adjacency matrix.

For every candidate community number K, we apply a matrix completion program, such as (5.3) or (5.7), on $\mathcal{P}_{\mathcal{E}^{(1)}}(\mathbf{A})$ to obtain an estimated community labeling function ϕ_K. Next define

$$W_{\phi_K} := \{(i, j) \in \mathcal{E} : \phi_K(i) = \phi_K(j)\}, \tag{5.8}$$

$$B_{\phi_K} := \{(i,j) \in \mathcal{E} : \phi_K(i) \neq \phi_K(j)\}; \tag{5.9}$$

and the four subsets

$$W^{(1)}_{\phi_K} := \{(i,j) \in \mathcal{E}^{(1)} : \phi_K(i) = \phi_K(j)\}, \tag{5.10}$$

$$W^{(2)}_{\phi_K} := \{(i,j) \in \mathcal{E}^{(2)} : \phi_K(i) = \phi_K(j)\}, \tag{5.11}$$

$$B^{(1)}_{\phi_K} := \{(i,j) \in \mathcal{E}^{(1)} : \phi_K(i) \neq \phi_K(j)\}; \tag{5.12}$$

$$B^{(2)}_{\phi_K} := \{(i,j) \in \mathcal{E}^{(2)} : \phi_K(i) \neq \phi_K(j)\}; \tag{5.13}$$

Then the within and between community connection probabilities p and q are estimated as

$$\hat{p} = \frac{\sum_{(i,j)\in W^{(1)}_{\phi_K}} A_{ij}}{|W^{(1)}_{\phi_K}|}, \qquad \hat{q} = \frac{\sum_{(i,j)\in B^{(1)}_{\phi_K}} A_{ij}}{|B^{(1)}_{\phi_K}|}. \tag{5.14}$$

Note that the equation (5.14) is a simplified version of (5.4), because we assume the within-community connection probabilities are all equal to p, and the between-community connection probabilities are all equal to q.

Finally, the log likelihood function for $\hat{p}$ and $\hat{q}$ given ϕ_K is evaluated on the test set $\mathcal{E}^{(2)}$ as

$$\begin{aligned} \ell\left(\hat{p}, \hat{q} \mid \phi_K\right) &= \sum_{(i,j)\in W^{(2)}_{\phi_K}} \Big[A_{ij} \log \hat{p} + (1 - A_{ij}) \log \hat{p}\Big] \\ &\quad + \sum_{(i,j)\in B^{(2)}_{\phi_K}} \Big[A_{ij} \log \hat{q} + (1 - A_{ij}) \log \hat{q}\Big]. \end{aligned} \tag{5.15}$$

To refine this log likelihood function, we further define

$$T^{(2)}_{11} := \{(i,j) \in \mathcal{E}^{(2)} : \phi_K(i) = \phi_K(j),\ \phi^*(i) = \phi^*(j)\} = W^{(2)}_{\phi_K} \cap W^{(2)}_{\phi^*};$$

$$
\begin{aligned}
T_{12}^{(2)} &:= \{(i,j) \in \mathcal{E}^{(2)} : \phi_K(i) \neq \phi_K(j),\ \phi^*(i) = \phi^*(j)\} = B_{\phi_K}^{(2)} \cap W_{\phi^*}^{(2)};\\
T_{21}^{(2)} &:= \{(i,j) \in \mathcal{E}^{(2)} : \phi_K(i) = \phi_K(j),\ \phi^*(i) \neq \phi^*(j)\} = W_{\phi_K}^{(2)} \cap B_{\phi^*}^{(2)};\\
T_{22}^{(2)} &:= \{(i,j) \in \mathcal{E}^{(2)} : \phi_K(i) \neq \phi_K(j),\ \phi^*(i) \neq \phi^*(j)\} = B_{\phi_K}^{(2)} \cap B_{\phi^*}^{(2)}. \quad (5.16)
\end{aligned}
$$

$T_{11}^{(2)}$ is a subset of $\mathcal{E}_2$ that consists of all pairs of members who are from the same community (under ϕ^*), and are correctly assigned to the same community under ϕ_K. $T_{12}^{(2)}$ is the complement set of $T_{11}^{(2)}$ for the pairs of members from the same community (under ϕ^*), but being wrongly clustered to different communities under ϕ_K. Similarly, $T_{21}^{(2)}$ and $T_{22}^{(2)}$ are for the pairs of members from different communities (under ϕ^*) but assigned to the same or different communities under ϕ_K respectively. Under these definitions, the log likelihood function (5.15) can be re-written as

$$
\begin{aligned}
\ell\left(\hat{p}, \hat{q} \mid \phi_K\right) &= \sum_{T_{11}^{(2)}} \Big[A_{ij} \log \hat{p} + (1 - A_{ij}) \log(1 - \hat{p})\Big]\\
&\quad + \sum_{T_{12}^{(2)}} \Big[A_{ij} \log \hat{q} + (1 - A_{ij}) \log(1 - \hat{q})\Big]\\
&\quad + \sum_{T_{21}^{(2)}} \Big[A_{ij} \log \hat{p} + (1 - A_{ij}) \log(1 - \hat{p})\Big]\\
&\quad + \sum_{T_{22}^{(2)}} \Big[A_{ij} \log \hat{q} + (1 - A_{ij}) \log(1 - \hat{q})\Big]. \quad (5.17)
\end{aligned}
$$

The conditional expectation for this log likelihood function given $\hat{\phi}$, $\hat{p}$, and $\hat{q}$ has a form

$$
\begin{aligned}
\xi_K &:= E\Big[\ell\left(\hat{p}, \hat{q} \mid \phi_K\right)\Big]\\
&= \sum_{T_{11}^{(2)}} \Big[p \log \hat{p} + (1 - p) \log(1 - \hat{p})\Big] + \sum_{T_{12}^{(2)}} \Big[p \log \hat{q} + (1 - p) \log(1 - \hat{q})\Big]\\
&\quad + \sum_{T_{21}^{(2)}} \Big[q \log \hat{p} + (1 - q) \log(1 - \hat{p})\Big] + \sum_{T_{22}^{(2)}} \Big[q \log \hat{q} + (1 - q) \log(1 - \hat{q})\Big]
\end{aligned}
$$

$$= \left|T_{11}^{(2)}\right|\Big[p\log\hat{p}+(1-p)\log(1-\hat{p})\Big]+\left|T_{12}^{(2)}\right|\Big[p\log\hat{q}+(1-p)\log(1-\hat{q})\Big] + \left|T_{21}^{(2)}\right|\Big[q\log\hat{p}+(1-q)\log(1-\hat{p})\Big]+\left|T_{22}^{(2)}\right|\Big[q\log\hat{q}+(1-q)\log(1-\hat{q})\Big]. \tag{5.18}$$

5.3.2 Supporting Lemmas

In this section, we introduce several supporting lemmas that will help prove the main consistency theorem for ES-CV. We start with a lemma that shows in ES-CV the selected edges in the training set $\mathcal{E}^{(1)}$ nearly covers all nodes in $\mathcal{N}$.

Lemma 5.3.1. *Let $\mathcal{N}^{(1)} := \{v \in \mathcal{N} : (i,v) \in \mathcal{E}^{(1)}$ or $(j,v) \in \mathcal{E}^{(1)}$ for some $i,j \in \mathcal{N}\}$ be a subset of $\mathcal{N}$ whose elements correspond to some edges in $\mathcal{E}^{(1)}$. Then,*

$$Pr\big(|\mathcal{N}_1| = n\big) \geq 1 - n(1-r)^{n-1}. \tag{5.19}$$

Proof. Since every edge has a probability r be chosen in $\mathcal{E}^{(1)}$, then for any $v \in \mathcal{N}$,

$$Pr\big(v \notin \mathcal{N}^{(1)}\big) = (1-r)^{n-1}.$$

Thus,

$$\begin{aligned} Pr\big(|\mathcal{N}^{(1)}| = n\big) &= 1 - Pr\big(\cup_{v=1}^{n}\{v \notin \mathcal{N}^{(1)}\}\big) \\ &\geq 1 - nPr\big(v \notin \mathcal{N}^{(1)}\big) \\ &= 1 - n(1-r)^{n-1}. \end{aligned}$$

□

Lemma 5.3.2. *In ES-CV, the cardinality of the test set $\mathcal{E}^{(2)}$ satisfies*

$$Pr\Big(\big|\mathcal{E}^{(2)}\big| \geq n^2(1-r)\Big) \leq \exp\Big(-\frac{n(n-1)(1-r)}{6}\Big). \tag{5.20}$$

Proof. Because every edge is chosen to the test set $\mathcal{E}^{(2)}$ with probability $1-r$ in ES-CV, thus $|\mathcal{E}^{(2)}| \sim \text{Binomial}\big(n(n-1)/2, 1-r\big)$. Hence, by the large deviation inequality for the Binomial distribution (e.g. Equation 1.6 of Janson 2016),

$$\begin{aligned} Pr\Big(|\mathcal{E}^{(2)}| \geq n^2(1-r)\Big) &\leq Pr\Big(|\mathcal{E}^{(2)}| \geq n(n-1)(1-r)\Big) \\ &\leq \exp\Big(-\frac{n(n-1)(1-r)}{6}\Big). \end{aligned}$$

□

Recall Definition 4.2.3 for networks with a balanced community structure. Next we introduce the following three conditions that are similar to those we presented for NS-CV in Section 4.2.2.

Condition 1. The true community structure (under ϕ^*) of a network and the recovered community structure (under ϕ_K) from ES-CV for $K \in \{2, \ldots, K_{\max}\}$ are balanced with respect to a constant $\pi_0 \in (0,1)$.

Condition 2. The within and between community probability p and q satisfy $0 < \delta < q < p < 1-\delta < 1$ for a small positive δ.

Condition 3. When the candidate community number $K = K^*$, the matrix completion program used in ES-CV gives an consistent estimate ϕ_K on $\mathcal{P}_{\mathcal{E}^{(1)}}(\mathbf{A})$ so that

$$Pr\left(\phi_{K^*} \neq \phi^*\right) \leq h(nr), \tag{5.21}$$

where n is the network size, r is the probability for an edge being assigned to the training set $\mathcal{E}^{(1)}$, and $h(x)$ is a positive function such that $h(x) \to 0$ as $x \to \infty$.

