THE STATISTICAL DETECTION OF CLUSTERS IN NETWORKS

by

MARCUS ALAN BALLARD

MARCUS B. PERRY, COMMITTEE CHAIR
VOLODYMYR MELNYKOV
MICHAEL PORTER
SHARIF MELOUK
DANIEL BACHRACH

A DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Information Systems,
Statistics, and Management Science
in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2018

ABSTRACT

A network consists of vertices and edges that connect the vertices. A network is clustered by assigning each of the N vertices to one of k groups, usually in order to optimize a given objective function. This dissertation proposes statistical likelihood as an objective function for network clustering for both undirected networks, in which edges have no direction, and directed networks, in which edges have direction.

Clustering networks by optimizing an objective function is computationally expensive and quickly becomes prohibitive as the number of vertices in a network grows large. To address this, theorems are developed to increase the efficiency of likelihood parameter estimation during the optimization and a significant decrease in time-to-solution is demonstrated.

When the clustering performance of likelihood is rigorously compared to competitor objective function modularity using Monte Carlo simulation, likelihood is frequently found to be superior. A novel statistical significance test for clusters identified when using likelihood as an objective function is also derived and both clustering using the likelihood objective function and subsequent significance testing are demonstrated on real-world networks, both undirected and directed.

DEDICATION

To my family

ACKNOWLEDGMENTS

Thank you to my advisor, Marcus Perry, and my dissertation committee members for their guidance in this research and in professional matters. I would also like to thank all the faculty and staff in the Department of Information Systems, Statistics and Management Science in the Manderson Graduate School of Business for their instruction and assistance, without which none of this would have been possible.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

A network can be described by a set of vertices and the set of edges that connect the vertices. One active area of network-related research is clustering. In network clustering, the vertices of the networks are assigned to one of k clusters, with the assignment of a particular vertex to a particular cluster usually made in order to optimize a specified objective function. Clustering networks is both a dimension reduction technique in which the original N vertices are reduced to k representative clusters and a method for revealing underlying structure within the network that may not be visible to the naked eye.

This dissertation is comprised of three related essays on the subject of network clustering. The first essay, Chapter 2 of this dissertation, is an introduction to the statistical detection of clusters in undirected networks - that is, networks in which edges may exist between vertices but without a directional component. This essay proposes statistical likelihood as an objective function for optimization and uses the properties of the related likelihood ratio test to develop a novel statistical significance test for identified clusters. The performance of the proposed objective function is then rigorously tested against a competitor objective function using Monte Carlo simulation. Finally, demonstrations of both the proposed objective function and the statistical significance test are made on real-world networks.

The objective function proposed in Chapter 2 is computationally expensive to optimize. The second essay of this dissertation, Chapter 3, presents several theorems that significantly decrease both the time-to-solution and

required memory when using a heuristic optimization algorithm to find a near-optimal solution. The increased speed is demonstrated by comparing the results of optimizing the statistical likelihood objective function in a naive simulated annealing algorithm and a simulated annealing algorithm that takes advantage of the efficiencies proposed in this chapter.

The third essay, Chapter 4 of this dissertation, is an extension of the materials developed in Chapters 2 and 3 to the directed network case. Unlike undirected networks, the edges connecting vertices of a directed network are associated with a specific direction. The presence of direction must be accounted for in the statistical likelihood model and new theorems developed for directed networks in order to maintain the increased computational efficiencies derived in Chapter 3. Like the undirected case, the performance of the directed statistical likelihood objective function is rigorously compared to that of a competitor method using simulated networks. Further, real-world networks are clustered using the proposed directed-network objective function and the application of the statistical significance test for clusters is again demonstrated.

2 ON THE STATISTICAL DETECTION OF CLUSTERS IN UNDIRECTED NETWORKS

2.1 Abstract

The goal of network clustering algorithms is to assign each node in a network to one of several mutually exclusive groups based upon the observed edge set. Of the network clustering algorithms widely available, most make the effort to maximize the modularity metric. Although modularity is an intuitive and effective means to cluster networks, it provides no direct basis for quantifying the statistical significance of the detected clusters. In this paper, we consider undirected networks and propose a new objective function to maximize over the space of possible group membership assignments. This new objective function lends naturally to the use of information criterion (e.g., Akaike or Bayesian) for determining the "best" number of groups, as well as to the development of a likelihood ratio test for determining if the clusters detected provide significant new information. The proposed method is demonstrated using two real-world networks. Additionally, using Monte Carlo simulation, we compare the performances of the proposed clustering framework relative to that achieved by maximizing the modularity objective when applied to LFR benchmark graphs.

2.2 Introduction

Clustering has a wide array of applications, from pattern recognition and spatial data analysis to data mining and military intelligence. Regardless of the application, clustering methodologies are often used to explore a data set where the goal is to partition the sample into distinct groups, or to provide new

understanding about the underlying structure of the data. Different approaches have been developed to address the problem of clustering. A popular approach, known as hierarchical clustering, seeks to identify nested clusters in a data set (Gordon (1987)). Either agglomerative or divisive, hierarchical clustering algorithms either combine or separate observational units in order to produce the clusters. The output of such an algorithm is the dendrogram, where the user is left to determine the appropriate number of clusters for the particular data set. Another approach is k-means clustering. Using this approach, the user specifies the number of groups a priori and then randomly assigns each observational unit to one of those groups. The centroid of each of the groups is calculated, and each observational unit is reassigned to the nearest cluster. The centroids of these new groups are then recalculated and the observational units are again reassigned to the closest group. The process continues until group membership stabilizes. A good review of k-means clustering is given by Steinley (2006).

Unlike the previous methods, spectral clustering does not require the user to specify the number of groups a priori. This approach requires the calculation of a matrix to describe the similarity between each pair of observational units, i.e., the similarity matrix. The eigenvectors and eigenvalues of this matrix (i.e., the spectrum) are then calculated and used to identify group membership, e.g., one might bipartition the sample based on the sign of the elements of the eigenvector associated with the largest eigenvalue. The interested reader is directed to von Luxburg (2007) for a straightforward review of the technique.

Although clustering algorithms are often applied to conventional data sets, they can also be applied to network data (e.g., social networks, biological networks, computer networks, etc.). In such a case, the goal is typically to assign each node in the network to one of several mutually exclusive groups based upon information contained in the edge set. In general, a network can be defined as a graph $G = (V, E)$ with vertex set V and edge set E, with edge $e_{ii'} \in E$ denotes a connection (or relationship) between node $i \in V$ and node $i' \in V$ (for

$i = 1, ..., n, i' = 1, ..., n$, where $i \neq i'$). Networks can be characterized based on the types of edges that exist between nodes. An undirected network is a network in which $e_{ii'} = e_{i'i}$, i.e., a relationship from node i to node i' implies an equal relationship from node i' to node i. In contrast, a directed network does not have this restriction. The values taken by the elements of E further characterize a network. In a binary network, the edges may take only the binary values 0 and 1, indicating the absence or presence of a link, respectively. Conversely, weighted networks allow the edges to take continuous values, although these values are often restricted to be non-negative. The most popular approach to network clustering is to maximize the modularity metric, originally proposed by Newman and Girvan (2004). Network clustering via modularity is available in a number of network analysis software packages and thus is widely available to network analysts. To define modularity, consider a network of size N, and let $\omega_{ij} = 1$ if node i belongs to group j, and 0 otherwise ($i = 1, ..., N$ and $j = 1, ..., k$). Further, let $A_{ii'}$ denote the ii'th element of the adjacency matrix $\mathbf{A}$, m denote the total number of edges in the network, and d_i the degree of node i. For a given N x k group membership matrix $\boldsymbol{\omega}$ and N x N adjacency matrix $\mathbf{A}$, the modularity is defined as

$$Q(\boldsymbol{\omega}|\mathbf{A}) = \frac{1}{2m} tr(\boldsymbol{\omega}^T \mathbf{B} \boldsymbol{\omega}) \tag{2.1}$$

where tr($\mathbf{G}$) denotes the trace of the matrix $\mathbf{G}$ and

$$B_{ii'} = A_{ii'} - \frac{d_i d_{i'}}{2m} \tag{2.2}$$

denotes the elements of the so called modularity matrix.

Modularity is a useful, intuitive, and effective statistic for measuring the extent to which a given partition of a network is modular. Specifically, it measures the fraction of edges that fall within the given groups minus the expected such fraction if edges were distributed at random. Larger values of

modularity suggest the presence of densely intra-connected and sparsely inter-connected nodes. The clustering problem involves finding $\boldsymbol{\omega}^*$, i.e., the group membership matrix in the set of all group membership matrices $\boldsymbol{\Omega}_k$ that yields the maximum modularity value, or

$$\boldsymbol{\omega}^* = \arg\max_{\boldsymbol{\omega} \in \boldsymbol{\Omega}_k} [Q(\boldsymbol{\omega}|\mathbf{A})]. \tag{2.3}$$

In general, the optimization problem given above is not an easy one. In particular, for even moderately-sized networks, the number of possible ways to partition the vertex set is quite vast, rendering an exhaustive search infeasible. As such, heuristic search algorithms are often employed by which the number of possible network partitions evaluated is greatly reduced. In what follows we discuss some of these methods. Although maximizing modularity is by far the most popular objective, the methods discussed below are not exclusive to modularity. For a more comprehensive review of the larger number of algorithms available for performing community detection in graphs, the interested reader is referred to Fortunato (2010).

The fastest method in common use was developed by Clauset et al. (2004). Their work is a modification of Newman (2004), where although they do not alter the general approach of Newman's algorithm, they do optimize its memory usage, data storage, and computational methods for use with sparse networks; i.e., a vast majority of networks of interest. Newman's algorithm is a greedy, agglomerative, hierarchical clustering algorithm that seeks to maximize modularity at each step. The algorithm begins by assuming that each node represents an individual module, then merges the modules that lead to the greatest increase in modularity.

Others have employed stochastic search methods such as simulated annealing or genetic algorithms (Guimera et al. (2004); Ku cukpetek et al. (2005)). These methods are generally found to be slower by more accurate than other deterministic methods. In fact, Danon et al. (2005) found that simulated

annealing produced the most accurate results of any of the algorithms that were tested.

Another approach involves examining the spectral properties of various matrices. Newman (2006), for example, denotes the assignment of nodes into two groups in terms of an N x 1 vector $\mathbf{s}$ in which node i's membership in subgroup 1 is denoted by $s_i = 1(-1)$. By choosing the assignment of group membership in such a way as to maximize the inner product of $\mathbf{s}$ and the eigenvector associated with the largest eigenvalue of a function of the adjacency matrix, an approximately optimal partition can be determined. Each of these two subgroups can then be divided using a similar procedure.

Still another approach uses extremal optimization (Duch and Arenas (2005)), which focuses on correcting those nodes with the worst fit. Kernigan and Lin (1970) proposed a similar but more simplistic approach in which the graph is divided into equal parts. Also, there exists a class of search procedures that work by cutting the links between particular nodes or by otherwise physically dividing the global network into smaller pieces, Girvan and Newman (2002) and Newman and Girvan (2004). Finally there exists a class of probabilistic models, such as those discussed in Snijders and Nowicki (1997) and Zanghi et al. (2007), that treat the group membership assignments as a latent-class random vector.

Despite the large variety of methods, all clustering algorithms have the common goal of detecting clusters. Some algorithms require the number of clusters as an input parameter to the algorithm (e.e., k-means clustering). Various model selection procedures, Akaike (1974) and Schwarz (1978), have been developed that can be applied to this problem. Still other algorithms, Newman (2004) and Reichardt and Bornholdt (2006), have some sort of stopping criterion built in to the algorithm itself.

A growing body of research exists to guide the user in determining the correct number of groups; however, an equally important and related problem is determining if the output of these clustering algorithms is simply the result of

randomness, or if it displays significant structure in the data. That is, what if the number of clusters is actually 1? A small but growing body of work exists in this area. Bianconi et al. (2009) define a measure, Θ, based on entropy measures that seeks to quantify the relevance of some detected community structure; however, no distribution for this measure is given. Lancichinetti et al. (2010) describe a procedure based on extreme and order statistics than can be used to determine the significance of clusters in unweighted, undirected networks. More recently, Lancichinetti et al. (2011) extended this method to account for more general network structures, including weighted and/or directed networks. Zhao et al. (2011) propose a formal statistical test that uses simulation and permutations to approximate the distribution of the test statistic under the null hypothesis, and thus, facilitate testing.

In this paper, we consider undirected networks (weighted or unweighted) and propose a new objective function to maximize over the vertex set partition space. Since the proposed objective function is a likelihood function, it lends nicely to the use of information criterion proposed by Akaike (1974) or Schwarz (1978) for determining the "best" number of groups or clusters. Further, using the proposed objective function, it is easy to develop an approximate statistical test for determining if the clusters detected are indeed meaningful, or are simply a result of the randomness associated with the stochastic process that produced the network under consideration. We derive an approximation to the distribution of the maximum likelihood ratio statistic under the null hypothesis of a single cluster and use this distribution to obtain critical values for the test. We then evaluate the clustering performance obtained by maximizing the proposed objective function, relative to that achieved by maximizing modularity. We consider the benchmark graphs discussed in Lancichinetti et al. (2008) (i.e. LFR benchmark graphs) in our evaluations, since these graphs possess some properties of real-world networks. we also assess the power of the proposed statistical test when applied to LFR benchmarks, and with optimal group membership

assignments obtained by maximizing the proposed objective function, as well as those obtained by maximizing modularity.

2.3 A new objective function for network clustering

Although modularity is an effective means to cluster networks, it provides no basis for quantifying the statistical significance of the detected clusters. Large positive values of modularity suggest that the number of edges contained within the groups exceeds that of a completely random network. However, it does not quantify the chance of sampling the observed network under the assumption of a completely random network. Having an estimate of this probability would provide insight into the significance of any new information obtained through the clustering effort.

In this section we propose a new objective function for network clustering that permits the development of a formal statistical test on the detected clusters. In addition, like modularity, the proposed objective function will permit a means to estimating the number of groups when this quantity is unknown. This is important because of the number of groups is unknown a priori.

In the development of the proposed objective function, we make the following fundamental assumptions.

A1 The number of network nodes N is fixed.

A2 There is no information available on the nodes, other than an arbitrary label.

A3 The edges between nodes are conditionally independent random variables, given the group membership assignment.

Under the above assumptions, and for a given group membership assignment vector, it is easy to write out the complete likelihood function of the network under an assumed distribution on the edge set (e.g., Bernoulli, binomial, Poisson, normal, etc.).

Let $\mathbf{z}_{(k)}$ denote an N x 1 group membership vector with elements taking on values $1, 2, ..., k$. In particular, the element $z_{i,(k)}$ represents the group assignment for node $i \in \{1, ..., N\}$. Further let $\boldsymbol{\theta} = [\theta_1, ..., \theta_k, \theta_b]$ denote a $k + 1$ unknown parameter vector, where element θ_h, $h \in \{1, ..., k\}$, denote the expected edge value between any two nodes belonging to group h and θ_b denotes the expected edge value between any two nodes not belonging to the same group. Finally, let $\mathbf{Y}|\mathbf{z}_{(k)} = [Y_1, ..., Y_k, Y_b]$ be a vector of sufficient statistics for the parameter vector $\boldsymbol{\theta}$, conditioned on the group membership assignment vector $\mathbf{z}_{(k)}$, and let $\mathbf{y}|\mathbf{z}_{(k)}$ denote a realization of $\mathbf{Y}|\mathbf{z}_{(k)}$. Then, by assumptions A1 and A3 above, the likelihood function for $\boldsymbol{\theta}$, given $\mathbf{z}_{(k)}$ and $\mathbf{y}|\mathbf{z}_{(k)}$, can be written as

$$L(\boldsymbol{\theta}|\mathbf{y}, \mathbf{z}_{(k)}) = \{\prod_{h=1}^{k} f_h(y_h|\theta_h, \mathbf{z}_{(k)})\} f_b(y_b|\theta_b, \mathbf{z}_{(k)}), \tag{2.4}$$

where $f_h(y_h|\theta_h, \mathbf{z}_{(k)})$ is the probability density (or mass) function for the sufficient statistic Y_h, $h \in \{1, ..., k\}$, and $f_b(y_b|\theta_b, \mathbf{z}_{(k)})$ is the probability density (or mass) function for the sufficient statistic Y_b.

For example, consider a network with edge values equal to 1 or 0, suggesting the existence or nonexistence of an edge between nodes, respectively. One could then model the edges between nodes contained in group h as Bernoulli(θ_h), $h \in \{1, ..., k\}$, and the edges between nodes not contained in the same group as Bernoulli(θ_b). As such, the sufficient statistics Y_h, $h \in \{1, ..., k\}$, are defined as the sum of the edges that exist between nodes contained in group h and Y_b is defined as the sum of the edges that exist between nodes not contained in the same group. Let N_h denote the total number of possible edges between nodes contained in group h and N_b the total number of possible edges between nodes not contained in the same group, where N_h and N_b are fixed. Then the likelihood function in Equation 2.4 can be written explicitly as

$$L(\boldsymbol{\theta}|\mathbf{y}, \mathbf{z}_{(k)}) = \prod_{h=1}^{k} \theta_h^{y_h} (1 - \theta_h)^{N_h - y_h} \theta_b (1 - \theta_b)^{N_b - y_b} \tag{2.5}$$

since the Y_h's are binomial (N_h, θ_h), Y_b is binomial (N_b, θ_b), and the Y_h's and Y_b are all mutually independent by assumption A3.

The general clustering problem considered in this paper can be stated as follows. Given an observed (symmetric) adjacency matrix $\mathbf{A}$ and number of clusters k, the objective is to find the group membership vector $\mathbf{z}^*_{(k)} \in \mathbf{Z}_{(k)}$ and corresponding parameter vector $\hat{\boldsymbol{\theta}}(\mathbf{z}^*_{(k)}) \in \boldsymbol{\Theta}(\mathbf{Z}_{(k)})$, where $\mathbf{z}^*_{(k)}$ and $\hat{\boldsymbol{\theta}}(\mathbf{z}^*_{(k)})$ are the maximizers of $L(\mathbf{z}_{(k)}, \boldsymbol{\theta}|\mathbf{y}(\mathbf{A}))$, with $\mathbf{Z}_{(k)}$ denoting the set of all possible partitions of N vertices into k groups and $\mathbf{y}(\mathbf{A})$ is the vector of sufficient statistics computed from the observed adjacency matrix. Recall that L was defined generally in Equation 2.4, so that the problem can be written as

$$\mathbf{z}^*_{(k)} = \arg \max_{\mathbf{z}_{(k)} \in \mathbf{Z}_{(k)}} [\max_{\boldsymbol{\theta}(\mathbf{z}_{(k)}) \in \boldsymbol{\Theta}_k(\mathbf{Z}_{(k)})} (\prod_{h=1}^{k} f_h(\mathbf{z}_{(k)}, \theta_h|y_h) f_b(\mathbf{z}_{(k)}, \theta_b|y_b))]. \tag{2.6}$$

Note that, for undirected/unweighted networks, such as those subsequently considered in this paper, the general problem in Equation 2.6 can be written explicitly as

$$\mathbf{z}^*_{(k)} = \arg \max_{\mathbf{z}_{(k)} \in \mathbf{Z}_{(k)}} [\prod_{h=1}^{k} \theta_h^{y_h} (1-\theta_h)^{N_h - y_h} \theta_b (1-\theta_b)^{N_b - y_b}]. \tag{2.7}$$

where $\hat{\theta}_h = y_h/N_h$, $h \in \{1, ..., k\}$, and $\hat{\theta}_b = y_b/N_b$ are the maximum likelihood estimators of the unknown parameters for a given $\mathbf{z}_{(k)}$. Further, as noted earlier, y_h, $h \in \{1, ..., k\}$, and y_b are the sufficient statistics and are defined as the number of observed edges between nodes contained in group h and the number of observed edges between nodes not contained in the same group, respectively.