Note that Condition 3 has been proved for some matrix completion programs under certain technical conditions, for example, Theorem 2 of Vinayak et al. 2014

for Algorithm 3. Under Condition 3, our next lemma states that when $K = K^*$, the estimated $\hat{p}$ and $\hat{q}$ by (5.14) converge to the true p and q in probability.

Lemma 5.3.3. *Under Condition 1 and 3 , when the candidate community number $K = K^*$, the estimated $\hat{p}$ and $\hat{q}$ by (5.14) satisfy*

$$Pr\left(|\hat{p} - p| > \frac{\sqrt{\log n}}{n\sqrt{r}}\right) \leq \exp\left(-c_1 n^2 r\right) + 2n^{-c_2} + h(nr), \tag{5.22}$$

$$Pr\left(|\hat{q} - q| > \frac{\sqrt{\log n}}{n\sqrt{r}}\right) \leq \exp\left(-c_3 n^2 r\right) + 2n^{-c_4} + h(nr), \tag{5.23}$$

where c_1, c_2, c_3, c_4 are positive constants dependent on K^* and π_0 from Condition 1, and the function $h(\cdot)$ is from Condition 3.

Proof. First write

$$\tilde{p} = \frac{\sum_{(i,j)\in W^{(1)}_{\phi^*}} A_{ij}}{|W^{(1)}_{\phi^*}|}, \quad \tilde{q} = \frac{\sum_{(i,j)\in B^{(1)}_{\phi^*}} A_{ij}}{|B^{(1)}_{\phi^*}|}, \tag{5.24}$$

where $W^{(1)}_{\phi^*}$ and $B^{(1)}_{\phi^*}$ are defined in (5.10) and (5.12) for the true community labeling function ϕ^* respectively. Because in ES-CV every edge follow a probability r to be selected in the training set $\mathcal{E}^{(1)}$, then given $|W_{\phi^*}| = z$,

$$|W^{(1)}_{\phi^*}| \sim \text{Binomial}\big(z,\ r\big) \tag{5.25}$$

Recall that Lemma 4.2.2 states

$$|W_{\phi^*}| \geq \frac{n^2 - nK^*}{2K^*}. \tag{5.26}$$

Thus by the large deviation inequality (e.g. Equation 1.9 of Janson 2016) for the

Binomial distribution, we have

$$\begin{aligned}
& Pr\left(\left|W_{\phi^*}^{(1)}\right| < \frac{(n^2 - nK^*)\, r}{4K^*}\right) \\
\leq\ & Pr\left(\left|W_{\phi^*}^{(1)}\right| \leq \frac{|W_{\phi^*}| r}{2}\right) \\
\leq\ & \sum_{z=\frac{n^2-nK^*}{2K^*}}^{\frac{n(n-1)}{2}} Pr\Big(\left|W_{\phi^*}^{(1)}\right| \leq \frac{zr}{2} \ \Big|\ |W_{\phi^*}| = z\Big) \cdot Pr\Big(|W_{\phi^*}| = z\Big) \\
\leq\ & \sum_{z=\frac{n^2-nK^*}{2K^*}}^{\frac{n(n-1)}{2}} \exp\Big(-\frac{zr}{8}\Big) \cdot Pr\Big(|W_{\phi^*}| = z\Big) \\
\leq\ & \exp\Big(-\frac{(n^2 - nK^*)r}{16K^*}\Big) \\
\leq\ & \exp\left(-c_1 n^2 r\right), \qquad (5.27)
\end{aligned}$$

for some constant $c_1 > 0$. Next let $\epsilon = \sqrt{\log n}/(n\sqrt{r})$. Using the law of total probability and the Hoeffding Inequality (Hoeffding, 1963), we obtain

$$\begin{aligned}
Pr\Big(|\tilde{p} - p| > \epsilon\Big) \ =\ & \sum_{t=0}^{\frac{n(n-1)}{2}} Pr\Big(|\tilde{p} - p| > \epsilon \ \Big|\ \left|W_{\phi^*}^{(1)}\right| = t\Big) \cdot Pr\Big(\left|W_{\phi^*}^{(1)}\right| = t\Big) \\
\leq\ & \sum_{t=\frac{(n^2-nK^*)r}{4K^*}}^{\frac{n(n-1)}{2}} Pr\Big(|\tilde{p} - p| > \epsilon \ \Big|\ \left|W_{\phi^*}^{(1)}\right| = t\Big) \cdot Pr\Big(\left|W_{\phi^*}^{(1)}\right| = t\Big) \\
& + Pr\Big(\left|W_{\phi^*}^{(1)}\right| < \frac{(n^2 - nK^*)r}{4K^*}\Big) \\
\leq\ & \sum_{t=\frac{(n^2-nK^*)r}{4K^*}}^{\frac{n(n-1)}{2}} 2\exp\Big(-2t\epsilon^2\Big) \cdot Pr\Big(\left|W_{\phi^*}^{(1)}\right| = t\Big) + \exp\Big(-d_1 n^2 r\Big) \\
\leq\ & 2\exp\Big(-2 \cdot \frac{(n^2 - nK^*)r}{4K^*} \cdot \frac{\log n}{n^2 r}\Big) + \exp\Big(-c_1 n^2 r\Big) \\
\leq\ & 2n^{-c_2} + \exp\left(-c_1 n^2 r\right), \qquad (5.28)
\end{aligned}$$

for some constant $d_2 > 0$. Therefore,

$$\begin{aligned} & Pr\Big(|\hat{p} - p| > \epsilon\Big) \\ = \; & Pr\Big(\big\{|\hat{p} - p| > \epsilon\big\} \cap \big\{\phi_K \neq \phi^*\big\}\Big) + Pr\Big(\big\{|\hat{p} - p| > \epsilon\big\} \cap \big\{\phi_K = \phi^*\big\}\Big) \\ \leq \; & Pr\Big(\phi_K \neq \phi^*\Big) + Pr\Big(|\tilde{p} - p| > \epsilon\Big) \\ \leq \; & h(nr) + 2n^{-c_2} + \exp\big(-c_1 n^2 r\big) \end{aligned} \tag{5.29}$$

The concentration inequality (5.23) for $\hat{q}$ can be proved in a similar way using the lower bound for $B_{\phi_{K^*}}$ given by Lemma 4.2.2 under Condition 1. Note that the positive constants c_3 and c_4 in (5.23) will be dependent on K^* and π_0. □

Lemma 5.3.3 applies to the case where $K = K^*$. When the candidate K is either greater or less than the true community number K^*, Lemma 4.2.7 and Lemma 4.2.8 still hold true for ES-CV under Condition 1. We rephrase Corollary 4.2.9 below for the cardinality of $T_{11}^{(2)}$, $T_{12}^{(2)}$, $T_{21}^{(2)}$, $T_{22}^{(2)}$ as defined by (5.16) for ES-CV with the its edge splitting probability r.

Corollary 5.3.4. *Under Condition 1, for any $K^+ > K^*$, there exist positive constants C_{12} and C_{22} dependent on K^+, K^*, and π_0 such that*

$$Pr\left(|T_{12}^{(2)}| \geq \frac{n^2(1-r)\gamma_1\pi_0^2}{2K^{*2}}\right) \geq 1 - \exp\Big(-C_{12}n^2(1-r)\Big), \tag{5.30}$$

$$Pr\left(|T_{22}^{(2)}| \geq \frac{n^2(1-r)\gamma_2\pi_0^2}{2K^{*2}}\right) \geq 1 - \exp\Big(-C_{22}n^2(1-r)\Big). \tag{5.31}$$

For any $K^- < K^*$, there exists positive constants C_{21} and C_{11} dependent on K^-, K^*, and π_0 such that

$$Pr\left(|T_{21}^{(2)}| \geq \frac{n^2(1-r)\nu_1\pi_0^2}{2K^{*2}}\right) \geq 1 - \exp\Big(-C_{21}n^2(1-r)\Big), \tag{5.32}$$

$$Pr\left(|T_{11}^{(2)}| \geq \frac{n^2(1-r)\nu_2\pi_0^2}{2K^{*2}}\right) \geq 1 - \exp\Big(-C_{11}n^2(1-r)\Big). \tag{5.33}$$

Proof. Because the ES-CV edge splitting follows a Binomial sampling with probability $1-r$ for the test set $\mathcal{E}^{(2)}$, it is easy to see that given $|T_{12}| = z$,

$$|T_{12}^{(2)}| \;\sim\; \text{Binomial}\,(z,\ 1-r)\,.$$

Thus by the large deviation inequality for the Binomial distribution (e.g. Equation 1.9 of Janson 2016) for the Binomial distribution, we have,

$$\begin{aligned}
& Pr\left(|T_{12}^{(2)}| < \frac{\gamma_1 \pi_0^2 n^2 (1-r)}{2K^{*2}}\right) \\
\leq\ & Pr\left(|T_{12}^{(2)}| < \frac{|T_{12}|(1-r)}{2}\right) \\
\leq\ & \sum_{z=(\gamma_1\pi_0^2 n^2)/K^{*2}}^{n(n-1)/2} Pr\left(|T_{12}^{(2)}| < \frac{z(1-r)}{2} \;\middle|\; |T_{12}| = z\right) \cdot Pr\Big(|T_{12}| = z\Big) \\
\leq\ & \sum_{z=(\gamma_1\pi_0^2 n^2)/K^{*2}}^{n(n-1)/2} \exp\left(\frac{z(1-r)}{8}\right) \cdot Pr\Big(|T_{12}| = z\Big) \\
\leq\ & \exp\left(-\frac{\gamma_1 \pi_0^2 n^2 (1-r)}{8K^{*2}}\right) \\
\leq\ & \exp\Big(-C_{12} n^2 (1-r)\Big) \qquad (5.34)
\end{aligned}$$

for some positive constant C_{12} dependent on K^+, K^*, and π_0 (Notice: γ_1 in Lemma 4.2.7 depends on K^+, so does C_{12}). The other three inequalities (5.31), (5.32), and (5.33) can be proved in a similar way. □

5.3.3 Main Consistency Theorem

We now present a consistency theorem for ES-CV under a general setting where the within and between community probability p, q, and the edge training splitting probability r are all considered dependent on the network size n. We begin with two assumptions that are needed for the main consistency theorem.

Assumption 5.3.5. *When the community labeling function ϕ_{K^+} is derived from ES-CV with a candidate $K^+ > K^*$, the estimated between-community probability $\hat{q}$ by equation (5.14) shall diverge from the true parameter q so that*

$$Pr(\hat{q} - q > \delta^+) \geq 1 - \exp\left(-C_{K^+} n^2 r \left(p - q\right)^2\right), \tag{5.35}$$

where $\delta^+ = C_{K^+}(p-q)/K^* > 0$ and C_{K^+} is a positive constant.

Assumption 5.3.6. *When the community labeling function $\hat{\phi}$ is derived from ES-CV with a candidate $K^- < K^*$, the estimated within-community probability $\hat{p}$ from equation (5.14) shall diverge from the true parameter p so that*

$$Pr(\hat{p} - p < -\delta^-) \geq 1 - \exp\left(-C_{K^-} n^2 r \left(p - q\right)^2\right), \tag{5.36}$$

where $\delta^- = C_{K^-}(p-q)/K^* > 0$ and C_{K^-} is a positive constant.

Theorem 5.3.7 (Consistency of ES-CV)**.** *Consider a network $(\mathcal{N}, \mathcal{E})$ generated from a stochastic block model with a true community number K^* satisfying $2 \leq K^* \leq K_{max}$ for some positive integer K_{max}. During the ES-CV procedure, every (i, j) with $1 \leq i < j \leq n$ is split to the training set $\mathcal{E}_1$ with a probability $r \in (0, 1)$ and to the test set with a probability $1 - r$. Under Assumptions 5.3.5, 5.3.6 and Condition 1, 2, and 3, if the following requirements are met:*

(i) $r = \omega(\sqrt{\log n}/n)$ *and* $1 - r = \omega(\sqrt{\log n}/n^2)$,

(ii) $r(p-q)^2 = \omega\left(\log n/n^2\right)$,

then the optimal community number obtained from the ES-CV procedure as

$$K_{ES\text{-}CV} = \underset{2 \leq K \leq K_{max}}{\arg\max}\ \ell\left(\hat{p}, \hat{q} \mid \phi_K\right) \tag{5.37}$$

satisfies

$$\lim_{n\to\infty} Pr\Big(K_{ES\text{-}CV} = K^*\Big) = 1. \tag{5.38}$$

Proof. First when $K = K^*$, denote the estimated within and between community probability as $\hat{p}^*$ and $\hat{q}^*$. To simplify notation, we write $\hat{\ell}_K = \ell\left(\hat{p}, \hat{q} \mid \phi_K\right)$ and $\tilde{\ell}_{K^*} = \ell\left(\hat{p}^*, \hat{q}^* \mid \phi_{K^*}\right)$. Their conditional expectation ξ_K and ξ_{K^*} are given in (5.18). In order to prove (5.38), it is equivalent to show for any $K \in \{2, \ldots, K_{max}\}$ and $K \neq K^*$,

$$Pr\Big(\tilde{\ell}_{K^*} < \hat{\ell}_K\Big) \to 0. \tag{5.39}$$

A set of sufficient conditions for (5.39) are

$$Pr\Big(\tilde{\ell}_{K^*} - \xi_{K^*} < -s\Big) \to 0, \tag{5.40}$$

$$Pr\Big(\hat{\ell}_K - \xi_K > t\Big) \to 0, \tag{5.41}$$

$$Pr\Big(\xi_{K^*} - \xi_K - s - t > 0\Big) \to 1, \tag{5.42}$$

for some $t > 0$ and $s > 0$. To prove (5.40), when $K = K^*$ define

$$\hat{p}^*_{ij} := \begin{cases} \hat{p}^* & \text{if } \phi_{K^*}(i) = \phi_{K^*}(j) \\ \hat{q}^* & \text{if } \phi_{K^*}(i) \neq \phi_{K^*}(j). \end{cases} \tag{5.43}$$

Then $\tilde{\ell}_{K^*} - \xi_{K^*}$ can be re-written as

$$\tilde{\ell}_{K^*} - \xi_{K^*} = \sum_{(i,j)\in\mathcal{E}^{(2)}} \left[\Big(A_{ij} - E(A_{ij})\Big) \log\left(\frac{\hat{p}^*_{ij}}{1-\hat{p}^*_{ij}}\right)\right], \tag{5.44}$$

where $E(A_{ij}) = p$ if $\phi^*(i) = \phi^*(j)$ and $E(A_{ij}) = q$ if $\phi^*(i) \neq \phi^*(j)$. By Condition 2

and Lemma 5.3.3, there exists a $\delta_1 \in (0,1)$ and $\delta_1 < \delta - \frac{\sqrt{\log n}}{n\sqrt{r}}$ such that

$$\begin{aligned} & Pr(\delta_1 < \hat{p}^* < 1-\delta_1) \\ \geq\ & Pr\left(\delta - \frac{\sqrt{\log n}}{n\sqrt{r}} < \hat{p}^* < 1 - \delta + \frac{\sqrt{\log n}}{n\sqrt{r}}\right) \\ \geq\ & Pr\left(|\hat{p}^* - p| < \frac{\sqrt{\log n}}{n\sqrt{r}}\right) \\ \geq\ & 1 - \exp\big(-c_1 n^2 r\big) - 2n^{-c_2} - h(nr). \end{aligned} \tag{5.45}$$

A similar probabilistic bound can be established for $\hat{q}^*$ too. Set $U = \left|\log\left(\frac{1-\delta_1}{\delta_1}\right)\right|$. Thus these exist two positive constants c' and c'' such that for any $(i,j) \in \mathcal{E}^{(2)}$,

$$Pr\left(\left|\log\left(\frac{\hat{p}^*_{ij}}{1-\hat{p}^*_{ij}}\right)\right| > U\right) \ \leq\ \exp\big(-c' n^2 r\big) + 2n^{-c''} + h(nr). \tag{5.46}$$

Then for $s = n\sqrt{(1-r)\log n}$, by Lemma 5.3.2 and the Hoeffding Inequality (Hoeffding, 1963), we have

$$\begin{aligned} & Pr\left(\tilde{\ell}_{K^*} - \xi_{K^*} < -s\right) \\ \leq\ & Pr\left(\left|\sum_{(i,j)\in\mathcal{E}^{(2)}} \left[\big(A_{ij} - E(A_{ij})\big)\log\left(\frac{\hat{p}^*_{ij}}{1-\hat{p}^*_{ij}}\right)\right]\right| > s\right) \\ \leq\ & Pr\left(\left|\sum_{(i,j)\in\mathcal{E}^{(2)}} \big[\big(A_{ij} - E(A_{ij})\big)\big]\right| > \frac{s}{U}\right) + Pr\left(\left|\log\left(\frac{\hat{p}^*_{ij}}{1-\hat{p}^*_{ij}}\right)\right| > U\right) \\ \leq\ & \sum_{t=0}^{n(n-1)/2} Pr\left(\left|\sum_{(i,j)\in\mathcal{E}^{(2)}} \big[\big(A_{ij} - E(A_{ij})\big)\big]\right| > \frac{s}{U} \,\middle|\, |\mathcal{E}^{(2)}| = t\right) Pr\Big(|\mathcal{E}^{(2)}| = t\Big) \\ & +\ \exp\left(-c' nr\right) + 2n^{-c''} + h(nr) \\ \leq\ & \sum_{t=0}^{n^2(1-r)} Pr\left(\left|\sum_{(i,j)\in\mathcal{E}^{(2)}} \big[\big(A_{ij} - E(A_{ij})\big)\big]\right| > \frac{s}{U} \,\middle|\, |\mathcal{E}^{(2)}| = t\right) Pr\Big(|\mathcal{E}^{(2)}| = t\Big) \\ & +\ Pr\Big(|\mathcal{E}^{(2)}| > n^2(1-r)\Big) + \exp\big(-c' n^2 r\big) + 2n^{-c''} + h(nr) \end{aligned}$$

$$
\begin{aligned}
&\leq 2\sum_{t=0}^{n^2(1-r)} \exp\Big(-\frac{2s^2}{tU^2}\Big) Pr\Big(|\mathcal{E}^{(2)}| = t\Big) + \exp\Big(-\frac{n(n-1)(1-r)}{6}\Big) \\
&\quad + \exp\big(-c'n^2r\big) + 2n^{-c''} + h(nr) \\
&\leq 2\exp\Big(-\frac{2\log n}{U^2}\Big) + \exp\Big(-\frac{n(n-1)(1-r)}{6}\Big) + \exp\big(-c'n^2r\big) + 2n^{-c''} + h(nr) \\
&\leq 2n^{-c'''} + \exp\Big(-\frac{n(n-1)(1-r)}{6}\Big) + \exp\big(-c'n^2r\big) + 2n^{-c''} + h(nr).
\end{aligned}
$$

This concludes (5.40) under condition (i). A similar proof applies to (5.41) for $t = n\sqrt{(1-r)\log n}$.

Our next goal is to prove (5.42). Because of Condition 3, it is sufficient to show

$$
Pr\Big(\xi_{K^*} - \xi_K - s - t > 0 \;\Big|\; \phi_{K^*} = \phi^*\Big) \to 1. \tag{5.47}
$$

Notice that given $\phi_{K^*} = \phi^*$, we can write

$$
\begin{aligned}
&\xi_{K^*} - \xi_K \\
&= |T_{11}^{(2)}|\Big(D_{KL}(p \,||\, \hat{p}) - D_{KL}(p \,||\, \hat{p}^*)\Big) + |T_{12}^{(2)}|\Big(D_{KL}(p \,||\, \hat{q}) - D_{KL}(p \,||\, \hat{p}^*)\Big) \\
&\quad + |T_{21}^{(2)}|\Big(D_{KL}(q \,||\, \hat{p}) - D_{KL}(q \,||\, \hat{q}^*)\Big) + |T_{22}^{(2)}|\Big(D_{KL}(q \,||\, \hat{q}) - D_{KL}(q \,||\, \hat{q}^*)\Big) \\
&\geq |T_{11}^{(2)}| D_{KL}(p \,||\, \hat{p}) - \big(|T_{11}^{(2)}| + |T_{12}^{(2)}|\big) D_{KL}(p \,||\, \hat{p}^*) - \big(|T_{21}^{(2)}| + |T_{22}^{(2)}|\big) D_{KL}(q \,||\, \hat{q}^*) \\
&\geq \frac{|T_{11}^{(2)}|(p-\hat{p})^2}{2} - \frac{|W_{\phi^*}^{(2)}|(p-\hat{p}^*)^2}{\hat{p}^*(1-\hat{p}^*)} - \frac{|B_{\phi^*}^{(2)}|(q-\hat{q}^*)^2}{\hat{q}^*(1-\hat{q}^*)} \\
&= I - II - III,
\end{aligned} \tag{5.48}
$$

where $W_{\phi^*}^{(2)}$ and $B_{\phi^*}^{(2)}$ are defined in (5.11) and (5.13) respectively, and $T_{11}^{(2)}$, $T_{12}^{(2)}$, $T_{21}^{(2)}$, and $T_{22}^{(2)}$ are defined in (5.16).