As with maximizing the modularity metric, the optimization problem given in Equation 2.6 is challenging, particularly due to the combinatorial explosion for large N and k. In general, the problem is NP hard even for moderately-sized networks. As a consequence, throughout this effort, we employ a simulated annealing (SA) algorithm to effectively search the partition space $\mathbf{Z}_{(k)}$ in attempts to locate $\mathbf{z}^*_{(k)}$ for any given objective function being optimized.

Although there are several alternatives for searching the partition space, the SA was chosen due to its simplicity and ease of programming.

We should point out that the proposed model is similar to the stochastic block model of Snijders and Nowicki (1997); however, these authors proposed a generative model that relies on Bayesian Markov Chain Monte Carlo (MCMC) methods to estimate parameters of the posterior distribution $\mathrm{P}(\mathbf{Z}_{(k)})$. Further, the proposed model is also similar to the mixed model approach of Zanghi et al. (2007), which maximizes the complete joint likelihood of the observations $\mathbf{Y}$ and the missing data $\mathbf{Z}_{(k)}$ using a modified expectation-maximization (EM) algorithm. In both of these papers, $\mathbf{Z}_{(k)}$ is treated as a random vector, whereas, in the proposed method, $\mathbf{z}_{(k)} \in \mathbf{Z}_{(k)}$ is treated as an unknown parameter vector that needs to be estimated. Thus, the computational problem involves searching the vertex set partition space to locate the group membership assignment vector that maximizes the likelihood of $\mathbf{z}_{(k)}$ given $\mathbf{y}$. Fortunately, this task can be accomplished with well-known optimization heuristics that are easily implemented in practice, such as simulated annealing and genetic algorithms.

There are some significant advantages to employing the proposed objective function in practice. First, since the proposed objective function is in fact a likelihood function, well-known model selection methods can be directly applied to estimate the number of groups, which is often unknown a priori in practice. For example to determine the "best" number of groups k, one can compute L^* for a range of values for k, where L^* denotes the maximum likelihood value obtained over $\mathbf{z}_{(k)}$ and corresponding θ. Since when k increases, so does the number of parameters (and hence $L^*(k) > L^*(k')(k' < k)$), one cannot simply choose the value of k that yields the maximum L^*. Instead, one can use, e.g., the Aikaike Information Criterion (AIC) proposed by Akaike (1974) or the Bayesian Information Criterion (BIC) proposed by Schwarz (1978) to choose the "best" k. Since the BIC yields a greater penalty for larger k, relative to the AIC, we chose to use the BIC in this paper. In particular, we choose the value of k that

minimizes

$$BIC(k) = -2 \ln L^*(k) + (k+1) \ln[\frac{1}{2}n(n-1)]. \tag{2.8}$$

Another significant advantage to using the proposed objective function is that it directly permits the development of a formal statistical test on the detected clusters by way of the likelihood ratio. Although the papers by Snijders and Nowicki (1997) and Zanghi et al. (2007) elegantly address the general clustering problem, neither paper formally addresses the problem of determining the significance of the detected clusters, relative to some hypothesized null model. In the next section we propose using a likelihood ratio to test the significance of k detected clusters and derive an approximation to the distribution of the likelihood ratio test statistic under the null hypothesis of a single cluster. The information provided by such a test can be used to quantify the chance of observing the given clusters if the observed network were truly generated at random.

2.4 A likelihood ratio test for detected clusters

The likelihood ratio test is a natural fit for the sort of problem considered in this manuscript. In general, such a test provides a means for comparing the goodness-of-fit of a more complex model to that of a less complex model in terms of the number of parameters. The more complex model includes more parameters, while the less complex model includes fewer parameters. In particular the two models are nested. For a good review of the likelihood ratio test and related statistical procedures, the reader is referred to any textbook on mathematical statistics, e.g. Hogg et al. (2005).

In the case of determining the significance of k detected clusters in network data, let $\mathbf{Y}$ and $\mathbf{z}_{(k)}$ be defined as in Section 2.3, where, independent of $\mathbf{Y}$, $z_{i,(k)}$ takes on integer values between 1 and k. The null hypothesis that the data are independently and identically distributed is tested against the alternative hypothesis that the observations, while independent and coming from the same

family of distributions, are not identically distributed; that is, the parameters of the distribution are different depending upon group membership.

Let $L_0(\theta_0|\mathbf{y})$ denote the likelihood function under the null hypothesis of a single cluster and $L_1(\boldsymbol{\theta}|\mathbf{y}, \mathbf{z}_{(k)})$ denote the likelihood function under the alternative hypothesis of k clusters. Here, θ_0 denotes a single parameter and $\boldsymbol{\theta}$ denotes the $k+1$ dimensional parameter vector previously defined in Section 2.3. By assigning group membership independently of the sample, the likelihood ratio of the unknown parameters can be written generally as

$$\Lambda(\theta_0, \boldsymbol{\theta}|\mathbf{y}, \mathbf{z}_{(k)}) = \frac{L_0(\theta_0|\mathbf{y})}{L_1(\boldsymbol{\theta}|\mathbf{y}, \mathbf{z}_{(k)})}. \tag{2.9}$$

The likelihood ratio test is conducted by computing the quantity $D = -2\ln\Lambda^*$, where

$$\Lambda^* = \frac{L_0^*}{L_1^*} \tag{2.10}$$

and L_0^* and L_1^* are the maximized likelihoods under the null and alternative hypotheses, respectively. For even moderately sized networks, the statistic D is approximately χ^2 with k degrees of freedom under the null hypothesis of a single cluster.

To illustrate, consider an undirected binary network. In this case, the null hypothesis states that the probability of observing an edge between any two nodes is simply θ_0, while the alternative hypothesis states that the probability of observing an edge between any two nodes is not equal to θ_0, but, more specifically, is dependent upon group membership of the nodes. To parameterize this, let y_h denote the number of observed edges between nodes assigned to group h, $h \in \{1, ..., k\}$, and y_b denote the number of observed edges between nodes not assigned to the same group. Then, the likelihood function specified under the alternative hypothesis can be written explicitly as

$$L(\boldsymbol{\theta}|\mathbf{y}, \mathbf{z}_{(k)}) = \prod_{h=1}^{k} \theta_h^{y_h}(1-\theta_h)^{N_h - y_h}\theta_b(1-\theta_b)^{N_b - y_b} \tag{2.11}$$

for $0 \leq \theta_h, \theta_b \leq 1$, where $y_h \in \{0, 1, ..., N_h\}$, $N_h = \frac{1}{2}n_h(n_h - 1)$ and n_h is the total number of nodes assigned to group h. Also, $y_b \in \{0, 1, ..., N_b\}$, where N_b denotes the total number of possible edges between nodes not assigned to the same group. Also, define

$$y = \sum_{h=1}^{k} y_h + y_b$$

and

$$y = \sum_{h=1}^{k} N_h + N_b$$

where y and N denote the total number of observed edges and the total number of possible edges in the network, respectively. Then, the likelihood function under the null hypothesis can be written explicitly as

$$L_0(\theta_0|y) = \theta_0^y(1 - \theta_0)^{N-y} \tag{2.12}$$

so that, for a given group membership vector $\mathbf{z}_{(k)}$, the ratio of the maximized likelihoods is given by

$$\Lambda^*(\mathbf{y}|\mathbf{z}_{(k)}) = \frac{\hat{\theta}_0^y(1 - \hat{\theta}_0)^{N-y}}{\prod_{h=1}^{k} \hat{\theta}_h^{y_h}(1 - \hat{\theta}_h)^{N_h - y_h}\hat{\theta}_b(1 - \hat{\theta}_b)^{N_b - y_b}} \tag{2.13}$$

where $\hat{\theta}_0 = y/N$ denotes the maximum likelihood estimate of θ_0 under the null hypothesis, and $\hat{\theta}_h = y_h/N_h$, $\{h = 1, ..., k\}$, and $\hat{\theta}_b = y_b/N_b$ denote the maximum likelihood estimates of the unknown parameters under the alternative hypothesis.

It is important to note that the likelihood ratio given generally in Eq. 2.9 is conditional on the group membership vector $\mathbf{z}_{(k)}$. In cases where attribute information is available at the vertices, and this information is used to specify $\mathbf{z}_{(k)}$ explicitly, then the group membership vector is known and the assumption that group membership is assigned to the vertices independent of $\mathbf{y}$ is valid. However, when $\mathbf{z}_{(k)}$ is unknown (which is the assumption in this effort), then is must be estimated from $\mathbf{y}$. As a result, group membership assignment will not be independent of the vector of sufficient statistics. Consequently, when $\mathbf{z}_{(k)}$ is

unknown and must be estimated using $\mathbf{y}$, the likelihood ratio statistic $D = -2 \ln \Lambda^*$ is not asymptotically χ^2_k under the null hypothesis. Since our aim is to estimate (and subsequently test) the unknown group membership assignments, in what follows we discuss a novel approach to the approximation of the null distribution of D when $\mathbf{z}_{(k)}$ is unknown.

It should be clear that, since the number of possible combinations of $\mathbf{z}_{(h)}$ is finite for a fixed N and k, the population of possible values of D is also finite for a fixed N and k. Let P denote the population of all possible values of D for a fixed N and k. The most interesting member of P is D_{max} where $D_{max} \geq D_l$(for all l), where $l = 1, ..., S(N, k)$ and $S(N, k)$ denotes the Stirling number of the second kind, i.e. the number of ways to partition N elements into k subsets. Explicitly, $S(N, k) = \frac{1}{k!} \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} j^m$.

The practitioner could simply generate random combinations of $\mathbf{z}_{(k)}$ in hopes of "stumbling" upon D_{max}. As the number of randomly generated $\mathbf{z}_{(k)}$'s increases, the probability of having selected the correct group membership vector, and hence D_{max}, increases. This approach is obviously problematic, however, even for moderately-sized networks. For example, suppose the practitioner suggests that a network of $N = 100$ nodes forms $k = 5$ clusters. The number of possible partitions is $S(100, 5) = 6.57$ x 10^{67}. Sampling enough times to have even a chance at selecting D_{max} is clearly infeasible.

Nonetheless, there is hope for producing an estimate for the null distribution of D_{max}. Let the G x 1 vector $\mathbf{D}$ denote a random sample *with replacement* from P. If D_{max} is included in this sample, then the sample maximum, $D_{(G)} = D_{max}$; if not, $D_{(G)} < D_{max}$. Intuitively, the larger the sample, the more likely that D_{max} is selected.

Since the sample is taken with replacement, the number of draws, M, until D_{max} is selected follows a negative binomial distribution with probability of selection $p = \frac{1}{S(N,k)}$ and the number of successes $r = 1$. The expected number of

draws until success then is $E(M) = \frac{1-p}{p} = S(N,k) - 1$, and

$$P(N,k,M) = 1 - (1 - \frac{1}{S(N,k)})^M$$

where $P(N,k,M)$ is the probability that D_{max} has been selected at least once after M draws.

Since the $\mathbf{z}_{(k)}$'s (and hence the D's) are randomly selected, group assignments are made independent of the sample. Consequently, under the null hypothesis, the asymptotic distribution of each $D_i \in \mathbf{D}$ is χ^2_k. As such, actually simulating random combinations of $\mathbf{z}_{(k)}$ is unnecessary; that is the pdf of the maximum of a random sample of size G from a χ^2 distribution is

$$f_{D_{(G)}} = GF(u)^{G-1}f(u) \tag{2.14}$$

where $F(u)$ and $f(u)$ are the cdf and pdf of the χ^2_k distribution, respectively. Because the support of the χ^2 distribution is unbounded on the right, increasing values of G produce increasing values of $D_{(G)}$. By choosing an appropriate value for G an estimate for the distribution of D_{max} under the null hypothesis can be obtained. Recalling that $E(M) = S(N,k) - 1$ is the expected number of draws required to select the maximum at random, set $G = E(M)$. It is important to note that the estimate for D_{max} (and thus, its distribution) is downwardly biased; that is $D_{(G)} \leq D_{max}$, with equality achieve when $D_{max} \in \mathbf{D}$.

Hence, the $100(1-\alpha)$th percentiles of the distribution of $D_{(G)}$ provide approximate critical values for a one-tailed test for the significance of detected clusters. The size of the test, however, will be somewhat less than α in the case that the search algorithm used fails to find the group membership assignment vector that maximizes the likelihood function specified under the alternative hypothesis.

Based on Eq. 2.14, approximate critical values then are calculated from

$$C_{1-\alpha}(N,k) = F^{-1}(\sqrt[G]{1-\alpha}) \tag{2.15}$$

where if $k = 2$, a closed-form expression is given explicitly by

$$C_{1-\alpha}(N,k) = -1\log_e(1 - \sqrt[G]{1-\alpha}). \tag{2.16}$$

For many networks, 1 - α is exceedingly close to 1, often too close for common statistical computing packages. The critical values displayed in the Appendix were calculated using MapleSoft (2011), which allows for variable precision in computation. Additionally, one can find a computation recipe for computing these extreme values of the χ^2 distribution and others in, e.g., Press et al. (2007).

In practice, the test is conducted by computing $D = \ln \Lambda^*$ for a given N and "best" k, and subsequently comparing to the critical value $C_{1-\alpha}(N,k)$. If $D > C_{1-\alpha}(N,k)$, then the test concludes in favor of the alternative hypothesis of k clusters. That is, it is unlikely that the groups detected by the clustering effort could have occurred by chance.

In the next section, we cluster two real-world networks in efforts to demonstrate exactly how the proposed clustering approach is applied in practice. In particular, we consider the well-known karate club network of Zachary (1977) and the terrorist acquaintance network first published by Sageman (2004).

2.5 Application to real-world networks

In this section, we apply our proposed clustering framework to the karate club friendship network of Zachary (1977) and the terrorist acquaintance network of Sageman (2004). For comparison purposes, we also find the group membership assignment vectors that maximize the modularity metric for both networks. A

simulated annealing algorithm[1] (see Kirkpatrick et al. (1983) and Cerny (1985), amongst others) was used in finding the "best" group assignment vectors when using both, the proposed and modularity objective functions.

To measure the similarities between the proposed and modularity group membership assignments found for each network, we computed the adjusted mutual information (AMI) between the two clusterings. The AMI is an information theoretic measure for comparing clusterings. It is "adjusted for chance" (i.e., corrected for randomness), where if the two clusterings are identical the AMI is equal to 1, and if the mutual information between the two clusterings is equal to the expected mutual information of two random clusterings, then the AMI is equal to 0. Thus, similar clusterings will have AMI close to 1. For additional information on the AMI measure used in this paper, as well as other theoretic measures for comparing clusterings, the reader is referred to Vinh and Epps (2010).

2.5.1 Zachary's karate club network

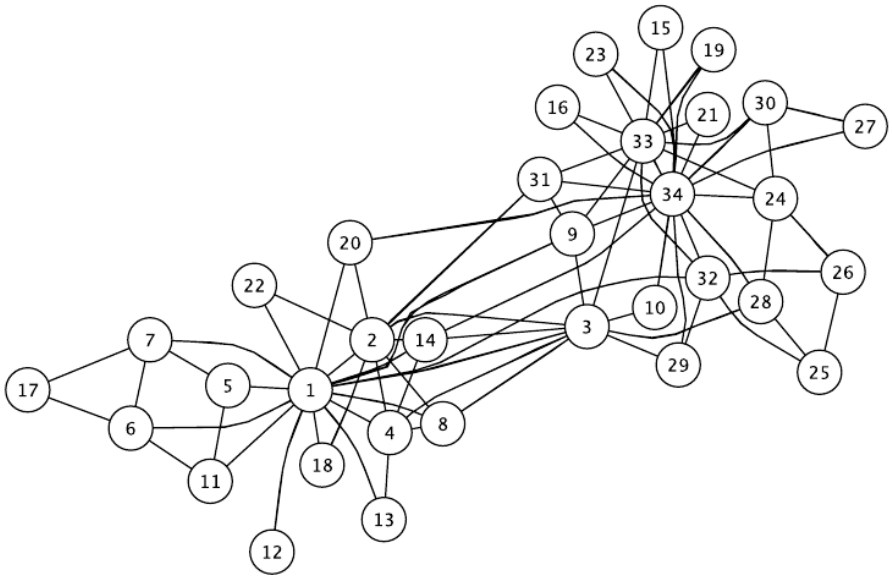


Figure 2.1: Zachary's karate club network

[1]Cooling schedule for maximizing the proposed objective function: Initial temp=0.0025, Cooling Rate: 0.9925, Temp Length: 15000.

To demonstrate our proposed clustering framework, we first consider the 34 node karate club network shown in Figure 2.1. This is a social network, with nodes representing members of a karate club observed by Zachary for roughly two years during the 1970s and edges indicating social interaction between members.

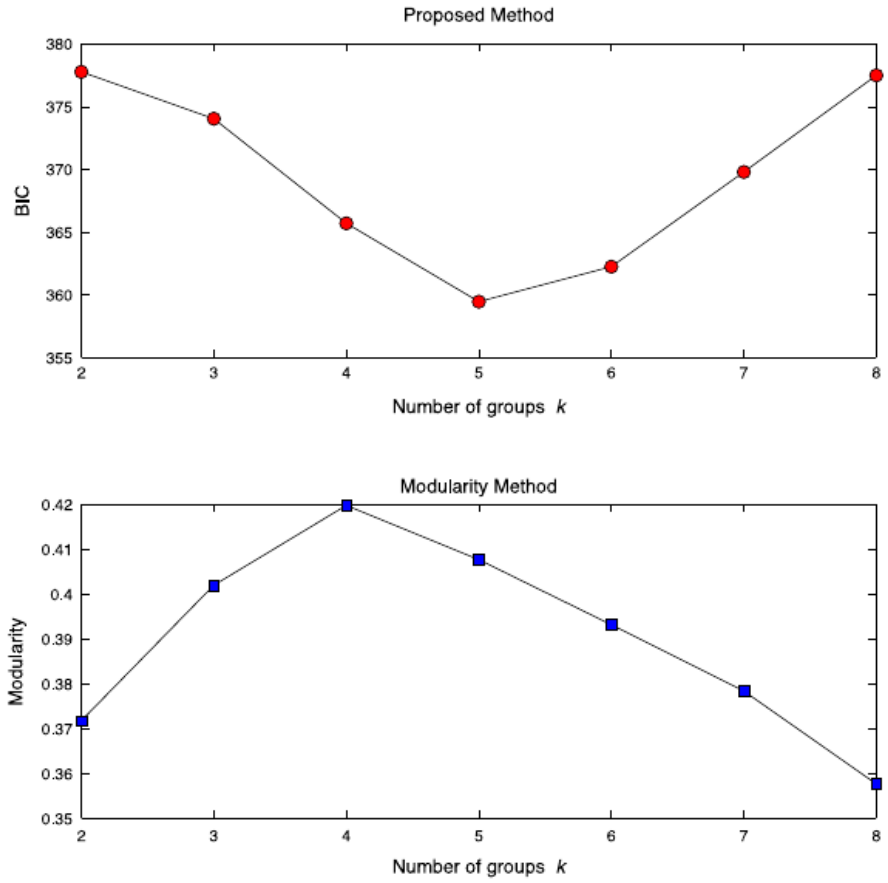


Figure 2.2: Model selection plots for Zachary's karate club network using the proposed method with the BIC (top plot) and the modularity method (bottom plot). The proposed method suggests 5 groups, whereas modularity suggests 4 groups.

Since the number of groups is unknown, then when clustering using the proposed framework, we employ the BIC to provide an estimate of this quantity. When clustering under the modularity objective, it is sufficient to find the number of groups k that maximizes modularity over the set of possible k. Figure 2.2 shows the model selection plots for the karate club network. Notice that using the proposed approach, the estimated number of groups is 5 since the BIC is minimized at $k = 5$. Further, using the modularity objective, the estimated

number of groups is 4 since modularity is maximized at $k = 4$.

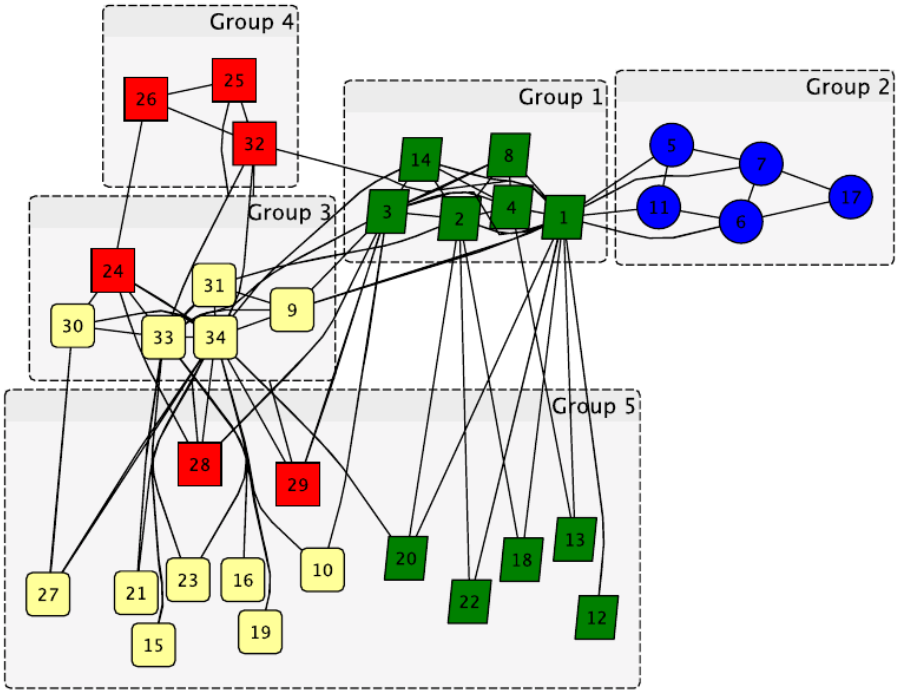


Figure 2.3: Zachary's karate club network clustered via maximizing the proposed objective function and maximizing modularity. The modularity solution is shown by the shape/color of the node and the proposed solution has nodes contained in the large rectangles.

Figure 2.3 shows the "best" solutions using both the proposed method and modularity, where the proposed solution has nodes contained in large rectangles and the modularity solution has nodes grouped by shape/color. The proposed solution seems to suggest that there are four modular groups and an additional group of "periphery" nodes. The characteristics of the periphery group are such that, typically, nodes contained in this group have little relative influence on the overall network (e.g., nodes with low eigencentrality measures). The AMI computed between the two estimated clusterings is 0.4891, which suggests some lack of agreement between the two solutions.

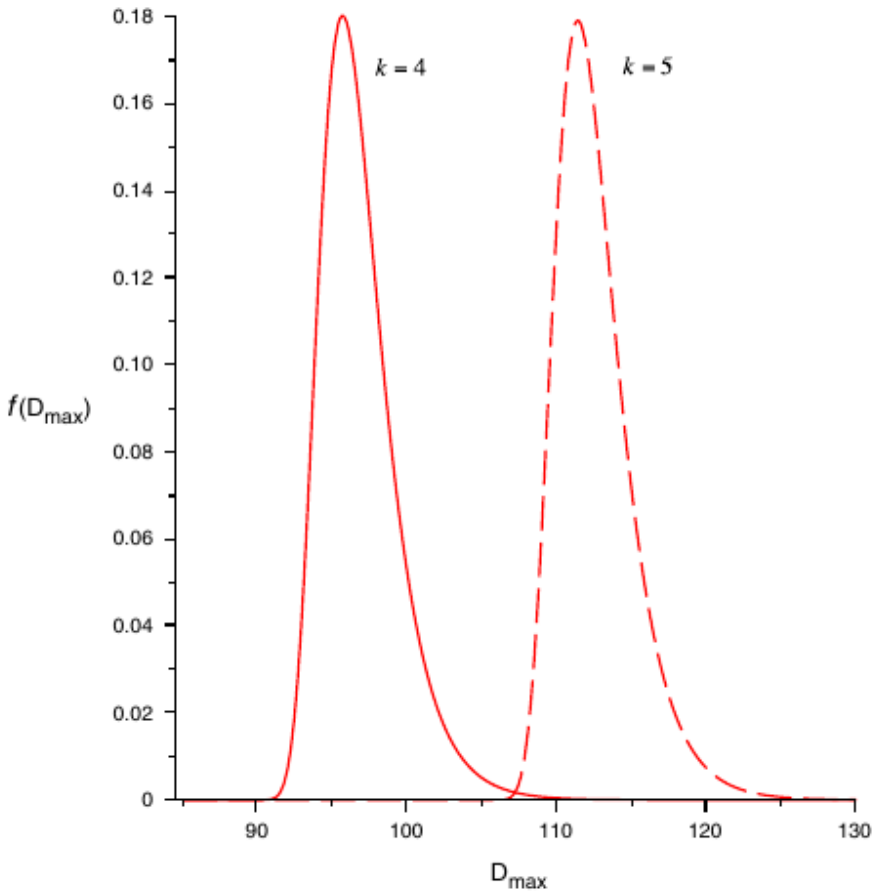


Figure 2.4: Density functions of the test statistic D_{max} under the null hypothesis with $N = 34$ and $k = 4$ and $k = 5$.

If we apply the likelihood ratio test developed in Section 1.3 to the karate club network using the '"best" group membership assignment vector obtained by maximizing modularity, we compute a test statistic of $D_{Mod} = 94.65$. Similarly, the test statistic using the proposed "best" group membership assignment vector is $D_{Lik} = 130.91$. The density functions of the maximum likelihood ratio statistic (i.e., D_{max}) under the null hypothesis with $N = 34$ and $k = 4$ and 5 are shown in Fig. 2.4. The critical value at the 95% significance level for the test using the modularity solution (i.e., $k = 4$) is $C_{0.95}(34, 4) = 101.75$, while the critical value using the proposed solution (i.e., $k = 5$) is $C_{0.95}(34, 5) = 117.50$. Notice that $D_{Lik} = 130.91 > C_{0.95}(34, 5) = 117.5$, with a corresponding p-value of 0.00007381 (so that the test is significant), suggesting a 0.007% chance that the clusters found by maximizing the proposed objective function could have occurred by

chance. Even further, $D_{Mod} = 94.65 < C_{0.95}(34, 4) = 101.75$, with a corresponding p-value of 0.810448 (so that the test is not significant), suggesting an 81% chance that the clusters found by maximizing the modularity metric could have occurred by chance. Thus, relative to modularity, there is much more evidence in support of the proposed solution.

It is important to note that the karate club later split into two factions following a disagreement between node 1 and node 34, and these two factions are often used as the "ground truth" communities in benchmark studies. We note that our proposed solution is very similar to the communities extracted by Zhao et al. (2011), with the exception of one additional subcommunity. In general, our method identified the cores of the two true factions (i.e., Group 1 and Group 3), as well as two tighter subcommunities (i.e., Group 2 and Group 4).

2.5.2 Sageman's terrorist acquaintance network

In this subsection, the $N = 210$ node terrorist acquaintance network of Sageman (2004) shown in Figure 2.5. is considered, where a link between two nodes suggests that the actors are acquaintances.

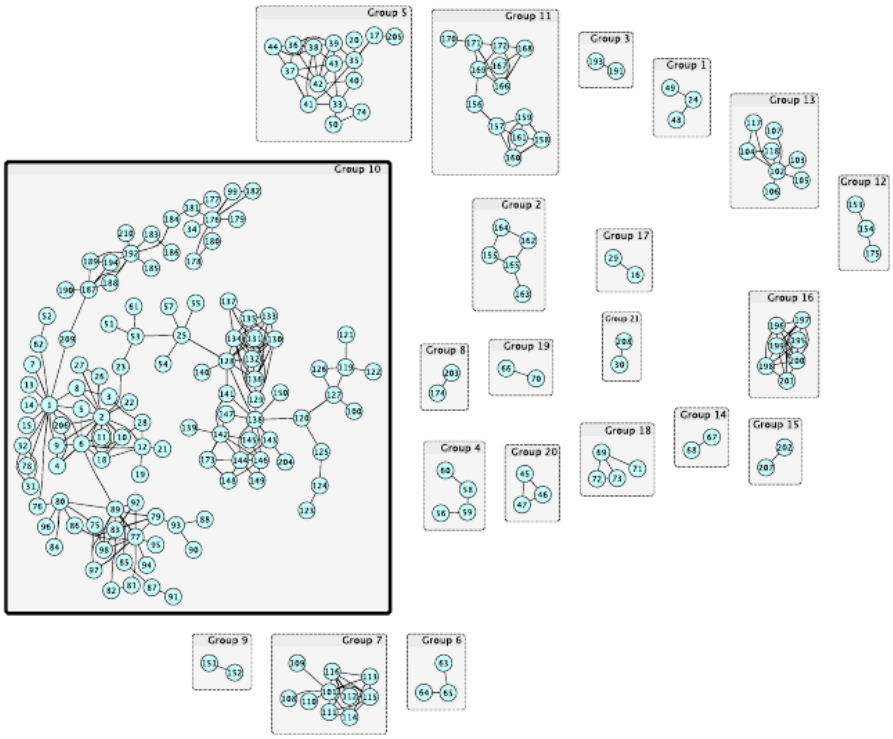


Figure 2.5: Sageman's terrorist acquaintance network (connected components). The giant connected component is outlined in solid black.

Notice that this network is comprised of 21 different connected components; thus in order to reduce the computational complexity of the problem, one can cluster each connected component independent of the others without losing any information. In this paper, we cluster the giant connected component, which has 125 nodes (i.e. the component in Figure 2.5 outlined in a solid black line.)

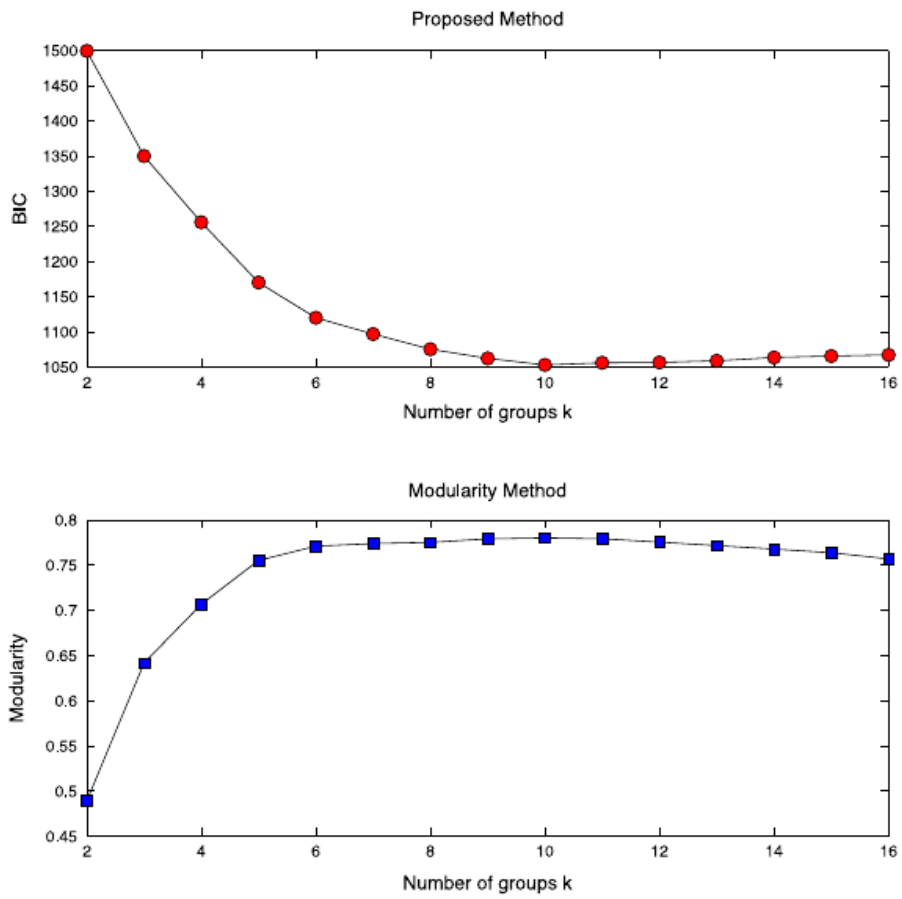


Figure 2.6: Model selection plots for Sageman's terrorist acquaintance network using the proposed method with the BIC (top plot) and the modularity method (bottom plot). Both methods suggest 10 groups.

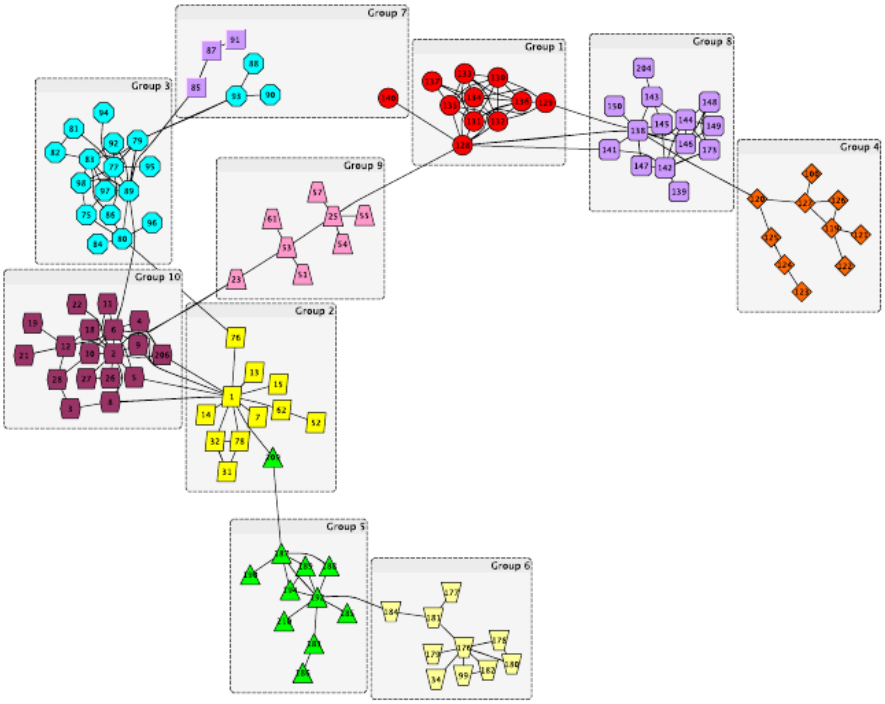


Figure 2.7: Giant connected component of Sageman's terrorist acquaintance network clustered via maximizing the proposed objective function and maximizing modularity. The modularity solution is shown by the shape/color of the node and the proposed solution has nodes contained in the larger rectangles.

Figure 2.6 shows the best model selection plots for this network, where both methods suggest that the number of groups is $k = 10$. Figure 2.7 shows the "best" solutions obtained using both the proposed method and modularity, where, as before, the proposed solution has nodes contained in large rectangles and the modularity solution has nodes group by shape/color. There appears to be a significant amount of agreement between the two solutions for this network, which is evidence by an AMI measure of 0.9296. One obvious difference, however, is the group of periphery nodes (i.e. Group 7) identified by using the proposed objective function. Again, periphery nodes are those that are less influential and not well connected within the network. A second difference between the two solutions is the classification of node 209, where modularity assigns this node to Group 5 and the proposed method assigns this node to Group 2.

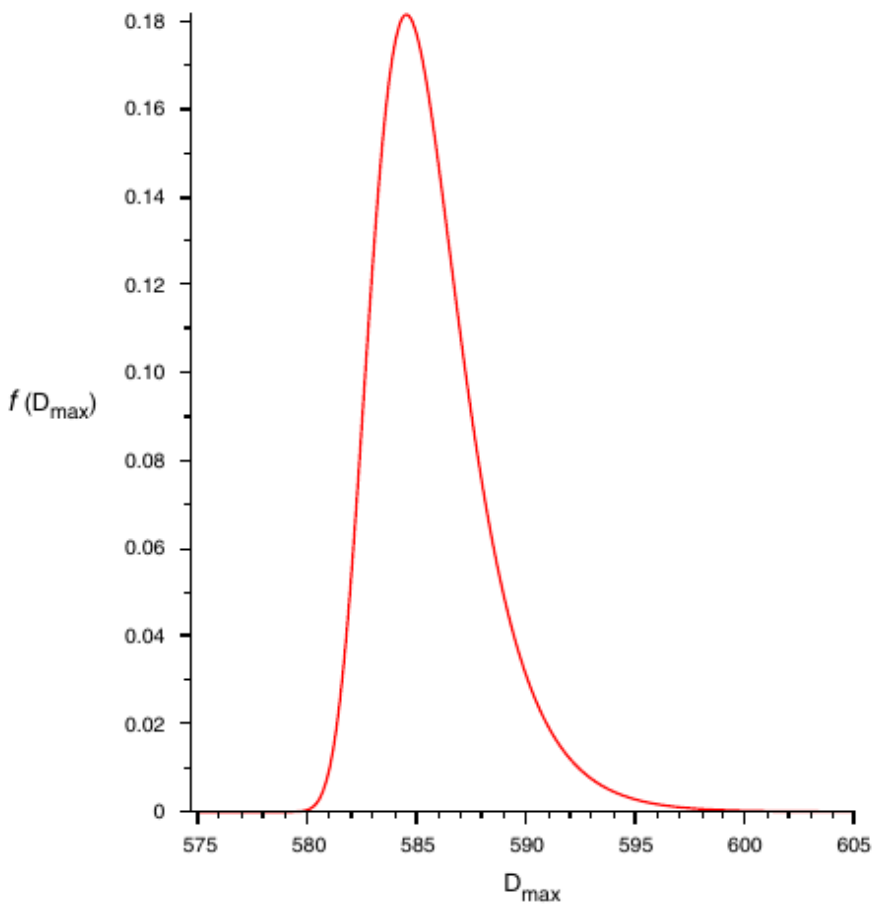


Figure 2.8: Density function of the test statistic D_{max} under the null hypothesis with $N = 125$ and $k = 10$.