When $K = K^- < K^*$, Assumption 5.3.6 requires the existence of $\delta^- = C_{K^-}(p -$

$q)/K^* > 0$ for some positive constant C_{K^-} such that

$$Pr\Big(\hat{p} - p < -\delta^-\Big) \geq 1 - \exp\Big(- C_{K^-} r n^2 \left(p - q\right)^2 \Big). \tag{5.49}$$

This implies

$$Pr\left((\hat{p} - p)^2 \geq \frac{C_{K^-}^2 (p-q)^2}{K^{*2}} \right) \geq 1 - \exp\Big(- C_{K^-} r n^2 \left(p - q\right)^2 \Big). \tag{5.50}$$

Combining (5.50) with (5.33), we obtain the following stochastic lower bound for the first term I in (5.48):

$$\begin{aligned} Pr\left(\; I \geq \frac{n^2(1-r)\gamma_1 \pi_0^2 C_{K^-}^2 (p-q)^2}{4K^{*4}} \right) \;\; \geq \;\; & 1 - \exp\Big(- C_{K^-} r n^2 \left(p - q\right)^2 \Big) \\ & - \exp\Big(- C_{11} n^2 (1-r) \Big). \end{aligned} \tag{5.51}$$

In other words, the term I grows in a minimum order of $n^2(1-r)(p-q)^2$ in probability.

For the term II in (5.48), first by Lemma 4.2.4, we know under Condition 1,

$$|W_{\phi^*}| \;\; \leq \;\; \frac{n^2}{2}\left(1 - \frac{K^* - 1}{K^*}\pi_0(2 - \pi_0)\right) - \frac{n}{2}. \tag{5.52}$$

Because $W_{\phi^*}^{(2)}$ comes from a Binomial sampling from W_{ϕ^*} with probability $1 - r$, by using the large deviation inequality for the Binomial distribution (e.g. Equation 1.6 of Janson 2016), we can derive for some positive constant c_w,

$$Pr\left(\left| W_{\phi^*}^{(2)} \right| > n^2(1-r) \right) \leq \exp\Big(- c_w n^2 (1-r) \Big). \tag{5.53}$$

Furthermore by Lemma 5.3.3,

$$Pr\left((p - \hat{p}^*)^2 > \frac{\log n}{n^2 r} \right) \leq \exp\Big(- c_1 n^2 r \Big) + 2n^{-c_2} + h(nr). \tag{5.54}$$

Combining (5.53), (5.54) with (5.45), we obtain

$$\begin{aligned}
& Pr\left(II > \frac{(1-r)\log n}{r\delta_1(1-\delta_1)}\right) \\
= \; & Pr\left(\frac{|W^{(2)}_{\phi^*}|(p-\hat{p}^*)^2}{\hat{p}^*(1-\hat{p}^*)} > \frac{n^2(1-r)\log n}{n^2 r\delta_1(1-\delta_1)}\right) \\
\leq \; & Pr\Big(|W^{(2)}_{\phi^*}| > n^2(1-r)\Big) + Pr\Big((p-\hat{p}^*)^2 > \frac{\log n}{n^2 r}\Big) \\
& + Pr\Big(\hat{p}^*(1-\hat{p}^*) \leq \delta_1(1-\delta_1)\Big) \\
\leq \; & \exp\Big(-c_w n^2(1-r)\Big) + \exp\Big(-c_1 n^2 r\Big) + 2n^{-c_2} + h(nr) \\
& + 1 - Pr\big(\delta_1 < \hat{p}^* < 1-\delta_1\big) \\
\leq \; & \exp\Big(-c_w n^2(1-r)\Big) + 2\exp\Big(-c_1 n^2 r\Big) + 4n^{-c_2} + 2h(nr). \qquad (5.55)
\end{aligned}$$

This concludes the term $II = O_p\Big((1-r)(\log n)/r\Big)$. The same order of stochastic upper bound can be derived for the term III in (5.48) using a similar argument

$$\begin{aligned}
Pr\left(III > \frac{(1-r)\log n}{r\delta_1(1-\delta_1)}\right) \; \leq \; & \exp\Big(-c_b n^2(1-r)\Big) + 2\exp\Big(-c_3 n^2 r\Big) \\
& + 4n^{-c_4} + 2h(nr). \qquad (5.56)
\end{aligned}$$

Combinining (5.51), (5.55), and (5.56) together, for a sufficiently large n and $s = t = n\sqrt{(1-r)\log n}$, we have

$$\frac{n^2(1-r)\gamma_1\pi_0^2 C^2_{K^-}(p-q)^2}{4K^{*4}} - \frac{2(1-r)\log n}{r\delta_1(1-\delta_1)} > s + t \qquad (5.57)$$

under condition (i), (ii), (iii), and (iv). Therefore when $\phi_{K^*} = \phi^*$,

$$Pr\Big(\xi_{K^*} - \xi_K - s - t > 0\Big) = Pr\Big(I - II - III - s - t > 0\Big) \to 1. \qquad (5.58)$$

This concludes (5.42) for $K = K^- < K^*$.

When $K = K^+ > K^*$, given $\phi^* = \hat{\phi}^*$, we can re-write

$$\begin{aligned}
& \xi_{K^*} - \xi_K \\
= \; & |T_{11}^{(2)}|\Big(D_{KL}(p \mid\mid \hat{p}) - D_{KL}(p \mid\mid \hat{p}^*)\Big) + |T_{12}^{(2)}|\Big(D_{KL}(p \mid\mid \hat{q}) - D_{KL}(p \mid\mid \hat{p}^*)\Big) \\
& + |T_{21}^{(2)}|\Big(D_{KL}(q \mid\mid \hat{p}) - D_{KL}(q \mid\mid \hat{q}^*)\Big) + |T_{22}^{(2)}|\big(D_{KL}(q \mid\mid \hat{q}) - D_{KL}(q \mid\mid \hat{q}^*)\Big) \\
\geq \; & |T_{22}^{(2)}|D_{KL}(q \mid\mid \hat{q}) - \big(|T_{11}^{(2)}| + |T_{12}^{(2)}|\big)D_{KL}(p \mid\mid \hat{p}^*) - \big(|T_{21}^{(2)}| + |T_{22}^{(2)}|\big)D_{KL}(q \mid\mid \hat{q}^*) \\
\geq \; & \frac{|T_{22}^{(2)}|(q-\hat{q})^2}{2} - \frac{|W_{\phi^*}^{(2)}|(p-\hat{p}^*)^2}{\hat{p}^*(1-\hat{p}^*)} - \frac{|B_{\phi^*}^{(2)}|(q-\hat{q}^*)^2}{\hat{q}^*(1-\hat{q}^*)} \\
= \; & IV - II - III
\end{aligned} \tag{5.59}$$

For IV, by a similar derivation as used for the term I in (5.48), we can obtain

$$\begin{aligned}
Pr\left(\; IV \geq \frac{n^2(1-r)\gamma_2\pi_0^2 C_{K^+}^2 (p-q)^2}{4K^{*4}}\right) \;\geq\; & 1 - \exp\Big(-C_{K^+} r n^2 (p-q)^2\Big) \\
& - \exp\Big(-C_{22} n^2 (1-r)\Big).
\end{aligned} \tag{5.60}$$

Therefore we can prove (5.42) for $K = K^+ > K^*$ as well. This concludes the proof of (5.38) for any $K \neq K^*$. □

By Theorem 2 of Vinayak et al. 2014, we know that when certain conditions that are weaker than theorem conditions (i) and (ii) are met, the ES-CV method by the improved convex optimization program in Algorithm 3 satisfies

$$Pr\left(\phi_{K^*} \neq \phi^*\right) \leq c_1 n^2 \exp(-c_2 n_{min}). \tag{5.61}$$

for some positive constant c_1 and c_2. This confirms Condition 3 with an explicit form of $h(\cdot)$. As a result, we can conclude that ES-CV by Algorithm 3 is a consistent method of determining the community number K.

5.4 Simulations

We run simulations to evaluate the ES-CV performance similar to those we ran for NS-CV in Section 4.3. We first evaluate the speed of convergence for ES-CV to choose the correct community number as the network size n increases. In this simulation, we generate networks with $K^* = 4$ communities under the Stochastic Block Model with the network size n growing from 100 to 500. We consider the same six combination of the within-community probability p and the between-community probability q as we did for NS-CV in Figure 4.1. In regard to the choices of the cross validation fold numbers, we explore 4-fold, 9-fold, 25-fold, and 64-fold ES-CV, which are corresponding to 2-fold, 3-fold, 5-fold, and 8-fold NS-CV in Figure 4.1. We plot the ES-CV success rate calculated from 100 iterations against the network size n in Figure 5.1.

According to Figure 5.1, the ES-CV success rate increases quickly to 1.0 as the network size n grows in all scenarios except Scenario 3 where $p = 0.3$ and $q = 0.2$. This matches what we have seen in the NS-CV simulation. In Scenario 3, because the p and q are close to each other, it is more difficult to distinguish communities regardless of the cross validation methods. As for the choice of the cross validation fold number, ES-CV with a large fold number is shown to outperform a small fold number. This is also consistent with what we have seen in NS-CV. Therefore, we may recommend a 9-fold as a default choice for ES-CV, which is aligned with our 3-fold NS-CV recommendation in Section 4.3.