If we apply our likelihood ratio test to the terrorist network using the "best" group membership assignment vector obtained by maximizing modularity, we compute a test statistic of $D_{Mod} = 768.59$. Similarly, the test statistic using the proposed "best" group membership assignment vector is $D_{Lik} = 781.62$. The density function of D_{max} under the null hypothesis with $N = 125$ and $k = 10$ is shown in Fig. 8. The critical value at the 95% significance level for the test using either solution is computed as $C_{0.95}(125, 10) = 590.592$ (since both methods suggest $k = 10$), implying that the test is highly significant regardless of which solution is used (i.e., D_{Mod} and D_{Lik} are both significantly greater than $C_{0.95}(125, 10)$). In particular, the p-values for both tests are essentially zero, suggesting that there is practically no chance that the observed clusters (obtained by using either the modularity solution or the proposed solution) could

have occurred simply by chance.

The results in this section suggest that the proposed clustering framework appears to perform quite well on real-world networks. However, for the networks analyzed in this section, the true clusters are actually unknown (with the exception of the split of the karate club network into two factions). Further, since we only have a single realization of each network, it is difficult to assess the clustering performance in an objective way. As a result, in the next section, we use Monte Carlo simulation to assess the clustering performance of the proposed objective function, as well as the power of the proposed statistical test, when applied to the LFR benchmark graphs by Lancichinetti et al. (2008), since these graphs possess the properties of real-world networks.

2.6 Performance evaluations

In this section we evaluate the clustering performance of the proposed objective function, as well as the power achieved by the proposed likelihood ratio test, when applied to the so called LFR benchmark graphs proposed by Lancichinetti et al. (2008). These graphs possess some properties of real-world networks, specifically, non-homogeneous degree distributions and community sizes. In the next subsection, we provide a brief description of the LFR benchmark graphs, to include the parameters involved in generating these graphs. Subsequently, we discuss the results of a simulation study used to assess the performance of the proposed clustering framework.

2.6.1 LFR benchmark graphs

Proposed by Lancichinetti et al. (2008) as a means to testing community detection algorithms, the LFR benchmark graphs possess several properties of real-world networks. In particular, most networks of interest have non-homogeneous (or skewed) degree distributions and non-homogeneous community (or cluster) sizes. For the LFR benchmark graphs, the degree

distribution follows a power law with parameter $2 \leq \gamma \leq 3$ and the community size distribution follows a power law with parameter $1 \leq \beta \leq 2$, which encompasses a vast array of graphs with non-homogeneous degree and community size distributions. Further, there are three additional parameters: (1) the mixing parameter μ, which represents the fraction of a node's edges connected to other nodes not contained in the same group (and thus, $1 - \mu$ represents the fraction of a node's edges connection to other nodes contained in the same group), (2) the average degree of the network, or *Ave Degree*, and (3) the maximum degree of the network, or *Max Degree*. Note that when *Ave Degree* = *Max Degree*, then all nodes have the same degree and the degree distribution is homogeneous.

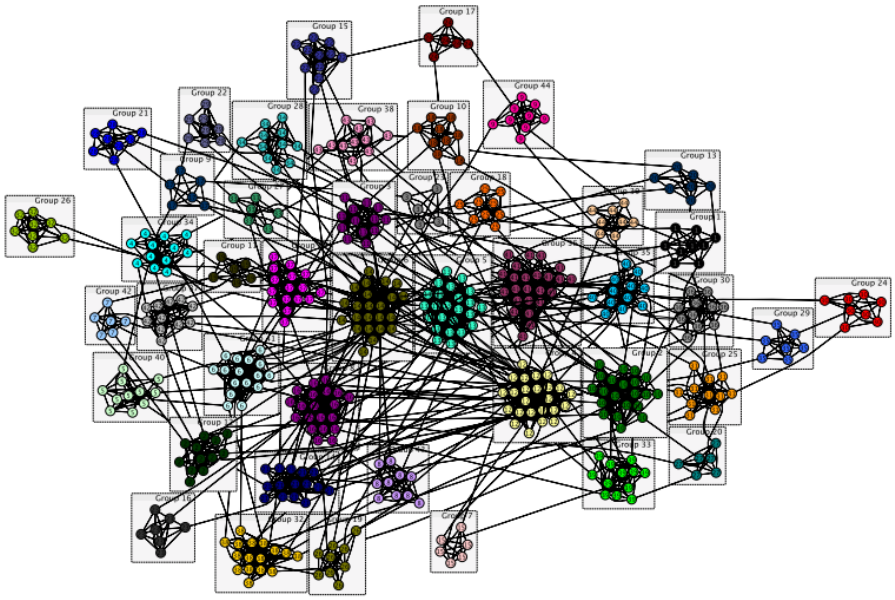


Figure 2.9: A single realization of an LFR benchmark graph (with the $k = 44$ true communities identified) with $N = 500$, $\gamma = 2$, $\beta = 1$, $\mu = 0.1$, *Ave Degree* $= 10$, and *Max Degree* $= 25$

To illustrate, Figure 2.9 shows a single realization of a 500 node LFR benchmark graph with $\gamma = 2$, $\beta = 1$, $\mu = 0.1$, *Ave Degree* $= 10$, and *Max Degree* $= 25$. In addition to the nodes and edges, the algorithm that generates the benchmark also provides the true community structure, as well as some network statistics as output. For example, the true community structure of the network is also shown in Figure 2.9, while Figure 2.10 shows plots of the descriptive statistics for this network. Notice that there are $k = 44$ groups for

this network, having a variety of different sizes. In Figure 2.10, notice the skewed degree distribution in the top plot, which is a common property of real-world networks. Also, the middle plot shows a histogram of the fraction of nodes having a given μ value. It is important to note that, in general, the algorithm that generates the benchmark attempts to set the μ-value of each node to the pre-specified input value. However, this cannot always be done, particularly for nodes with small degree. Thus, the distribution of the μ-values has a bell-shaped curve with a pronounced peak, such that the average of the μ-values is approximately equal to the pre-specified input value of μ. The bottom plot in Figure 2.10 simply illustrates the non-homogeneity of the sizes of the different communities in the network by showing a stem plot of the sizes of the communities versus the number of occurrences.

Since the LFR benchmark graphs possess properties inherent in real-world networks, we chose to evaluate the performance of our proposed clustering framework when applied to these graphs. It is important to note that real-world networks are often much more complex than the LFR graphs; however, since these graphs exhibit skewed degree distributions and non-homogeneous community sizes (properties that are inherent in a large number of real-world networks) they serve as adequate benchmarks for evaluating network clustering algorithms. In the next subsection we discuss the results of an extensive Monte Carlo simulation study used to evaluate the clustering and power performances of the proposed clustering framework, relative to that achieved by maximizing modularity.

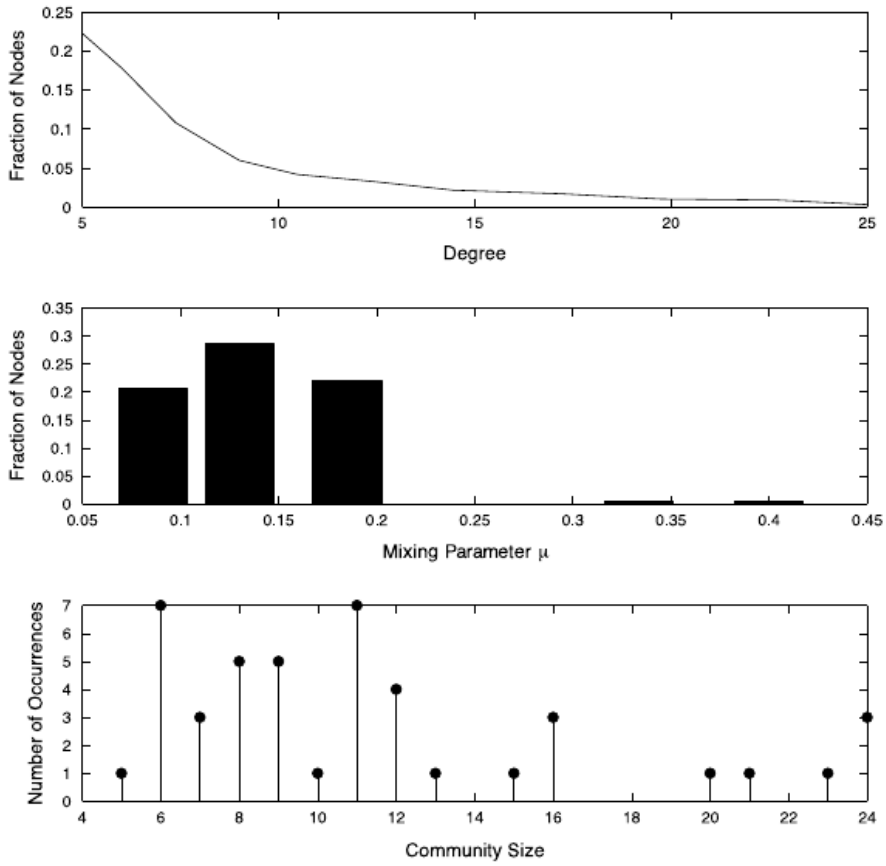


Figure 2.10: Network statistics for the single realization LFR benchmark graph in Fig. 2.9 with $N = 500$, $\gamma = 2$, $\beta = 1$, $\mu = 0.1$, *Ave Degree* $= 10$, and *Max Degree* $= 25$

2.6.2 Clustering and power performances of the proposed clustering framework when applied to the LFR benchmark graphs

In this subsection we report the results of a Monte Carlo simulation study where we applied our proposed clustering framework to the LFR benchmark graphs to assess its expected performance. We investigated clustering performance as a result of maximizing the proposed objective function, and for comparison purposes, we also investigated that achieved by maximizing the modularity metric. Additionally, we assessed the power of the proposed likelihood ratio test when the "best" group membership assignment vector (specified under the alternative hypothesis) is obtained through maximization of the proposed objective function, as well as through maximization of modularity. In what follows, details of the simulation model are discussed.

For any given combination of the LFR benchmark parameter settings studied[2], we generated 100 independent benchmark graphs. For each simulated graph and corresponding number of groups k (which was given for any particular graph), we used simulated annealing algorithms to find the group membership vectors that maximize the modularity metric and the proposed objective function, respectively. Once these vectors were found, we then computed the AMI between the true group membership vector (which was given) and the estimated group membership vectors obtained by maximizing the two objective functions. The average of the AMI values obtained via maximizing modularity, as well as that obtained by maximizing the proposed objective function, was then computed over the 100 independent simulated graphs.

We consider networks of size $N = 100$ nodes with *Max Degree* $= 20$ and three different values for *Ave Degree*, and *Ave Degree* $= 5$, 7.5 and 10. For each value of *Ave Degree*, we considered four different combinations for the parameters γ and β; namely $(\gamma, \beta) = (2, 1), (2, 2), (3, 1), (3, 2)$, which encompasses the extremes of the ranges of these parameters. Finally, we considered the values of the mixing parameter μ between 0.1 and 0.6, in increments of 0.05. Figures 2.11 and 2.12 show the estimated AMI curves over the parameter space μ corresponding to modularity and the proposed method, respectively.

[2] i.e., γ, β, μ, *Ave Degree*, *Max Degree*

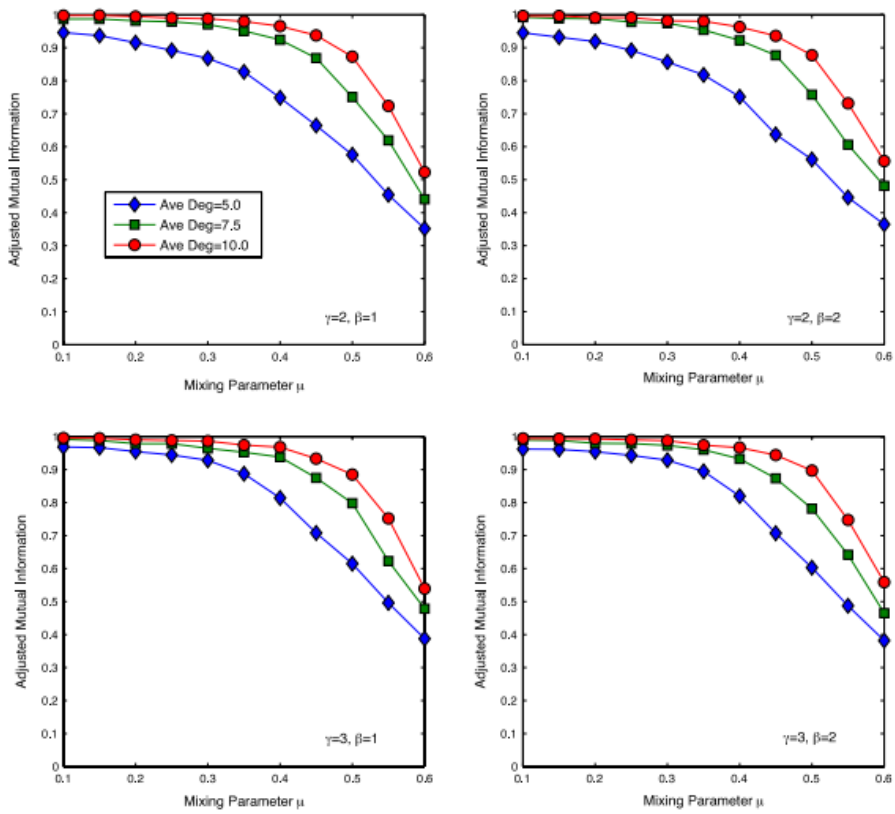


Figure 2.11: Clustering performance evaluation using the modularity objective function. Each point is the average adjusted mutual information over 100 simulated LFR benchmark graphs.

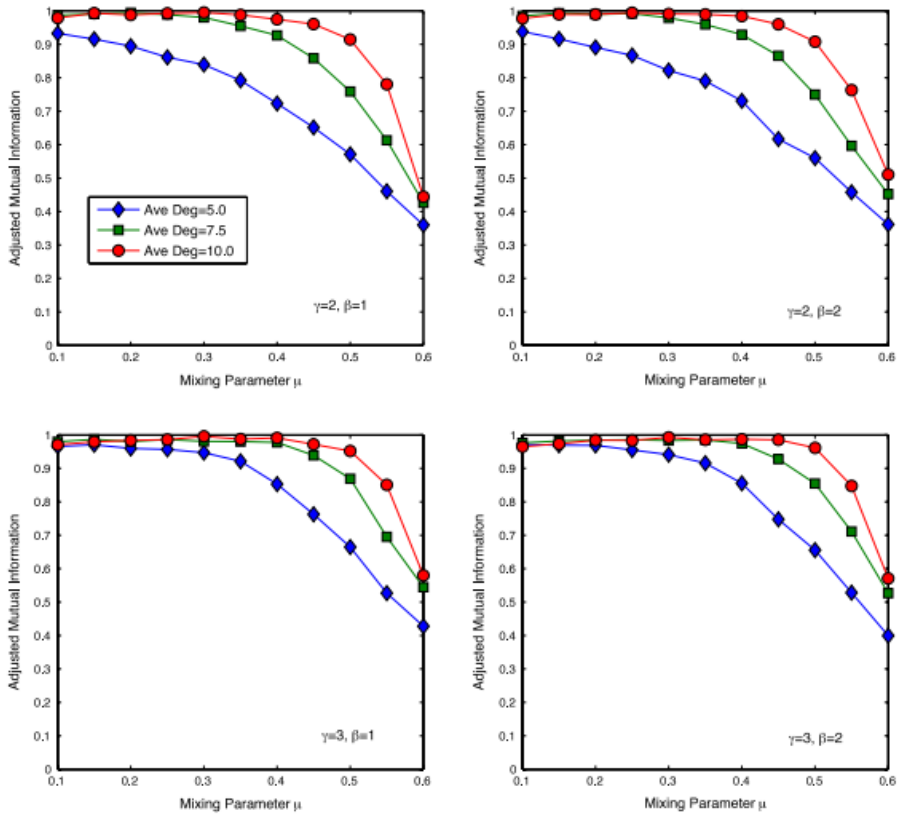


Figure 2.12: Clustering performance evaluation using the proposed objective function. Each point is the average adjusted mutual information over 100 simulated LFR benchmark graphs.

In addition to computing the average AMI, we also estimated the power of the proposed likelihood ratio test over the 100 simulated benchmark graphs, where the "best" group membership assignment vector was obtained either via maximizing modularity, or via maximizing the proposed objective function. Recall that the power of a statistical test is the probability of rejecting the null hypothesis, given that the null hypothesis is false. Figures 2.13 and 2.14 show the estimated power curves corresponding to the modular and proposed solutions, respectively.

From Figures 2.11–2.14, we can observe the following general results. (1) A decrease in the AMI and power performances is observed as the mixing parameter increases. This is intuitive since for all μ, the groups or communities are more densely intra-connected, and sparsely inter-connected (and vice versa for large μ). (2) An increase in AMI and power performances is observed as

Ave Degree approaches *Max Degree*, suggesting that as the degree distribution becomes more homogeneous, an increase in AMI and power performances is observed. (3) An increase in AMI and power performances is observed as the parameter γ increases, which is also intuitive since there is more variability in the degree distribution $\gamma = 2$ than when $\gamma = 3$. (4) No significant effect on AMI or power performances due to changes in the parameter β is observed. Although there does not appear to be a significant effect on performance due to changes in β, it is certainly possible that one might see a significant effect for networks with a greater number of nodes.

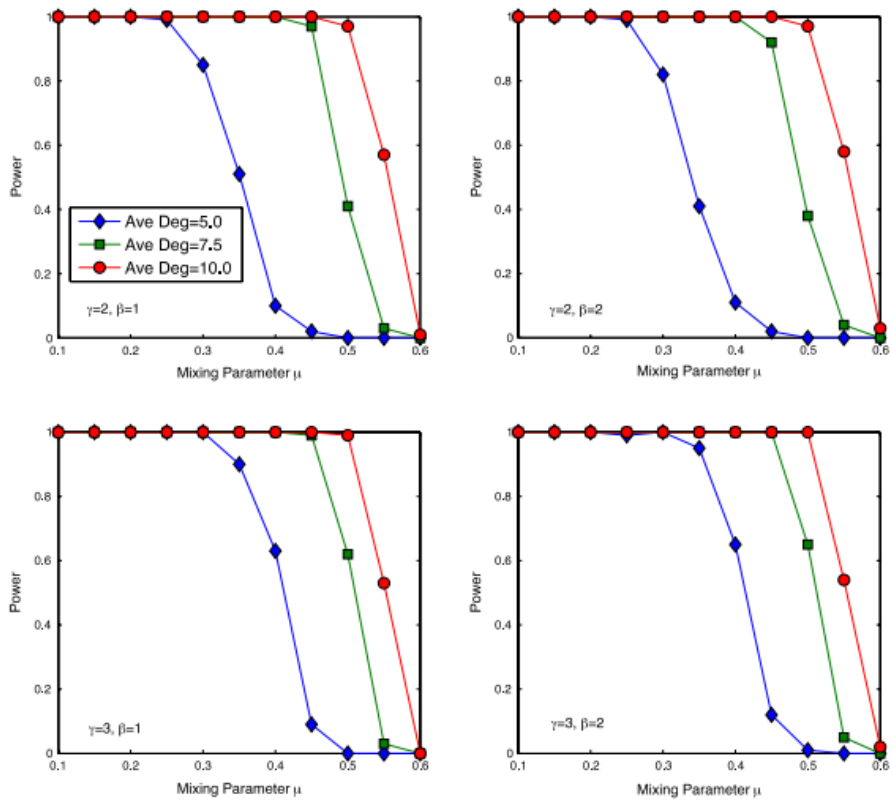


Figure 2.13: Power performance evaluation using the modularity objective function. Each point is estimated over 100 simulated LFR benchmark graphs.

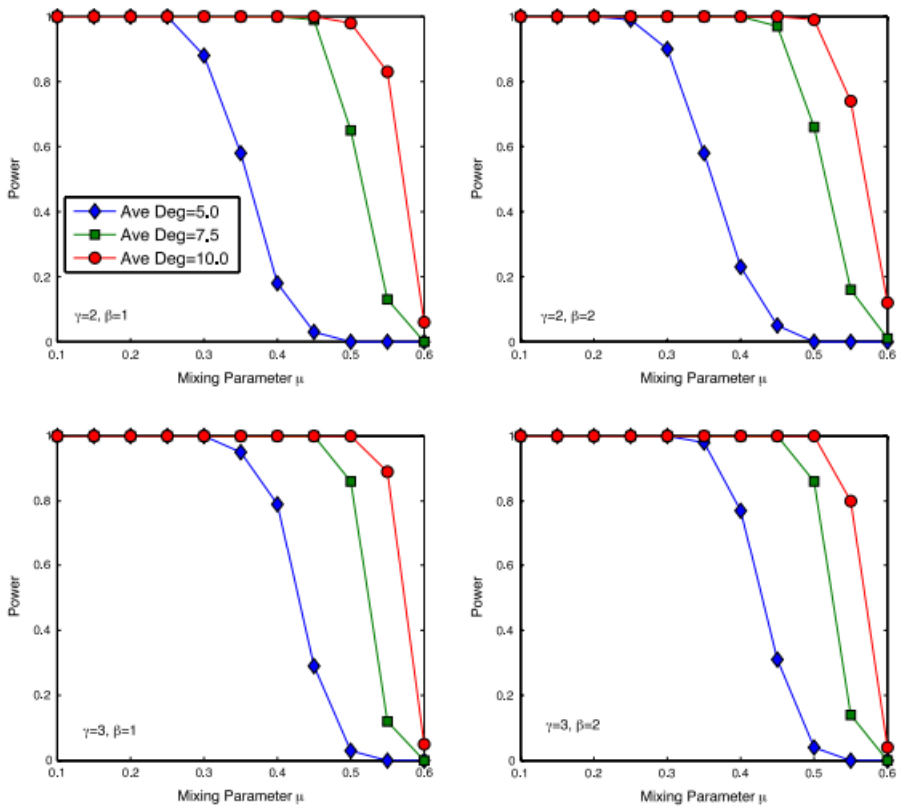


Figure 2.14: Power performance evaluation using the proposed objective function. Each point is estimated over 100 simulated LFR benchmark graphs.

In order to make direct performance comparisons between the proposed and modularity methods, we computed the *relative mean index* (RMI) over the range of values of μ considered in our simulation study, where the RMI for the vth method being compared (e.g., $v = [1, 2]$), where $v = 1$ corresponds to the modularity method and $v = 2$ corresponds to the proposed method) is calculated by

$$RMI_v = \frac{1}{r}\sum_{i=1}^{r}(m_i^* - m_{iv})(m_i^*)^{-1}, \tag{2.17}$$

where r denotes the number of levels of μ considered in the simulation study (e.g., $r = 11$ in our study since $\mu = 0.1, ..., 0.6$ in increments of 0.05). If we are comparing AMI performance, then m_{iv} is the AMI value at the ith level of μ and for the vth method being compared, and m_i^* is the largest AMI between the two methods at the ith level of μ. If we are comparing power performance, then m_{iv} and m_i^* are the corresponding power values. The method that yields the smallest

RMI value is most desirable, since this suggests better performance across the range of values of the mixing parameter μ.

Table 2.1 shows the RMI values corresponding to the AMI performance. For these results, it is evident that neither of the methods investigated performs uniformly best across the range of LFR benchmarks parameters investigated. instead, the method that has the best relative performance depends on the values of the parameters γ and *Ave Degree*. Figures 2.15 and 2.16 show linear interpolations of the RMIs given in Table 2.1 across the variable γ for the three values of *Ave Degree* considered in our study.

	γ	β	$RMI_{proposed}$	$RMI_{Modularity}$
Ave Degree = 5	2	1	0.0206	0.0033
	2	2	0.0206	0.0023
	3	1	0.0003	0.0383
	3	2	0.0000	0.0344
Ave Degree = 7.5	2	1	0.0050	0.0052
	2	2	0.0097	0.0036
	3	1	0.0012	0.0425
	3	2	0.0019	0.0405
Ave Degree = 10	2	1	0.0164	0.0156
	2	2	0.0100	0.0131
	3	1	0.0046	0.0310
	3	2	0.0060	0.0259

Table 2.1: RMIs for AMI Performance

From Figure 2.15, when *Ave Degree* = 5, the proposed method seems to outperform the modularity method over approximately 67.5% of the range of γ. Further, the proposed method also appears to be more robust to changes (and thus, to the variation in the degree distribution) since there is less change in the RMI values across the range of γ, relative to the modularity method. Finally, note that, from Figure 2.16, as the *Ave Degree* increases, then so does the relative performance of the proposed method. In fact, when *Ave Degree* = 10, the proposed approach achieves the best relative performance over the entire range of γ values.

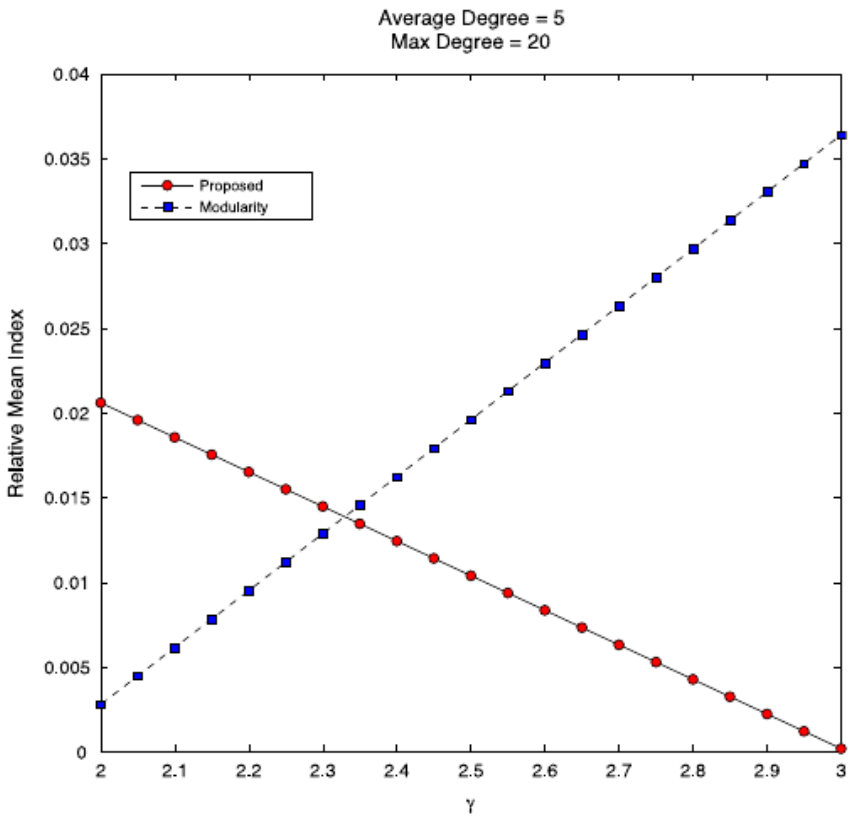


Figure 2.15: Linear interpolations of the RMIs across the parameter γ and for *Ave Degree* $= 5$. The solid line with red circles represents the proposed method, and the dotted-line with blue squares represents the modularity method.

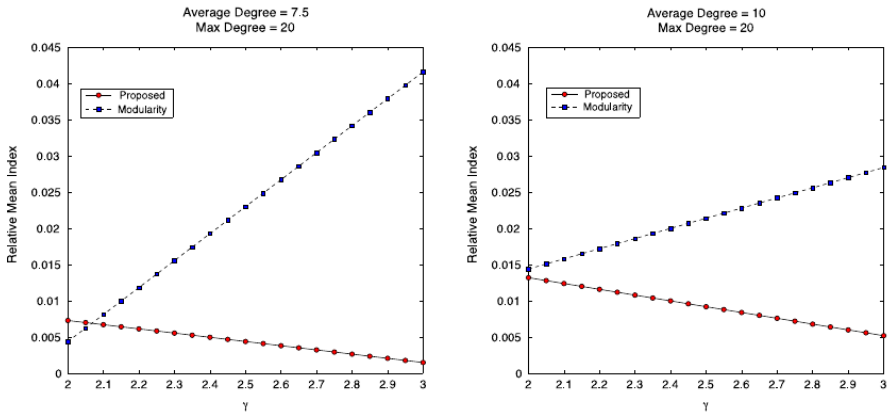


Figure 2.16: Linear interpolations of the RMIs across the parameter γ and for *Ave Degree* $= 7.5$ (left plot) and *Ave Degree* $= 10$ (right plot).

Table 2.2 shows the RMI values corresponding to the power performance. Notice that, since the RMI values corresponding to the proposed method are always zero, then relative to modularity, the proposed method achieves greater power across the entire spectrum of LFR benchmark parameter values considered

in our study. Recall that the proposed objective function is the denominator in the likelihood ratio test and specifies the likelihood function under the alternative hypothesis of k clusters. Since the group membership assignment vector that maximizes modularity is not necessarily equivalent to the one that maximizes the proposed objective function, it is not surprising that the power achieved using the modular solution is somewhat less than that achieved by using the proposed solution.

	γ	β	$RMI_{proposed}$	$RMI_{Modularity}$
$Ave\ Degree = 5$	2	1	0.0000	0.1178
	2	2	0.0000	0.1880
	3	1	0.0000	0.2161
	3	2	0.0000	0.1733
$Ave\ Degree = 7.5$	2	1	0.0000	0.1159
	2	2	0.0000	0.2023
	3	1	0.0000	0.1039
	3	2	0.0000	0.0887
$Ave\ Degree = 10$	2	1	0.0000	0.1052
	2	2	0.0000	0.0897
	3	1	0.0000	0.1286
	3	2	0.0000	0.0750

Table 2.2: RMIs for Power Performance

The results of the simulation study discussed in this section suggest that the proposed objective function is a viable alternative to modularity. In addition to better overall clustering performance, maximizing the proposed objective function will permit greater power to detect clusters, relative to computing the test statistic using the "best" modular solution. Although the proposed approach will provide greater power, our simulation results show that one can still achieve good power using the solution that maximizes the modularity metric. Since community detection via modularity maximization is widely available in several network analysis software packages, this result is particularly meaningful.

As a final note, our simulation study was conducted in the same manner as that given in Lancichinetti et al. (2008), where the assumption was made that the number of groups k is a known quantity. Obviously, this is not the case in

practice, and thus, an important future study is one that assesses relative performance between the two approaches when the number of groups k is unknown. Lastly, although our study only considered networks of size $N = 100$, there is no reason to believe that the performance observed would change dramatically for networks of larger size.

2.7 Summary and discussion

Clustering is a common first step in exploratory data analysis and is typically used to find groups of "similar" observations. The goal of clustering is to reveal underlying patterns in the data, and ultimately, to use these patterns for decision making purposes. Although the clustering of conventional data sets is very common, more recently, the clustering of network data sets has generated significant interest. In network clustering, one is often interested in finding clusters of nodes that are densely intra-connected to each other and sparsely inter-connected to external nodes. The literature contains a variety of methods available for solving this problem, where the most popular methods seek to maximize the modularity metric.

Although the modularity metric is an intuitive and effective objective function to maximize, it provides no direct statistical basis for quantifying the significance of the detected clusters. In this paper we proposed a new objective function for network clustering that extends easily to a likelihood ratio test, and thus, provides a well studied statistical basis for quantifying the significance of the detected clusters. We then derived a novel approximation to the distribution of the proposed likelihood ratio test statistic under the null hypothesis of a single cluster (i.e., $k = 1$). This distribution was then used to obtain approximate critical values for the statistical test. The application of our proposed clustering framework to the real-world networks in Zachary (1977) and Sageman (2004) was then demonstrated.

Finally, using Monte Carlo simulation, we evaluated the performance of the

proposed clustering framework, relative to the modularity method, when applied to the LFR benchmark graphs of Lancichinetti et al. (2008). Our simulations suggest that maximizing the proposed objective function will generally yield better clustering performance than that achieved by maximizing the modularity metric. Further, it was observed that if the likelihood ratio test statistic is computed using the solution obtained by maximizing modularity (as opposed to maximizing the denominator of the likelihood ratio), then the power of the test will be reduced, unless the group assignment vector that maximizes modularity also maximizes the likelihood function specified under the alternative hypothesis. Thus, for a given type I error probability α, to maintain a statistical test with the greatest power, one should make efforts to maximize the proposed objective function.

3 EFFICIENT LIKELIHOOD-BASED NETWORK CLUSTERING

3.1 Abstract

Clustering networks by maximizing likelihood produces high-quality clusters and allows for statistical significance testing but calculating parameter estimates is burdensome. We introduce iterative, efficient expressions for updating parameter estimates and compare the results between the naive and the proposed methods of calculating parameters.

3.2 Introduction

A network $G = (V, E)$ is a set of vertices V connected by a set of edges E. Clustering G involves assigning each of the network's N vertices to one, and only one, of k clusters. Once a network has been clustered, a metric can be calculated to evaluate the goodness of the clustering.

In a previous work by Perry et al. (2013), the statistical likelihood of a clustering was proposed and evaluated as an objective function to be optimized against modularity, the current gold standard of clustering metrics introduced by Newman (2006) . Perry et al. demonstrated that maximizing likelihood produces clusters of quality often exceeding those produced by maximizing modularity on benchmark networks created to simulate a range of real-world network types, and the likelihood objective function the additional benefit of a statistical significance test.

However, the calculation of likelihood and modularity for a given clustering are both computationally expensive, requiring the traversal of a network's entire

edge set E, at a minimum. Blondel et al. (2008) introduced a simple expression that required relatively little information to calculate the change-in-modularity when a vertex is reassigned from one cluster to another compared to the naive calculation of change-in-modularity in which two passes of E are necessary to calculate the difference in the pre- and post-reclustering modularities. They employed this expression in the so-called "Louvain Method", a greedy agglomerative algorithm that starts with each of N vertices assigned to its own cluster and then repeatedly groups and consolidates the vertices into an increasingly smaller number of clusters based on the observed change-in-modularity.

The Louvain Method does not allow a user-specified k, but rather continues consolidating vertices until no further increase in modularity is observed. The previously-proposed likelihood maximization method however, relies on heuristic optimization algorithms to find a near-optimal solution and requires a user-specified k at the outset, thus the methods and their solution times are not directly comparable. However, the ability of the Louvain Method to rapidly cluster networks is largely due to the efficient change-in-modularity expression and is the motivation for the work in this paper.

In this paper, we first briefly describe the likelihood maximization clustering process developed by Perry et al. Then we derive theorems that provide iterative, efficient methods for calculating the parameter estimates required for the loglikelihood, a metric equivalent to likelihood in optimization. Finally, we derive a change-in-loglikelihood formula that gains further efficiency given some reasonable parameter assumptions and then compare the time-to-solution required to cluster benchmark networks under the previous naive method and the proposed iterative method of calculating loglikelihood.

3.3 Likelihood-Based Clustering Review

Consider a simple undirected network $G = (V, E)$ with vertex set V and edge set E. Let $e_{i,i'} \in E$ be the edge weight between vertices i and $i' \in V$, and let $\mathbf{A}$ define the N x N adjacency matrix for network G such that $a_{i,i'} = e_{i,i'}$ if $e_{i,i'}$ exists, 0 otherwise. Since G is simple and undirected, $\mathbf{A}$ is symmetric with no self-edges, i.e. $a_{i,i'} = a_{i',i}$ and $a_{i,i} = 0$. This will be the assumption for all networks G going forward in Chapter 3.

Then, cluster the network G by assigning each of the N vertices to one, and only one, of k clusters. Let $\mathbf{z}_{(k)}$ be a N x 1 cluster membership vector and element $z_{i,(k)}$ denote the cluster assignment for vertex i, and $z_{i,(k)} \in \{1, 2, ...k\}$, $i \in (1, 2, ...N)$. Let $\theta = [\theta_1, ..., \theta_k, \theta_b]$ denote a $k + 1$ unknown parameter vector, with elements θ_h, $h \in \{1, 2, ...k\}$, denoting the expected edge value between any two vertices belonging to cluster h and θ_b denoting the expected edge value between any two vertices not belonging to the same cluster. Finally, let $\mathbf{Y}|\mathbf{z}_{(k)} = [Y_1, ..., Y_k, Y_b]$ be a vector of sufficient statistics for the parameter vector θ, conditioned on cluster assignment vector $\mathbf{z}_{(k)}$, and let $\mathbf{y}|\mathbf{z}_{(k)}$ denote a realization of $\mathbf{Y}|\mathbf{z}_{(k)}$. Given cluster membership assignment $\mathbf{z}_{(k)}$ and assuming that the number of vertices N is fixed, there is no information available on the vertices other than an arbitrary label and the edges between vertices are conditionally independent, the likelihood function for θ, given $\mathbf{z}_{(k)}$ and $\mathbf{y_b}|\mathbf{z}_{(k)}$, can then be written as

$$L(\theta|\mathbf{y}, \mathbf{z}_{(k)}) = \{\prod_{h=1}^{k} f_h(y_h|\theta_h, \mathbf{z}_{(k)})\} f_b(y_b|\theta_b, \mathbf{z}_{(k)}), \tag{3.1}$$

where $f_h(y_h|\theta_h, \mathbf{z}_{(k)})$ and $f_b(y_b|\theta_b, \mathbf{z}_{(k)})$ indicate the respective probability density (or mass) functions for the sufficient statistics Y_h and Y_b. The optimal clustering is then found by solving:

$$\mathbf{z}_{(k)} = \arg \max_{\mathbf{z}_{(k)} \in \mathbf{Z}_{(k)}} [\prod_{h=1}^{k} f_h(\hat{\theta}_h, \mathbf{z}_{(k)}|y_h) f_b(\hat{\theta}_b, \mathbf{z}_{(k)}|y_b)], \tag{3.2}$$

where $\hat{\theta}_h$ and $\hat{\theta}_b$ are the respective maximum likelihood estimators of θ_h and θ_h, and $\mathbf{Z}_{(k)}$ denotes the set of all possible partitions of N vertices into k clusters. This is a combinatorial optimization problem for which the number of possible clusterings is generally too great to evaluate every potential solution, even for small networks. There are, for example, 6.57 x 10^{67} ways to cluster a network of $N = 100$ vertices into $k = 5$ clusters. Thus a heuristic optimization algorithm such as simulated annealing must be used to search for a near-optimal clustering. Additionally, the loglikelihood is maximized in practice rather than the likelihood. This has no effect on the solution but is convenient to work with.

Readers are referred to the aforementioned Perry et al. (2013) paper for detailed information on clustering undirected networks by likelihood maximization, including comparison to alternate methods, application to real-world networks and the introduction of a novel statistical significance test for the solution clustering.