To compare Figure 5.1 with Figure 4.1, we find out that the ES-CV success rate grows slightly faster than NS-CV as the network size increases. For instance, in Scenario 1 ($p = 0.3, q = 0.1$), the 4-fold ES-CV achieves a 1.0 success rate at $n = 200$, while the corresponding 2-fold NS-CV has the success rate less than 0.8 at $n = 200$. For another example, in Scenario 4 ($p = 0.6, q = 0.2$), the success rate of ES-CV reaches 1.0 even at $n = 100$. However, the success rate of NS-CV is below 1.0 at

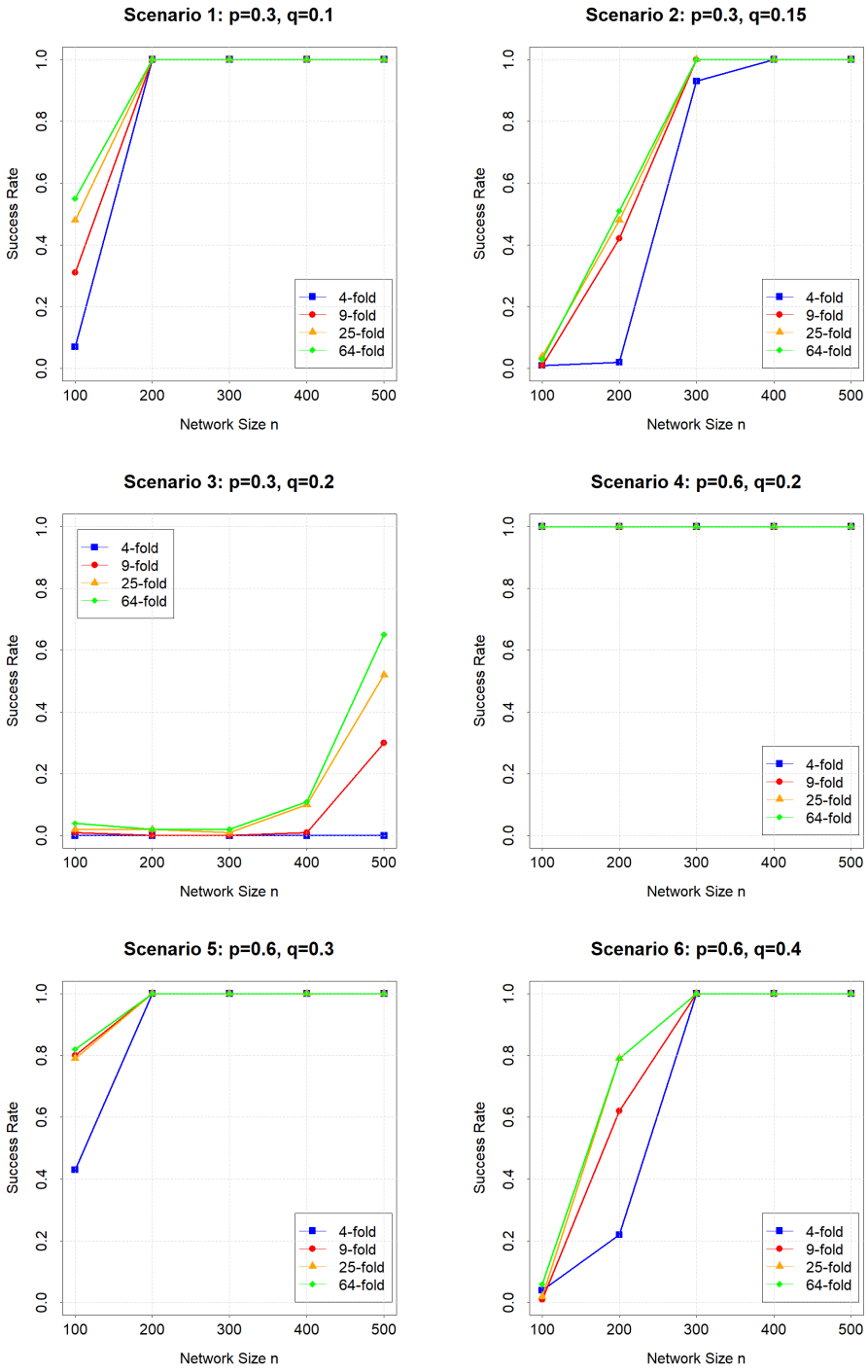


Figure 5.1: ES-CV success rates over the network sizes in six scenarios. The true model has $K^* = 4$. The success rate is calculated from 100 iterations.

$n = 100$. On the other hand, we can also see that the ES-CV can have a much worse performance than NS-CV when the p and q are very close to each other. For example, in the worst-case Scenario 3 ($p = 0.3, q = 0.2$), the ES-CV success rate stays near zero until $n = 500$. In contrast, although the NS-CV success rate is also low, it is at least maintained in the neighborhood of 0.2.

Our second simulation is to explore how the network sparsity and unbalance level affect the ES-CV performance. The simulation settings are nearly identical to what we did for NS-CV in Figure 4.2. The only exception is that we run a 9-fold ES-CV instead of a 3-fold NS-CV in this simulation. We plot the ES-CV success rates against the network sparsity level in Figure 5.2.

Figure 5.2 shows that ES-CV has a good performance similar to NS-CV when the network is relatively balanced. In this case, the ES-CV success rate is generally high provided the network is not too sparse (i.e. $q \geq 0.1, p \geq 0.3$). It is also seen that the ES-CV performance declines slightly at a larger K^* value. The same phenomenon occurred for NS-CV too. However, when the network becomes moderately or highly unbalanced, ES-CV underperforms NS-CV. For instance, in the first chart of Figure 5.2, when the ratio between the two community sizes is 50:450, the ES-CV success rate are near zero until $q \geq 0.15$. On the contrary, at the same community size ratio, Figure 4.2 shows the NS-CV success rate is close to 1.0 even at $q = 0.05$.

Finally we plot the success rates of 3-fold NS-CV, 9-fold ES-CV, and 3-fold Chen and Lei's CV method (Chen and Lei, 2018) over the network size n on the same chart for three combinations of p and q in Figure 5.3. The true model has $K^* = 4$ and balanced community sizes. Next to the success rate chart, we also plot the average chosen K from the three cross validation methods. Interestingly, both ES-CV and Chen and Lei's CV method tend to underestimate K, while NS-CV tends to overestimate it. However, when the network size grows large ($n = 500$), the average chosen K converges to the true $K^* = 4$ as described by the consistency theorem.

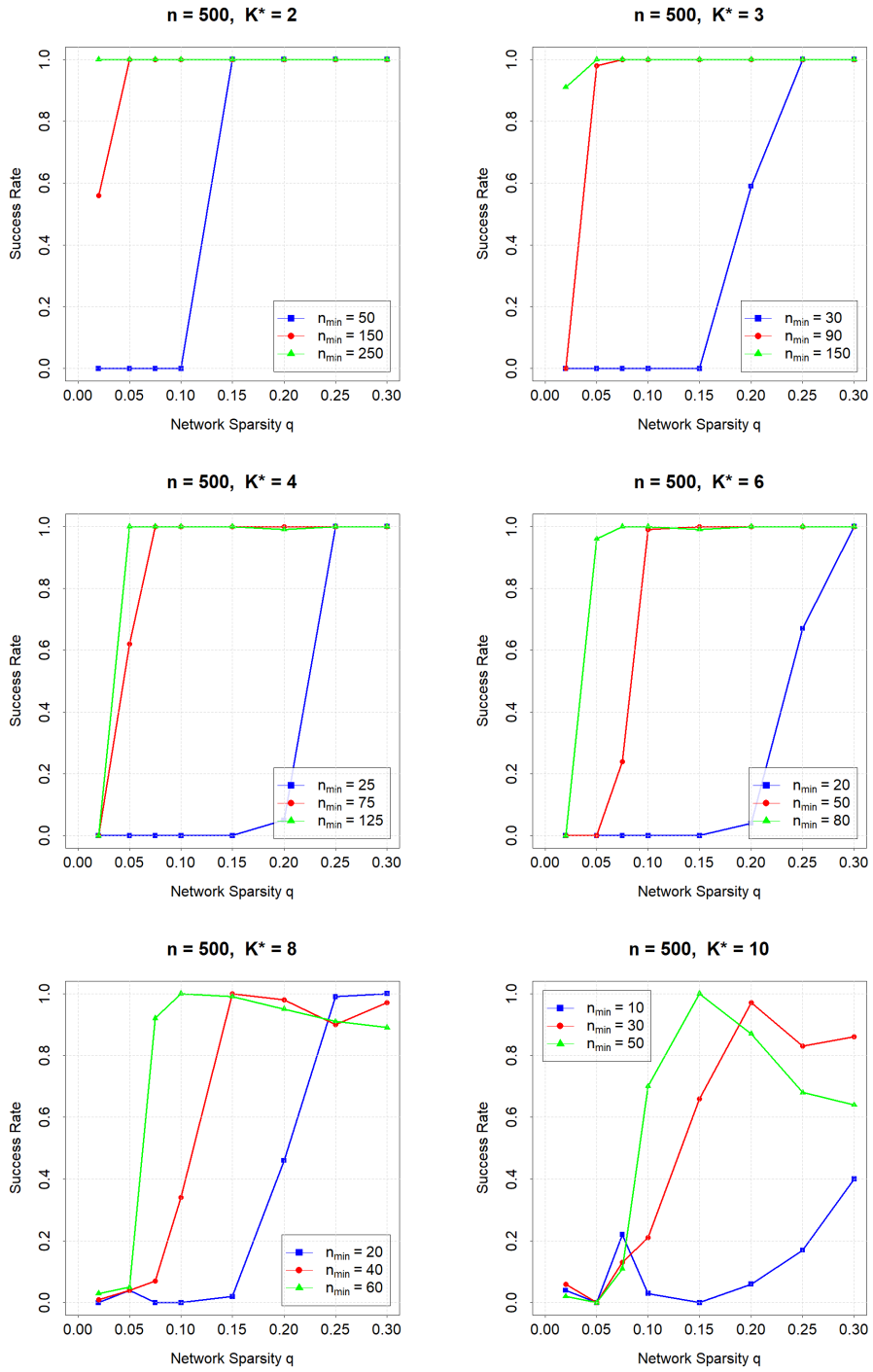


Figure 5.2: 9-fold ES-CV success rates over the network sparsity levels varying by choosing the between-community probability $q \in \{0.02, 0.05, 0.10, 0.20, 0.25, 0.3\}$ and setting the within-community probability $p = 3q$. The community imbalance levels are controlled by the smallest community size n_{min}. The success rates are calculated from 100 iterations.