3.4 Theorems

Proofs for Theorems 3.4.1 $\sim$ 3.4.5 are found in the Appendix.

Theorem 3.4.1 demonstrates how the change-in-loglikelihood due to a reclustering can be greatly simplified given some simple parameter constraints.

Theorem 3.4.1. *Assign each vertex $v \in V$ from $G = (V, E)$ into one of k clusters. Assume the edges within cluster $h \in \{1, 2..., k\}$ are distributed according to density $f_h(y_h|\theta_h, \mathbf{z}_{(k)})$ and the edges between clusters are distributed according to density $f_b(y_b|\theta_b, \mathbf{z}_{(k)})$. The loglikelihood of the clustering $\mathbf{z}_{(k)}$ is then written as*

$$\ln L(\theta_1, \theta_2, ..., \theta_k, \theta_b|\mathbf{y}, \mathbf{z}_{(k)}) = ln[\{\prod_{h=1}^{k} f_h(y_h|\theta_h, \mathbf{z}_{(k)})\} f_b(y_b|\theta_b, \mathbf{z}_{(k)})], \tag{3.3}$$

where each θ_h and θ_b are estimated by maximum likelihood estimators $\hat{\theta}_h$ and $\hat{\theta}_b$, respectively.

Then, consider a reclustering by reassigning vertex l^* from cluster ℓ to

cluster j. If the parameter estimates for the within-cluster edge densities other than f_ℓ and f_j are unaffected by the reclustering, i.e. $\hat{\theta}_h^1 = \hat{\theta}_h^0$, $h \notin \{\ell, j\}$, then the change-in-loglikelihood due to the reclustering can be written:

$$\Delta \ln L = ln \left[\frac{f_\ell(y_\ell|\hat{\theta}_\ell^1, \mathbf{z}_{(k)}^1) f_j(y_j|\hat{\theta}_j^1, \mathbf{z}_{(k)}^1) f_b(y_b|\hat{\theta}_b^1, \mathbf{z}_{(k)}^1)}{f_\ell(y_\ell|\hat{\theta}_\ell^0, \mathbf{z}_{(k)}^0) f_j(y_j|\hat{\theta}_j^0, \mathbf{z}_{(k)}^0) f_b(y_b|\hat{\theta}_b^0, \mathbf{z}_{(k)}^0)} \right], \tag{3.4}$$

where superscripts 0 and 1 indicate pre- and post-reclustering states, respectively.

Theorems 3.4.2 and 3.4.3 demonstrate how post-reclustering edge weight sums can be calculated from pre-reclustering information.

Let $\boldsymbol{\omega}_j$ be a N x 1 cluster membership vector for cluster j such that element $\omega_{i,j} = 1$ if vertex i belongs to cluster j, 0 otherwise. Let $\boldsymbol{\omega}$ be the N x k cluster membership matrix with ij^{th} element $= \omega_{i,j}$. Let $\mathbf{1_s}$ be a N x 1 vector of 0's and a single 1 in the sth position. If vertex l^* is removed from cluster ℓ and placed in cluster j, then:

$$\boldsymbol{\omega}_\ell^1 = \boldsymbol{\omega}_\ell^0 - \mathbf{1}_{l^*} \text{ and } \boldsymbol{\omega}_j^1 = \boldsymbol{\omega}_j^0 + \mathbf{1}_{l^*}.$$

Given clustering $\mathbf{z}_{(k)}$ on an undirected network, $\mathbf{OBS} = \boldsymbol{\omega}^T \mathbf{A} \boldsymbol{\omega}$ is a symmetric k x k matrix with element $obs_{s,t} = \boldsymbol{\omega}_s^T \mathbf{A} \boldsymbol{\omega}_t$, the sum of the observed edge weights between clusters s and t. However, since $\mathbf{A}$ is a symmetric matrix, extracting the rows and columns of $\mathbf{A}$ belonging to vertices of cluster s also forms a symmetric sub-matrix. Therefore $\boldsymbol{\omega}_s^T \mathbf{A} \boldsymbol{\omega}_s$ double counts the edge weights between vertices within the same cluster s (See Newman (2010), pg. 112). Thus, the correct **OBS** matrix is obtained by:

$$obs_{s,t} = \begin{cases} \boldsymbol{\omega}_s^T \mathbf{A} \boldsymbol{\omega}_t, & \text{if s} \neq \text{t .} \\ (\frac{1}{2}) \boldsymbol{\omega}_s^T \mathbf{A} \boldsymbol{\omega}_t, & \text{if s} = \text{t.} \end{cases}$$

Theorem 3.4.2. *Assign each vertex $v \in V$ from $G = (V, E)$ into one of k clusters. Then, consider a reclustering by reassigning vertex l^* from cluster ℓ to cluster j. The elements of* **OBS** *post-reclustering can be written completely in terms of pre-reclustering information. Specifically, the elements are:*

1. $obs^1_{s,t} = obs^0_{s,t}$

2. $obs^1_{s,s} = obs^0_{s,s}$

3. $obs^1_{\ell,s} = obs^0_{\ell,s} - Z^0_s$

4. $obs^1_{j,s} = obs^0_{j,s} + Z^0_s$

5. $obs^1_{\ell,j} = obs^0_{\ell,j} - Z^0_j + Z^0_\ell$

6. $obs^1_{\ell,\ell} = obs^0_{\ell,\ell} - Z^0_\ell$

7. $obs^1_{j,j} = obs^0_{j,j} + Z^0_j$,

where $s, t \notin \{\ell, j\}$, superscripts 0 and 1 indicate pre- and post-reclustering states, respectively, and Z^0_h is the sum of edge weights between vertex l^* and cluster h pre-reclustering, $h \in \{1, 2, ..., k\}$.

Theorem 3.4.3. *Assign each vertex $v \in V$ from $G = (V, E)$ into one of k clusters. Then, consider a reclustering by reassigning vertex l^* from cluster ℓ to cluster j. The sum of the distinct off-diagonal elements of* **OBS** *post-reclustering can be written completely in terms of pre-reclustering information. Specifically:*

$$obs^1_b = \sum obs^1_{s,t} = obs^0_b + Z^0_\ell - Z^0_j, \tag{3.5}$$

where $s, t \in \{1, 2, ...k\}$ and $s \neq t$, superscripts 0 and 1 indicate pre- and post-reclustering states, respectively, and Z^0_h is the sum of edge weights between vertex l^* and cluster h pre-reclustering, $h \in \{1, 2, ..., k\}$.

Theorems 3.4.4 and 3.4.5 demonstrate how the maximum possible number of edges within and between clusters post-reclustering can be calculated from pre-reclustering information.

Given clustering $\mathbf{z}_{(k)}$ on an undirected network, **POS** is a symmetric k x k matrix with element $pos_{s,t}$ containing the maximum number of distinct edges possible between the vertices of clusters s and t. This value is $pos_{s,t} = \binom{n_s+n_t}{2} - \binom{n_s}{2} - \binom{n_t}{2} = n_s n_t$ when $s \neq t$ and $pos_{s,t} = \frac{n_s(n_s-1)}{2}$ when $s = t$, where n_h is the number of distinct vertices in cluster h $\in \{1, 2, ..., k\}$.

Theorem 3.4.4. *Assign each vertex $v \in V$ from $G = (V, E)$ into one of k clusters. Then, consider a reclustering by reassigning vertex l^* from cluster ℓ to cluster j. The elements of* **POS** *post-reclustering can be written completely in terms of pre-reclustering information. Specifically, the elements are:*

1. $pos^1_{s,t} = pos^0_{s,t}$

2. $pos^1_{s,s} = pos^0_{s,s}$

3. $pos^1_{\ell,s} = n^0_s(n^0_\ell - 1)$

4. $pos^1_{j,s} = n^0_s(n^0_j + 1)$

5. $pos^1_{\ell,j} = (n^0_\ell - 1)(n^0_j + 1)$

6. $pos^1_{\ell,\ell} = \frac{(n^0_\ell - 1)(n^0_\ell - 2)}{2}$

7. $pos^1_{j,j} = \frac{(n^0_j + 1)n^0_j}{2}$,

where $s, t \notin \{\ell, j\}$, superscripts 0 and 1 indicate pre- and post-reclustering states, respectively, and n^0_t is the number of vertices in cluster t pre-reclustering.

Theorem 3.4.5. *Assign each vertex $v \in V$ from $G = (V, E)$ into one of k clusters. Then, consider a reclustering by reassigning vertex l^* from cluster ℓ to cluster j. The sum of the distinct off-diagonal elements of* **POS** *post-reclustering can be written completely in terms of pre-reclustering information. Specifically:*

$$pos^1_b = \sum pos^1_{s,t} = pos^0_b + n^0_\ell - n^0_j - 1, \tag{3.6}$$

where $s, t \in \{1, 2, ...k\}$ and $s \neq t$, superscripts 0 and 1 indicate pre- and post-reclustering states, respectively, and n^0_h is the number of vertices in cluster h pre-reclustering, $h \in \{1, 2, ..., k\}$.

3.5 Improvements to Loglikelihood-Based Clustering

3.5.1 Efficient calculation of change-in-loglikelihood

Theorem 3.4.1 states that the expression for the change in observed loglikelihood due to the reclustering of G by moving vertex l^* from cluster ℓ to cluster j can be greatly simplified if parameter estimates $\hat{\theta}_h$ for within-cluster edge densities other than f_ℓ and f_j remain unchanged, i.e. $\hat{\theta}^1_h = \hat{\theta}^0_h$, $h \notin \{\ell, j\}$. Rather than calculate the full expression for the change in loglikelihood,

$$\Delta \ln L(\hat{\theta}|\mathbf{y}, \mathbf{z}_{(k)}) = ln \left[\frac{\prod_{s \neq \ell, j} f_s(y_s|\hat{\theta}^1_s, \mathbf{z}^1_{(k)})}{\prod_{s \neq \ell, j} f_s(y_s|\hat{\theta}^0_s, \mathbf{z}^0_{(k)})} \frac{f_\ell(y_\ell|\hat{\theta}^1_\ell, \mathbf{z}^1_{(k)}) f_j(y_j|\hat{\theta}^1_j, \mathbf{z}^1_{(k)})}{f_\ell(y_\ell|\hat{\theta}^0_\ell, \mathbf{z}^0_{(k)}) f_j(y_j|\hat{\theta}^0_j, \mathbf{z}^0_{(k)})} \frac{f_b(y_b|\hat{\theta}^1_b, \mathbf{z}^1_{(k)})}{f_b(y_b|\hat{\theta}^0_b, \mathbf{z}^0_{(k)})} \right], \tag{3.7}$$

we can use the simplified expression (3.4) from Theorem 3.4.1 which only requires the two within-cluster densities f_ℓ and f_j and the single between-cluster density f_b. The computational savings achieved by eliminating the need to calculate the remaining (k - 2) within-cluster parameters, and the corresponding densities, increases as k grows large.

If the parameter estimates for densities other than f_ℓ and f_j can be calculated from suitable **OBS** and **POS** matrix entries, then Theorems 3.4.2 and 3.4.4 show that the parameter estimates will not change as a result of the reclustering. Thus, given this important assumption about the parameters, we can use Theorem 3.4.1.

This is clearly a useful result since the maximum likelihood parameter estimates for binomial and Poisson densities, crucial to the analysis of discretely-distributed networks, can be calculated directly from **OBS** and **POS**, namely $\hat{\theta}_s = \frac{obs_{s,s}}{pos_{s,s}}, s \in \{1, 2, ..., k\}$ and $\hat{\theta}_b = \frac{\sum obs_{s,t}}{\sum pos_{s,t}}, s, t \in \{1, 2, ..., k\},\ s \neq t$.

3.5.2 Easily calculated parameter components

Theorems 3.4.2 $\sim$ 3.4.5 provide iterative formulas for the efficient calculation of the parameter estimates required for Theorem 3.4.1, provided that those estimates can be calculated from **OBS** and **POS** matrices. To calculate the parameter estimates without the use of these formulas while storing the network as an adjacency matrix, all $\binom{N}{2}$ distinct elements must be read in for every proposed reclustering. Considering the sparsity of modern networks, most of these elements will be zero and only slow the calculation down. If a more efficient storage format containing only non-zero edge information were used, such as an edge list or an adjacency list, then the full set of E edges still must be read to calculate the parameter estimates for every proposed reclustering.

Theorems 3.4.2 $\sim$ 3.4.5, however, show how to calculate parameter estimates for a proposed reclustering almost immediately from the parameter estimates from the current clustering. The only "new" information necessary to

update the **OBS** matrix elements required for Theorem 3.4.1 is Z_ℓ^0 and Z_j^0, the total edge weights between vertex l^* and the clusters ℓ and j, respectively, before reclustering. This information can be obtained by referencing the pre-reclustering clustering assignment vector $\mathbf{z}_{(k)}$ against 1) l^*'s row in the adjacency matrix, 2) entries containing l^* in the edge list, or 3) l^*'s row in the adjacency list. For non-binary edge weights, 2) and 3) assume that the edge and adjacency lists also contain weight information. While 1) is an improvement over reading in the entire $\binom{N}{2}$ distinct adjacency matrix elements, requiring only N elements to be read, many of those elements will still likely be zero, and thus unhelpful. Options 2) and 3) are more efficient as the edge and adjacency lists contain only non-zero edge values. To calculate Z_ℓ^0 and Z_j^0 we need only to sum the edge weights of the vertices actually connected to vertex l^* by cluster, the number of which are equal to the degree of l^*, d_{l^*}. For any proposed reclustering, the expected number of elements necessary for calculating the required elements of the **OBS** matrix is $\mathrm{E}[d_{l^*}]$ = average degree of $G \ll N \ll \binom{N}{2}$.

Calculating the required elements of the **POS** matrix doesn't require any new information, simply a running count of vertices per cluster at the current clustering, $\mathbf{z}_{(k)}$. Cluster sizes are calculated when the inital assignment is made and maintained throughout the clustering process, requiring only a subtraction and addition of 1 to the counts for clusters ℓ and j, respectively, when a a proposed reclustering is accepted in the optimization process.

3.5.3 Reduced updating requirements

In the authors' prior implementation of likelihood-based clustering, the $2(k + \binom{k}{2})$ distinct elements of the post-reclustering $\mathbf{OBS}^1$ and $\mathbf{POS}^1$ matrices were calculated for each proposed reclustering. If the reclustering was accepted, then the $\mathbf{OBS}^1$ and $\mathbf{POS}^1$ matrices were reassigned to pre-reclustering $\mathbf{OBS}^0$ and $\mathbf{POS}^0$ before considering the next proposed reclustering. The $2\binom{k}{2}$ distinct off-diagonal $obs_{s,t}^1$ and $pos_{s,t}^1$ elements needed to be maintained in order to

calculate obs_b^1 and pos_b^1, elements required to calculate the between-cluster density parameter estimate $\hat{\theta}_b^1$.

Theorems 3.4.3 and 3.4.5, however, provide a means of calculating obs_b^1 and pos_b^1 directly from obs_b^0 and pos_b^0. It's now only necessary to maintain the $2(k+1)$ elements consisting of obs_h^0, pos_h^0, obs_b^0 and pos_b^0, h $\in \{1, 2, ..., k\}$ from iteration to iteration and if a reclustering is accepted, only obs_i^1 and $pos_i^1, i \in \{\ell, j, b\}$ need be updated. This efficiency increases as k grows large since it is no longer necessary to store or update the remaining $2(\binom{k}{2} - 1)$ elements.

To demonstrate the improved algorithm, LFR benchmark networks (Lancichinetti et al. (2008)), of size $N = 250$, 500, 750, 1000, 1250, 1500, 1750 and 2000 were generated with parameters selected to emulate those used in Figure 2.9. Each network was then clustered into 10, 20, 30, 40 and 50 clusters using both the old naive and newly-proposed methods of calculating the loglikelihood parameters and the change-in-loglikelihood during the optimization process. Both methods used simulated annealing with a common cooling schedule[1] to find the solution clusters. Additionally, the Mersenne Twister proposed by Matsumoto and Nishimura (1998) was used to generate identical random number streams for both methods, thus controlling for clustering quality. Although we have chosen simulated annealing as an optimization algorithm in this demonstration, the results apply to other heuristics as well, provided that they can use change-in-loglikelihood as the accept/reject criteria.

Figure 3.1 demonstrates the improvement in time-to-solution between the two methods of calculating parameter estimates and the change-in-loglikelihood. At $N = 2000$, the old method requires $19 \sim 20$ minutes while the new method reaches a solution in under 4.5 seconds, regardless of k.

[1]Initial Temp: 1, Cooling Rate: 0.99, Temp Length: 300

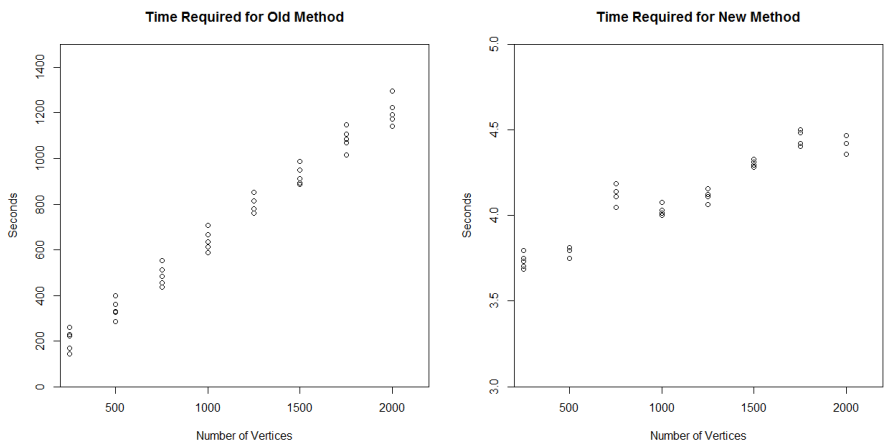


Figure 3.1: Time comparison of clustering using naive and proposed methods of MLE parameter calculation, by Vertex Count

A further test was done with a LFR network of size $N = 100,000$ generated under identical conditions. The network was clustered into $k = 500$ clusters. The old method ran for more than 24 hours, and was finally terminated for time consideration, while the newly-proposed method arrived at a solution in 39 seconds.

3.6 Summary

The original paper on likelihood-based network clustering demonstrated the utility of maximized likelihood as a clustering metric. However, the need to read in the entire edge list E to calculate the loglikelihood for each of the large number of proposed solutions considered during optimization was a bottleneck to further use and development. In this paper we have presented iterative, efficient expressions that drastically reduce the effort required to calculate the change-in-loglikelihood when considering a proposed reclustering, and thus the overall time-to-solution, for a class of statistical distributions important to network analysis, and we have demonstrated the improved time-to-solution on several benchmark networks.

4 ON THE STATISTICAL DETECTION OF CLUSTERS IN DIRECTED NETWORKS

4.1 Abstract

The goal of network clustering algorithms is to assign each node in a network to one of several mutually exclusive groups based upon the observed edge set. Perry et al. (2013) proposed a likelihood metric for the undirected network case and demonstrated its competitiveness against modularity, in addition to introducing a novel statistical significance test. In this paper, we address clustering of directed networks, i.e. when the edges connecting vertices do so in a specific direction by proposing a directed version of the likelihood clustering metric. Like the undirected metric, the directed likelihood metric is maximized over the space of possible group membership assignments and is thus a candidate for the use of information criterion methods such as Akaike or Bayesian to determine the "best" number of clusters. Further, the aforementioned statistical significance test is applicable in the directed case as well, without modification. Using Monte Carlo simulation, we compare the performance of the proposed likelihood objective function for network clustering against that of directed modularity using LFR benchmark networks, and demonstrate the use of the proposed objective function on real world networks.

4.2 Introduction

Networks are ubiquitous in the modern world. The Internet, and every device connected to it, is part of a network and has been a target of study for

many years (Broder et al. (2000), Barabasi (2001), Zhang et al. (2005)).

In addition to communications networks, the associations between people is a popular area of study. In previous years, information might be gathered on friendships among people in a certain group and analyzed as a social network [Hansell (1984)). With the advent of the Internet and the explosion in popularity of social networking sites such as Facebook and Linkedin, social networks are available on a scale nearly as large as the internet itself. Motivations for the study of networks are as broad as the types of networks, from improved marketing efforts based on an understand of connected peer groups to increasing the understanding of structure within terrorist organizations (Basu (2005)).

Networks can be broadly classified into two categories: undirected and directed. Undirected networks can be further characterized by the edges that connect the vertices. In the unweighted case, the edges are binary and either exist or they do not. In the weighted case, edges that exist also have an associated weight attribute. Frequently these weights are non-negative integers representing counts but they can in theory take any value. Directed networks share these same qualities with the exception that edges have an additional attribute: direction. In the undirected case, vertices a and b may be connected by and edge e_{ab} with weight w but the connection exists equally to and from a and b. In the directed case, an edge e_{ab} may exist from a to b with weight w_{ab} but may not exist from b to a at all. Or, e_{ba} may exist but with a different weight $w_{ba} \neq w_{ab}$.

The clustering of undirected networks has been a field of study for many years and many algorithms have been developed in that time. Readers are directed to Fortunato (2010) for a comprehensive review of these algorithms. Clustering algorithms for directed networks have also been developed, and these are generally modifications of directed network clustering algorithms. Examples include directed modularity (Leicht and Newman (2008)), directed spectral clustering [Gleich (2006)], directed random walk methods (Huang and Zhu (2006)) and directed stochastic block models (Rohe et al. (2006)).

To define undirected modularity, consider a network of size N, and let $\boldsymbol{\omega}_{ij} = 1$ if node i belongs to group j, and 0 otherwise ($i = 1, ..., N$ and $j = 1, ..., k$). Further, let $A_{ii'}$ denote the ii'th element of the adjacency matrix $\mathbf{A}$, m denote the total number of edges in the network, and d_i the degree of node i. For a given N x k group membership matrix and N x N adjacency matrix $\mathbf{A}$, the modularity is defined as

$$Q(\boldsymbol{\omega}|\mathbf{A}) = \frac{1}{2m} tr(\boldsymbol{\omega}^T \boldsymbol{B} \boldsymbol{\omega}) \tag{4.1}$$

where tr($\mathbf{G}$) denotes the trace of the matrix $\mathbf{G}$ and

$$B_{ii'} = A_{ii'} - \frac{d_i d_{i'}}{2m} \tag{4.2}$$

denotes the elements of the so called modularity matrix.

Modularity measures the fraction of edges that fall within the given groups minus the expected such fraction if edges were distributed at random. Large modularity values indicate the presence of densely intra-connected and sparsely inter-connected vertices.

In the directed case introduced by Leicht and Newman (2008), modularity is defined by first creating symmetry in the modularity matrix ($\boldsymbol{B} + \boldsymbol{B}^T$) and recognizing that the number of possible edges in the network has now doubled to $4m$

$$Q(\boldsymbol{\omega}|\mathbf{A}) = \frac{1}{4m} tr(\boldsymbol{\omega}^T (\boldsymbol{B} + \boldsymbol{B}^T) \boldsymbol{\omega}) \tag{4.3}$$

where the definitions of $\boldsymbol{\omega}$ and $\boldsymbol{B}$ remain as above.

In both the directed and undirected case, the clustering problem involves finding $\boldsymbol{\omega}^*$, i.e., the group membership matrix in the set of all group membership matrices $\boldsymbol{\Omega}_k$ that yields the maximum modularity value, or

$$\boldsymbol{\omega}^* = \arg\max_{\boldsymbol{\omega} \in \boldsymbol{\Omega}_k} [Q(\boldsymbol{\omega}|\mathbf{A})]. \tag{4.4}$$

Whether directed or undirected, the optimization problem above is extremely difficult due to the combinatorial explosion in the number of possible solutions requiring evaluation. Rather than an exhaustive search, researchers often employ heuristic algorithms to search a subset of $\mathbf{\Omega}_k$ for a "good", albeit potentially sub-optimal, solution. Heuristic algorithms include stochastic search methods such as simulated annealing or genetic algorithms (Guimera et al. (2004); Ku cukpetek et al. (2005)). These methods are generally found to be slower but more accurate than other deterministic methods. Danon et al. (2005), for example, found simulated annealing to produce the most accurate results of all algorithsm tested. However, heuristic algorithms are no limited to the use of the modularity metric and there are several other non-heuristic algorithms available as well. Readers are referred to Malliaros and Vazirgiannis (2013) for a comprehensive review of clustering algorithms available for detecting communities in directed networks.

Regardless of method, all clustering algorithms seek to classify the vertices of a network into some number of clusters. Some algorithms require the number of clusters to be specified a priori as an input parameter to the algorithm (Leicht and Newman (2008), Newman and Leicht (2007)) while some algorithms have an internal stopping criterion built-in (Dugue and Perez (2015)). While any clustering algorithm may cluster a network, an important question is whether that clustering is significant or simply the result of randomness. Bianconi et al. (2009) define a measure, Θ, based on entropy measures that seeks to quantify the relevance of some detected community structure; however, no distribution for this measure is given. Lancichinetti et al. (2010) describe a procedure based on extreme and order statistics than can be used to determine the significance of clusters in unweighted, undirected networks. Lancichinetti and Fortunato (2009) extended this method to account for more general network structures, including weighted and/or directed networks. Zhao et al. (2011) propose a formal statistical test that uses simulation and permutations to approximate the

distribution of the test statistic under the null hypothesis, and thus, facilitate testing. Perry et al. (2013) proposed a formal statistical test for the significance of a clustering based on maximum likelihood methods and order statistics.

In this paper, we propose a new objective function for maximization on directed networks (weighted or unweighted) based on the undirected objective function of Perry et al. (2013). Since the proposed objective function is a likelihood function, it lends nicely to the use of information criterion proposed by Akaike (1974) or Schwarz (1978) for determining the "best" number of groups or clusters. Further, the statistical significance test developed and demonstrated on undirected networks by Perry et al. (2013) can be used as-is on directed networks. We will evaluate the clustering performance obtained by maximizing the proposed objective function, relative to that achieve by maximizing directed modularity. We consider the benchmark directed networks developed by Lancichinetti and Fortunato (2009) (i.e. directed LFR benchmark networks) in our evaluation of the objective functions since these graphs possess some properties of real-world networks.

4.3 A new objective function for directed network clustering

In the following sections, we review the likelihood objective function for the undirected network case and present its directed network counterpart.

4.3.1 Undirected case

In Section 2.3, a model was developed to describe the likelihood of an undirected network clustering $\boldsymbol{z}_{(k)}$ given that the edges falling within cluster h are distributed according to density $f_h(y_h|\theta_h, \mathbf{z}_{(k)})$ and the edges falling between clusters are distributed according to density $f_b(y_b|\theta_b, \mathbf{z}_{(k)})$ can be written as

$$L(\boldsymbol{\theta}|\mathbf{y}, \mathbf{z}_{(k)}) = \{\prod_{h=1}^{k} f_h(y_h|\theta_h, \mathbf{z}_{(k)})\} f_b(y_b|\theta_b, \mathbf{z}_{(k)}), \tag{4.5}$$

where k is the number of clusters and $\boldsymbol{\theta} = [\theta_1, ..., \theta_k, \theta_b]$ denotes a k + 1 unknown parameter vector.

The solution clustering is

$$\mathbf{z}^*_{(k)} = \arg \max_{\mathbf{z}_{(k)} \in \mathbf{Z}_{(k)}} [\max_{\boldsymbol{\theta}(\mathbf{z}_{(k)}) \in \boldsymbol{\Theta}_k(\mathbf{Z}_{(k)})} (\prod_{h=1}^{k} f_h(\mathbf{z}_{(k)}, \theta_h | y_h) f_b(\mathbf{z}_{(k)}, \theta_b | y_b))]. \tag{4.6}$$

where y_h and y_b are sufficient statistics calculated by summing the observed edge weights within cluster $h(h = 1, ..., k)$ and summing edge weights between clusters, respectively.

In the specific case of binomial densities, the solution clustering is

$$\mathbf{z}^*_{(k)} = \arg \max_{\mathbf{z}_{(k)} \in \mathbf{Z}_{(k)}} [\prod_{h=1}^{k} \theta_h^{y_h} (1 - \theta_h)^{N_h - y_h} \theta_b (1 - \theta_b)^{N_b - y_b}]. \tag{4.7}$$

where $\hat{\theta}_h = y_h / N_h, (h = 1, ..., k)$ and $\hat{\theta}_b = y_b / N_b$ are the maximum likelihood estimators of the unknown parameters for a given $\mathbf{z}_{(k)}$ and N_h and N_b are the number of potential edges within cluster $h(h = 1, ..., k)$ and between clusters, respectively.

4.3.2 Directed case

The directed form of likelihood takes on the same form of Equation 4.5, with edges falling within cluster h distributed according to density $f_h(y_h | \theta_h, \mathbf{z}_{(k)})$ and edges falling between clusters distributed according to density $f_b(y_b | \theta_b, \mathbf{z}_{(k)})$. Furthermore, the solution is the clustering $\mathbf{z}_{(k)}$ that maximizes Equation 4.7.

Although their forms are identical, the directed and undirected versions of Equation 4.5 differ in the way the sufficient statistics, y_h and y_b, and possible edge counts, N_h and N_b, are derived in order to calculate maximum likelihood estimators for each density. In the undirected case, it was sufficient to know if an edge exists between two vertices and its weight, if any. In the directed case, we must take into account that edges can now exist in one or both directions between two vertices and be aware that each of these edges may have a different

	1	2	3	4	5
1	0	1	1	1	1
2	1	0	1	1	1
3	1	1	0	1	0
4	1	1	1	0	1
5	1	1	0	1	0

(a) Cluster 1 edges

	6	7	8	9
1	1	1	0	1
2	1	0	0	1
3	0	0	1	0
4	0	0	1	0
5	1	0	1	1

(b) Clusters 1 and 2 edges

	1	2	3	4	5
6	1	1	0	0	1
7	1	0	0	0	0
8	0	0	1	1	1
9	1	1	0	0	1

(c) Clusters 2 and 1 edges

Figure 4.1: Edges within cluster 1 {vertices 1,2,3,4,5}, and between cluster 1 and cluster 2 {vertices 6,7,8,9} in a simple, undirected binary network.

weight. It is also possible for two vertices to have multiple edges extending in one or both directions, but such a situation is beyond the scope of this manuscript. Therefore, in the directed case, y_h is the total weight of the edges extending from vertex v_{h_i} to vertex $v_{h_{i'}}$ and the edges from $v_{h_{i'}}$ to v_{h_i}, where both $v_{h_{i'}}$ and v_{h_i} belong to cluster h and $i \neq i'$. Further, y_b is the total weight of edges extending from vertex v_{h_i} to vertex $v_{j_{i'}}$ and the edges extending from $v_{j_{i'}}$ to v_{h_i}, where v_{h_i} belongs to cluster h, $v_{j_{i'}}$ belongs to cluster j, $h, j \in \{1, 2, ..., k\}$, and $h \neq j$.

The possible number of edges between vertices of cluster h in the undirected case was $N_h = \binom{n_h}{2} = \frac{n_h(n_h-1)}{2}$ and between clusters was $N_b = \sum \binom{n_h+n_j}{2} - N_h - N_j = n_h n_j$, $h, j \in \{1, 2, ..., k\}$,$h \neq j$. In the directed case, it is possible that edges can extend in either direction between any two vertices, thus these counts are double those of the undirected case, i.e in the directed case $N_h = n_h(n_h - 1)$ and $N_b = 2n_h n_j$.

The difference is illustrated in Figures 4.1 and 4.2. Figure 4.1 shows partial edge information for an undirected network, assuming $k = 2$ and clustering $\mathbf{z}_{(2)}$. Since the edges are binary, a binomial distribution is appropriate. Cluster 1 contains vertices $\{1, 2, 3, 4, 5\}$ and cluster 2 contains vertices $\{6, 7, 8, 9\}$, thus $n_1 = 5$ and $n_2 = 4$. The edges that extend from the vertices of cluster 1 to other

vertices of cluster 1 are shown in matrix format in 4.1a. The symmetry of the matrix is reflective of the undirected nature of the network. The single edge connecting vertices 1 and 5, shown in [row 1, column 5] is the same edge as the edge connecting vertices 5 and 1, shown in [row 5, column 1]. Since we are only dealing with simple networks in this manuscript, a vertex may not have an edge that connects to itself and thus the diagonals are shaded to illustrate this point. Further, since the edge set represented in the lower triangle is the same as that of the upper triangle, we shade the lower triangle of the adjacency matrix and leave the upper triangle unshaded. This upper triangle then represents the unique edges within cluster 1 and summing the binary edge weights yields $y_1 = 8$. Further, $N_1 = \binom{5}{2} = 10$, the number of unshaded elements. The maximum likelihood parameter estimate for the cluster 1 density in Equation 4.7 given $\mathbf{z}_{(2)}$ is then $\hat{\theta}_1 = 8/10 = 0.8$.

The edges between clusters 1 and 2 can be displayed in two ways, as shown in Figures 4.1b and 4.1c. Since both matrices represent the same edge set, we shade Figure 4.1c entirely and sum only the edge weights in Figure 4.1b to get $y_b = 10$. Further, the possible edge count is $N_b = 5 * 4 = 20$. The maximum likelihood parameter estimate for the between-cluster edge density in Equation 4.7 given $\mathbf{z}_{(2)}$ is then $\hat{\theta}_b = 10/20 = 0.5$

The directed case requires more information when calculating parameter estimates. Consider the partial edge information of a directed network, again assuming $k = 2$ and clustering $\mathbf{z}_{(2)}$, shown in Figure 4.2. As the network is still simple, the diagonal element is also shaded in Figure 4.2a. However, the matrix is no longer symmetric since the edge set in the upper triangle is no longer the same edge set in the lower triangle. The edge connecting vertex 5 to vertex 1 in the lower triangle, for example, is now distinct from the edge connecting vertex 1 to vertex 5 in the upper triangle. Thus, when calculating y_1 in the directed case we sum the off-diagonal elements of both the lower and upper matrices to get $y_1 = 14$. The possible number of edges is now the count of unshaded elements in

	1	2	3	4	5
1	0	1	1	1	1
2	1	0	1	1	0
3	0	1	0	0	0
4	1	1	1	0	1
5	1	1	0	0	0

(a) Cluster 1 edges

	6	7	8	9
1	1	1	0	1
2	1	0	0	1
3	0	0	1	0
4	0	0	1	0
5	1	0	1	1

(b) Edges from cluster 1 to cluster 2

	1	2	3	4	5
6	0	0	1	1	1
7	1	1	0	1	0
8	0	0	0	1	1
9	0	1	0	0	0

(c) Edges from cluster 2 to cluster 1

Figure 4.2: Edges within cluster 1 {vertices 1,2,3,4,5}, and extending to and from clusters 1 and 2 {vertices 6,7,8,9} in a simple, directed binary network.

both the lower and upper matrices, $N_1 = 20$. The maximum likelihood parameter estimate for cluster 1 in Equation 4.7 given $\mathbf{z}_{(2)}$ is $\hat{\theta}_1 = 14/20 = 0.7$.

Unlike the undirected case, the edge set pointing from the vertices of cluster 1 to the vertices of cluster 2, Figure 4.2b, is not the same set of edges pointing from the vertices of cluster 2 to the vertices of cluster 1, Figure 4.2c. Thus we cannot ignore Figure 4.2c as before in Figure 4.1c. We sum over each edge set to get $y_{1\to 2} = 10$ and $y_{2\to 1} = 9$. $N_{1\to 2} = 5 * 4 = 20$ and $N_{2\to 1} = 4 * 5 = 20$. The maximum likelihood parameter estimate for the between-cluster edge density in Equation 4.7 given $\mathbf{z}_{(2)}$ is then $\hat{\theta}_b = \frac{10+9}{20+20} = 0.95$

Once all $k+1$ density parameter estimates have been calculated, they can be used to calculate the clustering likelihood given $\mathbf{z}_{(2)}$ presented in Equation 4.5. In the next section, we present and discuss expressions for the efficient calculation of these parameters for directed networks.

4.4 Efficient likelihood-based directed network clustering

In Chapter 3, several expressions were introduced to streamline the calculation of parameter estimates and the change-in-loglikelihood from an

accepted clustering $\mathbf{z}^0_{(k)}$ to a newly-proposed clustering $\mathbf{z}^1_{(k)}$ given an undirected network. These were useful results since likelihood Equation 4.5 is often converted to loglikelihood with no change in the solution clustering, and calculating the parameters required to evaluate the change-in-loglikelihood when reclustering is extremely computationally expensive.

In this section, we present the corresponding expressions required to efficiently calculate the change-in-loglikelihood and density parameters when reclustering a directed network.

4.5 Theorems

Proofs for Theorems 4.5.1 $\sim$ 4.5.5 are found in the Appendix. All networks G, with vertex set V and edge set E, referred to in this section are simple, directed networks.

Theorem 4.5.1 demonstrates how the change-in-loglikelihood due to a reclustering can be greatly simplified given some simple parameter constraints.

Theorem 4.5.1. *Assign each vertex $v \in V$ from $G = (V, E)$ into one of k clusters. Assume the edges within cluster $h \in \{1, 2..., k\}$ are distributed according to density $f_h(y_h|\theta_h, \mathbf{z}_{(k)})$ and the edges between clusters are distributed according to density $f_b(y_b|\theta_b, \mathbf{z}_{(k)})$. The loglikelihood of the clustering $\mathbf{z}_{(k)}$ is then written as*

$$\ln L(\theta_1, \theta_2, ..., \theta_k, \theta_b|\mathbf{y}, \mathbf{z}_{(k)}) = ln[\{\prod_{h=1}^{k} f_h(y_h|\theta_h, \mathbf{z}_{(k)})\} f_b(y_b|\theta_b, \mathbf{z}_{(k)})], \tag{4.8}$$

where each θ_h and θ_b are estimated by maximum likelihood estimators $\hat{\theta}_h$ and $\hat{\theta}_b$, respectively.