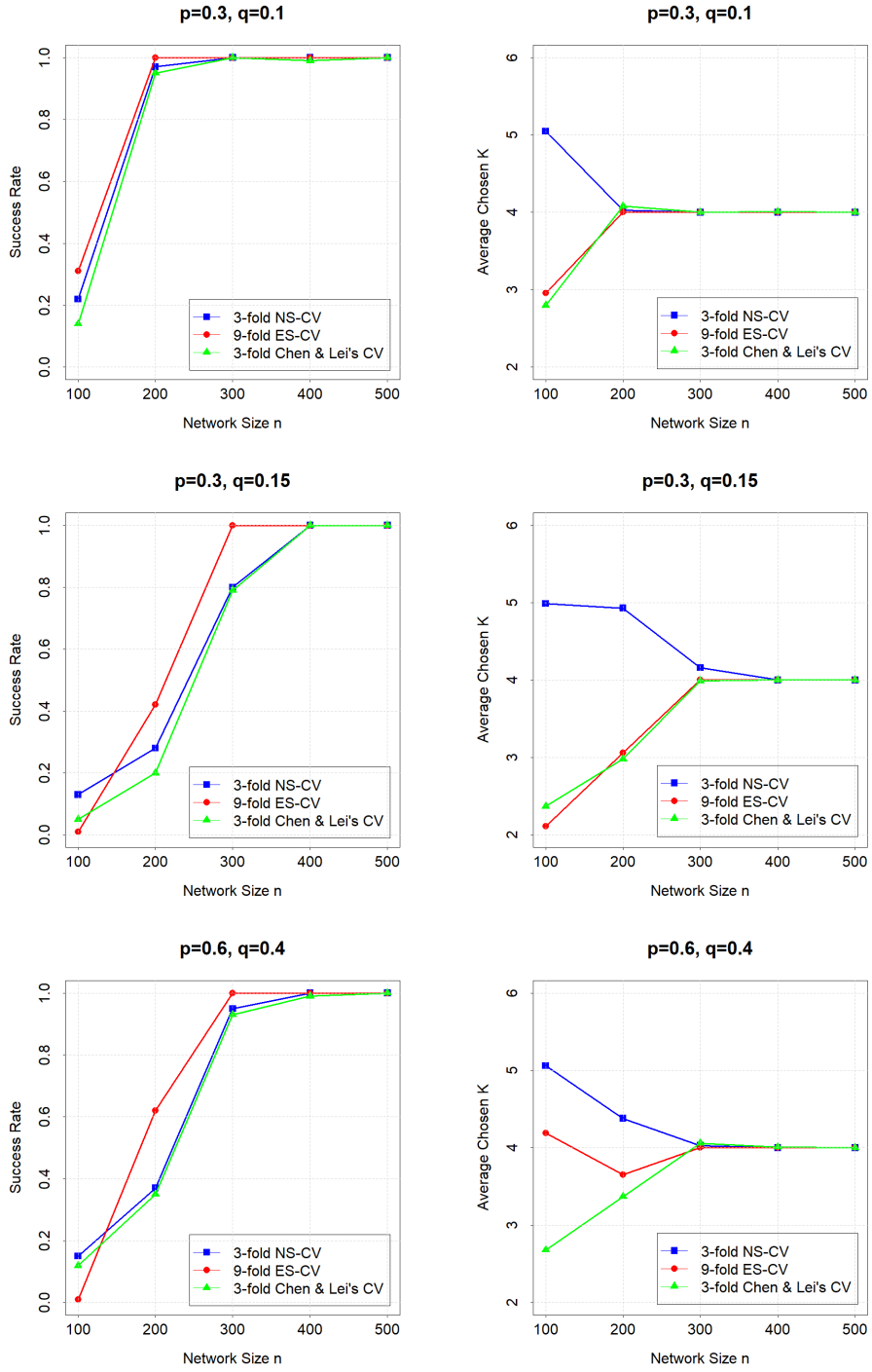


Figure 5.3: Compare the success rates and the average chosen K for 3-fold NS-CV, 9-fold ES-CV, and 3-fold Chen & Lei's CV method (Chen and Lei, 2018). The true model has $K^* = 4$ and balanced communities sizes. The success rate and average chosen K are calculated from 100 iterations.

5.5 Applications

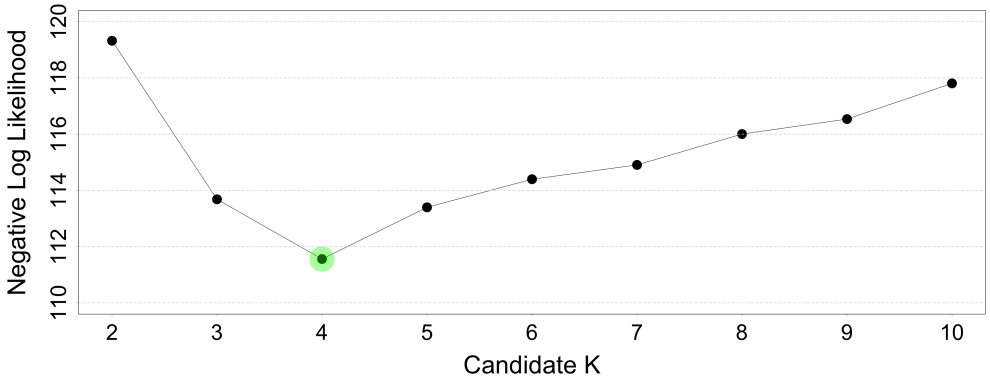
5.5.1 International Trade

We apply the ES-CV method to analyze the international trade network that consists of 58 countries whose import and export amounts (in US dollars) between 1981 and 2000 were collected by Westveld and Hoff (2011). We construct the adjacency matrix in the same way as we did in Section 4.4.1 that are based on the total trade between any two countries in year 2000. We apply the Algorithm 2 with use of the "Soft-Impute" method to run a 9-fold ES-CV on the international trade data. We plot the ES-CV negative log likelihood function against the candidate K values from 2 to 10 in Figure 5.4(a), which suggests the optimal community number is $K = 4$.

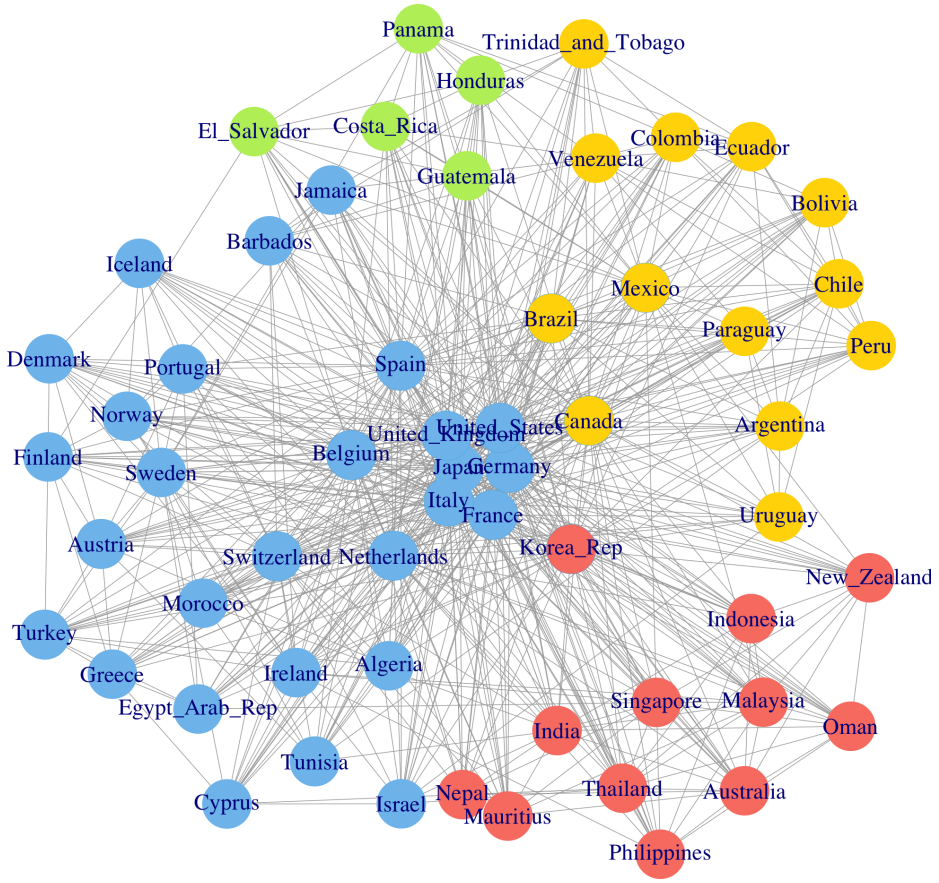
Because ES-CV determines the same community number $K = 4$ as NS-CV, the international trade network is clustered by Spectral Clustering to the same four trading groups as we showed in Section 4.4.1. For completeness of the analysis, we replicate Figure 4.4(b) as Figure 5.4(b) in this section. Our interpretation of the community composition in Section 4.4.1 remains unchanged.

5.5.2 The U.S. Senate Network

We apply 9-fold ES-CV to re-analyze the 108th U.S. Senate network. For the purpose of comparison, we maintain the same definition of a connection between two senators based on ≥ 5 sponsor-cosponsorship in legislation. The ES-CV negative log likelihood function is charted against the candidate K value from 2 to 10 in Figure 5.5(a). Note that this negative log likelihood function features an overall increasing trend with the minimal value achieved at $K = 3$. Yet, the difference of the log likelihood between $K = 2$ and $K = 3$ is very small, which suggests that ES-CV may also cluster the network into $K = 2$ communities. This observation matches the simulation result



(a) ES-CV Optimal K Achieved at 4

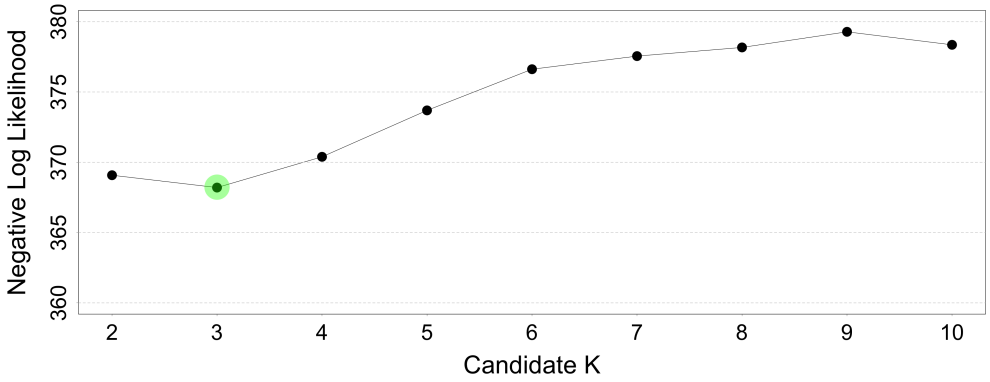


(b) Clustering of International Trade Countries ($K = 4$)

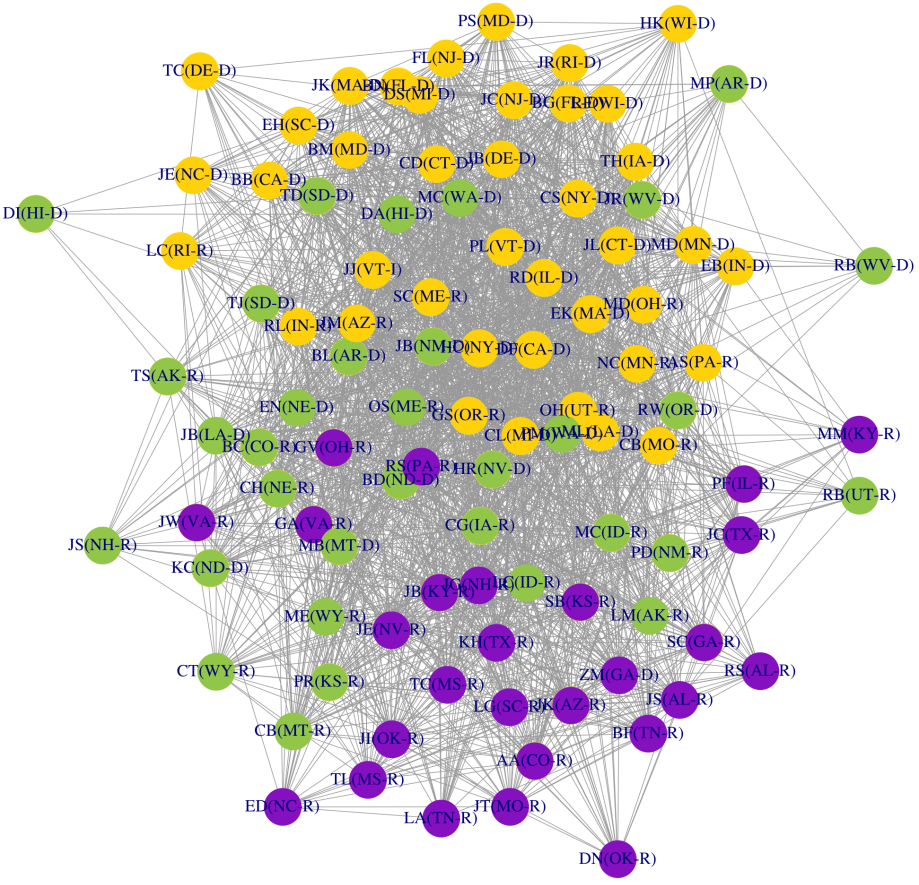
Figure 5.4: ES-CV Community Detection for the 2000 international trade countries Network.

in Figure 5.3, where ES-CV is shown to have a tendency of underestimating $K = 3$ especially under a small network size ($n = 100$ in this case).

We draw the three clusters of the U.S. Senate members and their connections in Figure 5.5(b). We tabulate the detailed composition of the three clusters of the Senate members in Table 5.1 and also summarize the party affiliations for the three clusters in Table 5.2. Based on these results, we see Cluster 1 has a majority of Democrats, while Cluster 3 has a majority of Republicans. Cluster 2 consists of an approximately equal number of Democrats and Republicans. It may be considered a moderate group inside the U.S. Senate. Note that because ES-CV tends to underestimate the community number, the NS-CV method may be especially preferred when the goal is to detect any small caucuses inside the Senate beyond party lines.



(a) Optimal K Achieved at 3



(b) Clustering of the U.S. Senate Members ($K = 3$)

Figure 5.5: ES-CV Community Detection for the 108th US Senate Network.

Cluster	**Senators** Names
Cluster 1	EB(IN-D) JB(DE-D) CB(MO-R) BB(CA-D) TC(DE-D) LC(RI-R) HC(NY-D) NC(MN-R) SC(ME-R) JC(NJ-D) MD(MN-D) MD(OH-R) RD(IL-D) JE(NC-D) RF(WI-D) DF(CA-D) TH(IA-D) OH(UT-R) JJ(VT-I) EK(MA-D) HK(WI-D) ML(LA-D) FL(NJ-D) CL(MI-D) JL(CT-D) JM(AZ-R) BM(MD-D) BN(FL-D) PS(MD-D) CS(NY-D) GS(OR-R) DS(MI-D) RL(IN-R) AS(PA-R) CD(CT-D) BG(FL-D) EH(SC-D) JK(MA-D) PL(VT-D) JR(RI-D)
Cluster 2	DA(HI-D) JB(NM-D) CB(MT-R) RB(WV-D) MC(WA-D) KC(ND-D) MC(ID-R) PD(NM-R) BD(ND-D) ME(WY-R) CG(IA-R) CH(NE-R) TJ(SD-D) BL(AR-D) LM(AK-R) PM(WA-D) EN(NE-D) MP(AR-D) PR(KS-R) OS(ME-R) TS(AK-R) JS(NH-R) CT(WY-R) RW(OR-D) RB(UT-R) BC(CO-R) LC(ID-R) MB(MT-D) JB(LA-D) JR(WV-D) TD(SD-D) HR(NV-D) DI(HI-D)
Cluster 3	LA(TN-R) AA(CO-R) GA(VA-R) SB(KS-R) JB(KY-R) SC(GA-R) TC(MS-R) JC(TX-R) ED(NC-R) JE(NV-R) PF(IL-R) LG(SC-R) KH(TX-R) JI(OK-R) JK(AZ-R) TL(MS-R) ZM(GA-D) DN(OK-R) RS(PA-R)JS(AL-R) RS(AL-R) JT(MO-R) JW(VA-R) JG(NH-R) GV(OH-R) BF(TN-R) MM(KY-R)

Table 5.1: Composition of the 3 Clusters in the 108th U.S. Senate Determined by ES-CV.

		Party Democrat	Republican	Independent	Total
Cluster	1	29	10	1	40
	2	18	15	0	33
	3	1	26	0	27
	Total	48	51	1	100

Table 5.2: Party Summary for the 3 Clusters in the 108th U.S. Senate.

Chapter 6

Conclusion and Future work

In this dissertation, we have proposed two cross validation methods: NS-CV and ES-CV, to determine how many communities are in a network. These two methods assume that the network $(\mathcal{N}, \mathcal{E})$ follows the stochastic block model where the probability of a connection between any two members are dependent on the communities they belong to. The two cross validation methods follow different strategies of training and test partition. For NS-CV, we split the node set $\mathcal{N}$ into the training set $\mathcal{N}^{(1)}$ and the test set $\mathcal{N}^{(2)}$. Community detection is made first by applying the standard Spectral Clustering for the nodes in $\mathcal{N}^{(1)}$, followed by a use of the One-vs-Other classifier to cluster the nodes in $\mathcal{N}^{(2)}$. For ES-CV, we partition the edge set $\mathcal{E}$ into the training set $\mathcal{E}^{(1)}$ and the test set $\mathcal{E}^{(2)}$. We apply the matrix completion method on the adjacency matrix whose entries corresponding to the test set $\mathcal{E}^{(2)}$ are eliminated from the matrix. The singular vectors corresponding to a low rank recovery of the adjacency matrix are used to cluster the nodes into communities. In the validation step of NS-CV and ES-CV, the optimal community number is chosen among the candidate K values based on the maximum log likelihood criterion.

Theoretically, we prove consistency for NS-CV and ES-CV under some conditions and assumptions about the network composition and the training method for community detection, as well as some technical requirements regarding the network size,

the cross validation splitting ratio, and the difference between the within-community connection probability p and the between-community connection probability q. Our simulation shows that when the network is balanced and p and q are distant from each other, both ES-CV and NS-CV achieve a satisfactory success rate even at a network size as low as $n = 100$. The NS-CV method is better than ES-CV for an unbalanced or sparse network. Our simulation also shows that both NS-CV and ES-CV methods outperform the network cross validation method introduced by Chen and Lei for community detection.

We have also identified a few enhancement opportunities for NS-CV and ES-CV. We present them below for future research.

1. We would like to extend NS-CV and ES-CV to the Degree-Corrected Block Model, or DCBM (Dasgupta et al., 2004; Karrer and Newman, 2011), which is believed to have a better representation of many real-life networks than the Stochastic Block Model. Specifically, in NS-CV, the standard Spectral Clustering method and the "One-vs-Other" classifier may need to be updated to account for the additional degree-connection parameters for every node in the network. In ES-CV, it would be interesting to explore how the introduction of the degree-connection parameters would affect the matrix completion process and the subsequent clustering outcomes.

2. Our current NS-CV method uses the "One-vs-Other" classifier to cluster the nodes in $\mathcal{N}^{(2)}$ based on their connections to the nodes in $\mathcal{N}^{(1)}$. Although this is one of the most commonly used classification methods that can be easily understood and applied to the network clustering, many other advanced multiclass classification methods are developed in recent machine learning and statistics research to improve the classification outcome. Some notable methods include Naive Bayes (Rish et al., 2001), Support Vector Machine (Bredensteiner

and Bennett, 1999; Lee et al., 2004), Regression Trees (Breiman, 2017), Error-Correcting Output-Coding (Dietterich and Bakiri, 1994), Generalized Coding (Allwein et al., 2000), and Binary Hierarchical Classification (Kumar et al., 2002), etc. An application of any of these advanced classification methods in NS-CV may improve its performance especially when the network is either sparse or has an unbalanced structure.

3. In this dissertation, we introduce several programs and algorithms for matrix completion used in the ES-CV method. In recent years, this has become a very productive area where many new and efficient algorithms are developed to recover matrices with missing or corrupted values through a low rank optimization. In addition, a related technique known as Robust Principal Component Analysis also attracts a great deal of attention in machine learning and signal/image processing (Candès et al., 2011; Wright et al., 2009). These new algorithms and/or techniques may be applied to ES-CV to improve learning of the network structure solely from the training portion of the adjacency matrix. As seen in our proof of consistency for ES-CV, the convergence rate of these algorithms will also play a crucial role in determining how well the cross validation method can identify the correct community number for the network.

References

Airoldi, E. M., Blei, D. M., Fienberg, S. E., Xing, E. P., and Jaakkola, T. (2006). Mixed membership stochastic block models for relational data with application to protein-protein interactions. In *Proceedings of the international biometrics society annual meeting*, volume 15.

Allwein, E. L., Schapire, R. E., and Singer, Y. (2000). Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of machine learning research*, 1(Dec):113–141.

Amini, A. A., Chen, A., Bickel, P. J., Levina, E., et al. (2013). Pseudo-likelihood methods for community detection in large sparse networks. *The Annals of Statistics*, 41(4):2097–2122.

Bickel, P., Choi, D., Chang, X., Zhang, H., et al. (2013). Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels. *The Annals of Statistics*, 41(4):1922–1943.

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

Bickel, P. J. and Sarkar, P. (2016). Hypothesis testing for automated community detection in networks. *Journal of the Royal Statistical Society*, Series B(3):253–273.

Bredensteiner, E. J. and Bennett, K. P. (1999). Multicategory classification by support vector machines. In *Computational Optimization*, pages 53–79. Springer.

Breiman, L. (2017). *Classification and regression trees*. Routledge.

Cabreros, I., Abbe, E., and Tsirigos, A. (2016). Detecting community structures in hi-c genomic data. In *Information Science and Systems (CISS), 2016 Annual Conference on*, pages 584–589. IEEE.

Cai, J.-F., Candès, E. J., and Shen, Z. (2010). A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982.

Campbell, J. E. (1982). Cosponsoring legislation in the us congress. *Legislative Studies Quarterly*, 7(3):415–422.

Candès, E. J., Li, X., Ma, Y., and Wright, J. (2011). Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11.

Candès, E. J. and Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772.

Candès, E. J. and Tao, T. (2010). The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080.

Chen, K. and Lei, J. (2018). Network cross-validation for determining the number of communities in network data. *Journal of the American Statistical Association*, 113(521):241–251.

Chen, Y., Jalali, A., Sanghavi, S., and Xu, H. (2014). Clustering partially observed graphs via convex optimization. *Journal of Machine Learning Research*, 15(1):2213–2238.

Clauset, A., Newman, M. E., and Moore, C. (2004). Finding community structure in very large networks. *Physical review E*, 70(6):066111.

Dabbs, B. and Junker, B. (2016). Comparison of cross-validation methods for stochastic block models. *arXiv preprint arXiv:1605.03000*.

Dasgupta, A., Hopcroft, J. E., and McSherry, F. (2004). Spectral analysis of random graphs with skewed degree distributions. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 602–610. IEEE.

Daudin, J.-J., Picard, F., and Robin, S. (2008). A mixture model for random graphs. *Statistics and computing*, 18(2):173–183.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923.

Dietterich, T. G. and Bakiri, G. (1994). Solving multiclass learning problems via error-correcting output codes. *Journal of artificial intelligence research*, 2:263–286.

Donath, W. E. and Hoffman, A. J. (1973). Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425.

Drineas, P., Frieze, A., Kannan, R., Vempala, S., and Vinay, V. (2004). Clustering large graphs via the singular value decomposition. *Machine learning*, 56(1-3):9–33.

Fang, J. (2014). Targeted advertising based on social network analysis. In *Materials Science, Civil Engineering and Architecture Science, Mechanical Engineering and Manufacturing Technology*, volume 488 of *Applied Mechanics and Materials*, pages 1306–1309. Trans Tech Publications.

Fowler, J. H. (2006). Connecting the congress: A study of cosponsorship networks. *Political Analysis*, 14(4):456–487.

Gao, C., Ma, Z., Zhang, A. Y., Zhou, H. H., et al. (2018). Community detection in degree-corrected block models. *The Annals of Statistics*, 46(5):2153–2185.

Greenbaum, G., Templeton, A. R., and Bar-David, S. (2016). Inference and analysis of population structure using genetic data and network theory. *Genetics*, pages genetics–115.

Gulikers, L., Lelarge, M., and Massoulié, L. (2017). A spectral method for community detection in moderately sparse degree-corrected stochastic block models. *Advances in Applied Probability*, 49(3):686–721.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30.

Hoff, P. (2008). Modeling homophily and stochastic equivalence in symmetric relational data. In *Advances in Neural Information Processing Systems*, pages 657–664.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137.

Jain, P., Netrapalli, P., and Sanghavi, S. (2013). Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674. ACM.

Janson, S. (2016). Large deviation inequalities for sums of indicator variables. *arXiv preprint arXiv:1609.00533*.

Jia, Z. (2013). The research on parameters of spectral clustering based on svd. In *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology*, pages 23–27. IEEE.

Jin, J. (2015). Fast community detection by score. *The Annals of Statistics*, 43(1):57–89.

Karrer, B. and Newman, M. E. (2011). Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107.

Kumar, S., Ghosh, J., and Crawford, M. M. (2002). Hierarchical fusion of multiple classifiers for hyperspectral data analysis. *Pattern Analysis & Applications*, 5(2):210–220.

Latouche, P., Birmele, E., and Ambroise, C. (2012). Variational bayesian inference and complexity control for stochastic block models. *Statistical Modelling*, 12(1):93–115.

Lee, Y., Lin, Y., and Wahba, G. (2004). Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99(465):67–81.

Lei, J., Rinaldo, A., et al. (2015). Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1):215–237.

Li, X. and Chen, L. (2011). Recommendations based on network analysis. In *Advanced Computer Science and Information System (ICACSIS), 2011 International Conference on*, pages 9–16. IEEE.

Lin, Z., Chen, M., and Ma, Y. (2010). The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*.

Lindsay, B. G. (1988). Composite likelihood methods. *Contemporary Mathematics*, 80:221–239.

Mazumder, R., Hastie, T., and Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research*, 11(Aug):2287–2322.

McDaid, A. F., Murphy, T. B., Friel, N., and Hurley, N. J. (2013). Improved bayesian inference for the stochastic block model with application to large networks. *Computational Statistics & Data Analysis*, 60:12–31.

McSherry, F. (2001). Spectral partitioning of random graphs. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 529–537. IEEE.

Neville, J., Gallagher, B., Eliassi-Rad, T., and Wang, T. (2012). Correcting evaluation bias of relational classifiers with network cross validation. *Knowledge and information systems*, 30(1):31–55.

Newman, M. E. (2004). Detecting community structure in networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(2):321–330.

Newman, M. E. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113.

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856.

Nobile, A. and Fearnside, A. T. (2007). Bayesian finite mixtures with an unknown number of components: The allocation sampler. *Statistics and Computing*, 17(2):147–162.

Papadopoulos, S., Zigkolis, C., Tolias, G., Kalantidis, Y., Mylonas, P., Kompatsiaris, Y., and Vakali, A. (2010). Image clustering through community detection on hybrid

image similarity graphs. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 2353–2356. IEEE.

Qin, T. and Rohe, K. (2013). Regularized spectral clustering under the degree-corrected stochastic blockmodel. In *Advances in Neural Information Processing Systems*, pages 3120–3128.

Rish, I. et al. (2001). An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46.

Rohe, K., Qin, T., and Fan, H. (2014). The highest dimensional stochastic blockmodel with a regularized estimator. *Statistica Sinica*, pages 1771–1786.

Saldana, D. F., Yu, Y., and Feng, Y. (2017). How many communities are there? *Journal of Computational and Graphical Statistics*, 26(1):171–181.

Saldana, D.F, Y. Y. and Feng, Y. (2017). How many communities are there? *Journal of Computational and Graphical Statistics*, 26(1):171–181.

Sarkar, P. J. B. P. (2016). Hypothesis testing for automated community detection in networks. *Journal of the Royal Statistical Society*, B(78):253–273.

Serfling, R. J. (1974). Probability inequalities for the sum in sampling without replacement. *The Annals of Statistics*, 2(1):39–48.

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

Snijders, T. A. and Nowicki, K. (1997). Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of classification*, 14(1):75–100.

Soshnikov, A. (1999). Universality at the edge of the spectrum in wigner random matrices. *Communications in mathematical physics*, 207(3):697–733.

Spielmat, D. A. and Teng, S.-H. (1996). Spectral partitioning works: Planar graphs and finite element meshes. In *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*, pages 96–105. IEEE.

Tracy, C. A. and Widom, H. (1994). Level-spacing distributions and the airy kernel. *Communications in Mathematical Physics*, 159(1):151–174.

Vinayak, R. K., Oymak, S., and Hassibi, B. (2014). Graph clustering with missing data: Convex algorithms and analysis. In *Advances in Neural Information Processing Systems*, pages 2996–3004.

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

Von Luxburg, U., Belkin, M., and Bousquet, O. (2008). Consistency of spectral clustering. *The Annals of Statistics*, pages 555–586.

Wang, T., Neville, J., Gallagher, B., and Eliassi-Rad, T. (2011). Correcting bias in statistical tests for network classifier evaluation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 506–521. Springer.

Wang, Y. and Bickel, P. J. (2017). Likelihood-based model selection for stochastic block models. *The Annals of Statistics*, 45(2):500–528.

Wang, Y. J. and Wong, G. Y. (1987). Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8–19.

Wei, Y.-C. and Cheng, C.-K. (1989). Towards efficient hierarchical designs by ratio cut partitioning. In *Computer-Aided Design, 1989. ICCAD-89. Digest of Technical Papers., 1989 IEEE International Conference on*, pages 298–301. IEEE.

Westveld, A. H. and Hoff, P. D. (2011). A mixed effects model for longitudinal relational and network data, with applications to international trade and conflict. *The Annals of Applied Statistics*, pages 843–872.

Wigner, E. P. (1958). On the distribution of the roots of certain symmetric matrices. *Annals of Mathematics*, pages 325–327.

Wright, J., Ganesh, A., Rao, S., Peng, Y., and Ma, Y. (2009). Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, pages 2080–2088.

Yang, Y. (2007). Consistency of cross validation for comparing regression procedures. *The Annals of Statistics*, pages 2450–2473.

Zhao, Y., Levina, E., and Zhu, J. (2011). Community extraction for social networks. *Proceedings of the National Academy of Sciences*, 108(18):7321–7326.

Zhao, Y., Levina, E., and Zhu, J. (2012). Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, pages 2266–2292.