Then, consider a reclustering by reassigning vertex l^* from cluster ℓ to cluster j. If the parameter estimates for the within-cluster edge densities other than f_ℓ and f_j are unaffected by the reclustering, i.e. $\hat{\theta}^1_h = \hat{\theta}^0_h$, $h \notin \{\ell, j\}$, then

the change-in-loglikelihood due to the reclustering can be written:

$$\Delta \ln L = ln \left[\frac{f_\ell(y_\ell|\hat{\theta}^1_\ell, \mathbf{z}^1_{(k)}) f_j(y_j|\hat{\theta}^1_j, \mathbf{z}^1_{(k)}) f_b(y_b|\hat{\theta}^1_b, \mathbf{z}^1_{(k)})}{f_\ell(y_\ell|\hat{\theta}^0_\ell, \mathbf{z}^0_{(k)}) f_j(y_j|\hat{\theta}^0_j, \mathbf{z}^0_{(k)}) f_b(y_b|\hat{\theta}^0_b, \mathbf{z}^0_{(k)})} \right], \tag{4.9}$$

where superscripts 0 and 1 indicate pre- and post-reclustering states, respectively.

Theorems 4.5.2 and 4.5.3 demonstrate how post-reclustering edge weight sums can be calculated from pre-reclustering information.

Let $\boldsymbol{\omega}_j$ be a N x 1 cluster membership vector for cluster j such that element $\omega_{i,j} = 1$ if vertex i belongs to cluster j, 0 otherwise. Let $\boldsymbol{\omega}$ be the N x k cluster membership matrix with ij^{th} element $= \boldsymbol{\omega}_{i,j}$. Let $\mathbf{1_s}$ be a N x 1 vector of 0's and a single 1 in the sth position. If vertex l^* is removed from cluster ℓ and placed in cluster j, then:

$$\boldsymbol{\omega}^1_\ell = \boldsymbol{\omega}^0_\ell - \mathbf{1}_{l^*} \text{ and } \boldsymbol{\omega}^1_j = \boldsymbol{\omega}^0_j + \mathbf{1}_{l^*}.$$

Given clustering $\mathbf{z}_{(k)}$ on a directed network, $\mathbf{OBS} = \boldsymbol{\omega}^T \mathbf{A} \boldsymbol{\omega}$ is a (possibly) non-symmetric k x k matrix with element $obs_{s,t} = \boldsymbol{\omega}^T_s \mathbf{A} \boldsymbol{\omega}_t$, the sum of the observed edge weights between clusters s and t in both directions. Unlike the undirected network case where $\boldsymbol{\omega}^T_s \mathbf{A} \boldsymbol{\omega}_s$ double counts the edge weights among vertices in cluster s and requires an adjustment, $\boldsymbol{\omega}^T_s \mathbf{A} \boldsymbol{\omega}_s$ for directed networks need no adjustment

Theorem 4.5.2. *Assign each vertex $v \in V$ from $G = (V, E)$ into one of k clusters. Then, consider a reclustering by reassigning vertex l^* from cluster ℓ to cluster j. The elements of* **OBS** *post-reclustering can be written completely in terms of pre-reclustering information. Specifically, the elements are:*

1. $obs^1_{s,t} = obs^0_{s,t}$
2. $obs^1_{t,s} = obs^0_{t,s}$
3. $obs^1_{s,s} = obs^0_{s,s}$
4. $obs^1_{\ell,s} = obs^0_{\ell,s} - Z^0_{to\ s}$
5. $obs^1_{s,\ell} = obs^0_{s,\ell} - Z^0_{from\ s}$
6. $obs^1_{j,s} = obs^0_{j,s} + Z^0_{to\ s}$
7. $obs^1_{s,j} = obs^0_{j,s} + Z^0_{from\ s}$
8. $obs^1_{\ell,j} = obs^0_{\ell,j} - Z^0_{to\ j} + Z^0_{from\ \ell}$

9. $obs^1_{j,\ell} = obs^0_{j,\ell} + Z^0_{to\ \ell} - Z^0_{from\ j}$

10. $obs^1_{\ell,\ell} = obs^0_{\ell,\ell} - Z^0_{to\ \ell} - Z^0_{from\ \ell}$

11. $obs^1_{j,j} = obs^0_{j,j} + Z^0_{to\ j} + Z^0_{from\ j}$,

where $s, t \notin \{\ell, j\}$, superscripts 0 and 1 indicate pre- and post-reclustering states, respectively, $Z^0_{to\ h}$ is the sum of edge weights extending from vertex l^* to vertices in cluster h pre-reclustering, and $Z^0_{from\ h}$ is the sum of edge weights extending from vertices in cluster h to vertex l^* pre-reclustering, $h \in \{1, 2, ..., k\}$.

Theorem 4.5.3. *Assign each vertex $v \in V$ from $G = (V, E)$ into one of k clusters. Then, consider a reclustering by reassigning vertex l^* from cluster ℓ to cluster j. The sum of the distinct off-diagonal elements of* **OBS** *post-reclustering can be written completely in terms of pre-reclustering information. Specifically:*

$$obs^1_b = \sum obs^1_{s,t} = obs^0_b - Z^0_{to\ j} - Z^0_{from\ j} + Z^0_{to\ \ell} + Z^0_{from\ \ell}, \tag{4.10}$$

where $s, t \in \{1, 2, ...k\}$ and $s \neq t$, superscripts 0 and 1 indicate pre- and post-reclustering states, respectively, $Z^0_{to\ h}$ is the sum of edge weights extending from vertex l^* to vertices in cluster h pre-reclustering, and $Z^0_{from\ h}$ is the sum of edge weights extending from vertices in cluster h to vertex l^* pre-reclustering, h $\in \{1, 2, ..., k\}$.

Theorems 4.5.4 and 4.5.5 demonstrate how the maximum possible number of edges within and between clusters post-reclustering can be calculated from pre-reclustering information.

Given clustering $\mathbf{z}_{(k)}$ on a directed network, **POS** is a symmetric k x k matrix with element $pos_{s,t}$ containing the maximum number of distinct edges that can extend from the vertices of clusters s to the vertices of cluster t. This value is $pos_{s,t} = \binom{n_s+n_t}{2} - \binom{n_s}{2} - \binom{n_t}{2} = n_s n_t$ when $s \neq t$ and $pos_{s,t} = \frac{n_s(n_s-1)}{2}$ when $s =$ t, where n_h is the number of distinct vertices in cluster h $\in \{1, 2, ..., k\}$. Note that for fixed n_s and n_t, $pos_{s,t} = pos_{t,s}$.

Theorem 4.5.4. *Assign each vertex $v \in V$ from $G = (V, E)$ into one of k*

clusters. Then, consider a reclustering by reassigning vertex l^* from cluster ℓ to cluster j. The elements of **POS** *post-reclustering can be written completely in terms of pre-reclustering information. Specifically, the elements are:*

1. $pos^1_{s,t} = pos^0_{s,t}$
2. $pos^1_{t,s} = pos^0_{t,s}$
3. $pos^1_{s,s} = pos^0_{s,s}$
4. $pos^1_{\ell,s} = n^0_s(n^0_\ell - 1)$
5. $pos^1_{s,\ell} = n^0_s(n^0_\ell - 1)$
6. $pos^1_{j,s} = n^0_s(n^0_j + 1)$
7. $pos^1_{s,j} = n^0_s(n^0_j + 1)$
8. $pos^1_{\ell,j} = (n^0_\ell - 1)(n^0_j + 1)$
9. $pos^1_{j,\ell} = (n^0_\ell - 1)(n^0_j + 1)$
10. $pos^1_{\ell,\ell} = (n^0_\ell - 1)(n^0_\ell - 2)$
11. $pos^1_{j,j} = (n^0_j + 1)n^0_j$,

where $s, t \notin \{\ell, j\}$, superscripts 0 and 1 indicate pre- and post-reclustering states, respectively, and n^0_t is the number of vertices in cluster t pre-reclustering.

Theorem 4.5.5. *Assign each vertex $v \in V$ from $G = (V, E)$ into one of k clusters. Then, consider a reclustering by reassigning vertex l^* from cluster ℓ to cluster j. The sum of the distinct off-diagonal elements of* **POS** *post-reclustering can be written completely in terms of pre-reclustering information. Specifically:*

$$pos^1_b = \sum pos^1_{s,t} = pos^0_b + 2(n^0_\ell - n^0_j - 1), \tag{4.11}$$

where $s, t \in \{1, 2, ...k\}$ and $s \neq t$, superscripts 0 and 1 indicate pre- and post-reclustering states, respectively, and n^0_h is the number of vertices in cluster h pre-reclustering, $h \in \{1, 2, ..., k\}$.

4.5.1 Easily calculated parameter components

The k **OBS** elements $obs_{h,h}$, $h \in \{1, 2, ..., k\}$, $h \neq \{\ell, j\}$, $obs_{\ell,\ell}$ and $obs_{j,j}$ correspond to the k sufficient statistics y_h in Section 4.3, and the single calculated value obs_b corresponds to the sufficient statistic y_b of the same section. Thus, the expressions in the preceding Theorems provide an efficient means of

calculating sufficient statistics for a proposed solution, given the sufficient statistics from the current solution. This necessarily requires that the entire edge set E be initially read so as to populate **OBS** for the initial cluster assignment during optimization. After which the elements of **OBS** can be updated using the above expressions if a proposed solution is accepted as the new current solution or, if rejected, remain unchanged until a new proposed solution is considered.

The k **POS** elements $pos_{h,h}$, $h \in \{1, 2, ..., k\}$, $h \neq \{\ell, j\}$, $pos_{\ell,\ell}$, $pos_{j,j}$ and the single calculated value pos_b are also easily updated when considering a proposed solution, given that the number of vertices in each cluster, n_h, n_ℓ, and n_j, are available for the current solution. This, of course, means that the number of vertices per cluster must be stored during the initial cluster assignment of an optimization algorithm. After which, only the elements n_ℓ and n_j need by updated by subtracting 1 from n_ℓ and adding 1 to n_j if a proposed solution is accepted as the new current solution. Otherwise, the counts for the current solution remained unchanged until a new proposed solution is considered.

For certain distributions, particularly the binomial and Poisson densities, crucial to the analysis of discretely-distributed networks, maximum likelihood parameter estimates can easily be calculated from the elements of **OBS** and **POS**. namely $\hat{\theta}_s = \frac{obs_{s,s}}{pos_{s,s}}, s \in \{1, 2, ..., k\}$ and $\hat{\theta}_b = \frac{\sum obs_{s,t}}{\sum pos_{s,t}}, s, t \in \{1, 2, ..., k\}$, $s \neq t$. Thus, Theorems 4.5.2 $\sim$ 4.5.5 provide efficient methods for calculating the parameter estimates needed to calculate the loglikelihood for a proposed solution.

Further, Theorem 4.5.1 shows that the expression for the change in observed loglikelihood due to the reclustering of G by moving vertex l^* from cluster ℓ to cluster j can be greatly simplified if parameter estimates $\hat{\theta}_h$ for within-cluster edge densities other than f_ℓ and f_j remain unchanged, i.e. $\hat{\theta}^1_h = \hat{\theta}^0_h$, $h \notin \{\ell, j\}$. Rather than calculate the full expression for the change in loglikelihood,

$$\Delta \ln L(\hat{\theta}|\mathbf{y}, \mathbf{z}_{(k)}) = ln \left[\frac{\prod_{s \neq \ell, j} f_s(y_s|\hat{\theta}^1_s, \mathbf{z}^1_{(k)})}{\prod_{s \neq \ell, j} f_s(y_s|\hat{\theta}^0_s, \mathbf{z}^0_{(k)})} \frac{f_\ell(y_\ell|\hat{\theta}^1_\ell, \mathbf{z}^1_{(k)}) f_j(y_j|\hat{\theta}^1_j, \mathbf{z}^1_{(k)})}{f_\ell(y_\ell|\hat{\theta}^0_\ell, \mathbf{z}^0_{(k)}) f_j(y_j|\hat{\theta}^0_j, \mathbf{z}^0_{(k)})} \frac{f_b(y_b|\hat{\theta}^1_b, \mathbf{z}^1_{(k)})}{f_b(y_b|\hat{\theta}^0_b, \mathbf{z}^0_{(k)})} \right],$$

we can use the simplified expression (4.9) from Theorem 4.5.1 which only requires the two within-cluster densities f_ℓ and f_j and the single between-cluster density f_b. If the parameter estimates for densities other than f_ℓ and f_j can be calculated from suitable **OBS** and **POS** matrix entries, then Theorems 4.5.2 and 4.5.4 show that the parameter estimates will not change as a result of the reclustering. Thus, given this important assumption about the parameters, we can use Theorem 4.5.1.

Together, these theorems significantly reduce the computational requirements for calculating the change-in-loglikelihood for each proposed solution considered in an optimization algorithm. Without them, we would be required to traverse the entire edge set E in order to populate an **OBS** matrix for each proposed solution. If the directed network data is stored in adjacency matrix format, this would require reading $N(N-1)$ elements. An adjacency matrix for a real-world network would contain many zeros, none of which contribute to the sufficient statistics and all of which are wasteful. If a more efficient storage format containing only non-zero edge information were used, such as an edge list or an adjacency list, then the full edge set E must still be read to calculate the sufficient statistics, and thus the parameter estimates, for each proposed reclustering.

Theorems 4.5.2 $\sim$ 4.5.5, however, show how to calculate information needed for the parameter estimates for a proposed reclustering almost immediately from the parameter estimates from the current clustering. The only "new" information necessary to update the **OBS** matrix elements required for the parameters of Theorem 4.5.1 is $Z^0_{\text{to }\ell}$, $Z^0_{\text{from }\ell}$, $Z^0_{\text{to j}}$, and $Z^0_{\text{from j}}$. This information can be obtained by referencing the pre-reclustering clustering assignment vector $\mathbf{z}_{(k)}$ against 1) l^*'s row and column in the adjacency matrix, 2) entries containing edges to, or edges from, l^* in the edge list, or 3) l^*'s row in the adjacency list. For non-binary edge weights, 2) and 3) assume that the edge and adjacency lists also contain weight information. While 1) is an improvement over reading in the

$N(N-1)$ distinct adjacency matrix elements, requiring only $2N-1$ elements to be read, many of those elements will still likely be zero, and thus unhelpful. Options 2) and 3) are more efficient as the edge and adjacency lists contain only non-zero edge values. To calculate $Z^0_{\text{to }\ell}$, $Z^0_{\text{from }\ell}$, $Z^0_{\text{to j}}$, and $Z^0_{\text{from j}}$ we need only to sum the weights of edges extending to vertex l^* by cluster and sum the weight of the edges extending from vertex l^* by cluster, the total number of which are equal to the degree of l^*, d_{l^*}. In the case of an adjacency list, this involves reading only the elements on the list row belong to vertex l^*. For any proposed reclustering, the expected number of elements necessary for calculating the required elements of the **OBS** matrix is $\mathrm{E}[d_{l^*}]$ = average degree of $G \ll 2N-1 \ll N(N-1)$.

4.5.2 Reduced updating requirements

In a naive implementation of likelihood-based clustering, the $4(k+\binom{k}{2})$ distinct elements of the post-reclustering $\mathbf{OBS}^1$ and $\mathbf{POS}^1$ matrices must be calculated for each proposed reclustering. If the reclustering is accepted, then the $\mathbf{OBS}^1$ and $\mathbf{POS}^1$ matrices are reassigned to pre-reclustering $\mathbf{OBS}^0$ and $\mathbf{POS}^0$ before considering the next proposed reclustering. The $2\binom{k}{2}$ distinct off-diagonal $obs^1_{s,t}$ and the $2\binom{k}{2}$ distinct off-diagonal $pos^1_{s,t}$ elements must be maintained in order to calculate obs^1_b and pos^1_b, elements required to calculate the between-cluster density parameter estimate $\hat{\theta}^1_b$.

Theorems 4.5.3 and 4.5.5, however, provide a means of calculating obs^1_b and pos^1_b directly from obs^0_b and pos^0_b. It's now only necessary to maintain the $2(k+1)$ elements consisting of obs^0_h, pos^0_h, obs^0_b and pos^0_b, h $\in \{1,2,...,k\}$ from iteration to iteration and if a reclustering is accepted, only obs^1_i and $pos^1_i, i \in \{\ell, j, b\}$ need be updated. This efficiency increases as k grows large since it is no longer necessary to store or update the remaining $2(k^2-k-1)$ elements in the full **OBS** and **POS** matrices.

4.6 A likelihood ratio test for detected clusters

The likelihood ratio test for detected clusters in undirected networks was derived in Section 2.4. Here we summarize the derivation and the formal test, and note that the test needs no modification for use with directed networks.

The likelihood ratio test is convenient for comparing nested models. In particular, we are interested in testing the null hypothesis of a single cluster, i.e. there is no underlying structure to the network and edges are distributed at random, versus the alternative hypothesis that the network is properly segmented into k clusters. The test statistic for the likelihood ratio test is

$$D = -2(\ln \frac{L_0(\theta_0|\mathbf{y})}{L_1(\boldsymbol{\theta}|\mathbf{y}, \mathbf{z}_{(k)})}) \tag{4.12}$$

for a clustering $\mathbf{z}_{(k)}$ where $L_0(\theta_0|\mathbf{y})$ denotes the likelihood function under the null hypothesis and $L_1(\boldsymbol{\theta}|\mathbf{y}, \mathbf{z}_{(k)})$ denotes the likelihood function under the alternative hypothesis. D approximately follows a χ^2 distribution with k degrees of freedom.

One assumption of the likelihood ratio test is that the data are independently distributed. Since, by the assumptions in Section 2.3, we have no external information by which to specify the group membership $\mathbf{z}_{(k)}$ explicitly, it must be estimated from (and is therefore not independent of) the vector of sufficient statistics $\mathbf{y}$. Consequently, when $\mathbf{z}_{(k)}$ is unknown and must be estimated using $\mathbf{y}$, the likelihood ratio statistic D is not asymptotically χ^2_k under the null hypothesis.

For a fixed N and k, the number of possible clusterings $\mathbf{z}_{(k)}$ is finite, and thus the values of D are also finite. We are, in any case, primarily interested in the test statistic D_{max} among all possible test statistics D for a given N and k. Since it is not feasible to test all possible $\mathbf{z}_{(k)}$ to find D_{max} even for relatively small networks, another approach must be taken. Let $\mathbf{D}$ be a vector of cluster assignments $\mathbf{z}_{(k)}$ sampled *with replacement* from the set of all possible k-cluster assignments. Since the sample is taken with replacement, the number of draws,

M, until D_{max} is selected follows a negative binomial distribution with probability of selection $p = \frac{1}{S(N,k)}$ and the number of successes $r = 1$, $S(N, k)$ denotes the number of ways to partition N elements into k subsets.

Since the $\mathbf{z}_{(k)}$'s (and hence the D's) are randomly selected, group assignments are made independent of the sample. Consequently, under the null hypothesis, the asymptotic distribution of each $D_i \in \mathbf{D}$ is χ^2_k. The pdf of the maximum of a random sample of size G from a χ^2 distribution is the

$$f_{D_{(G)}} = GF(u)^{G-1} f(u) \tag{4.13}$$

where $F(u)$ and $f(u)$ are the cdf and pdf of the χ^2_k distribution, respectively. Because the support of the χ^2 distribution is unbounded on the right, increasing values of G produce increasing values of the sample maximum, $D_{(G)}$. By choosing an appropriate value for G an estimate for the distribution of D_{max} under the null hypothesis can be obtained. Recalling that $E(M) = S(N, k) - 1$ is the expected number of draws required to select the maximum at random, set $G = E(M)$.

Hence, the $100(1 - \alpha)$th percentiles of the distribution of $D_{(G)}$ provide approximate critical values for a one-tailed test for the significance of detected clusters. Based on Equation 4.13, approximate critical values then are calculated from

$$C_{1-\alpha}(N, k) = F^{-1}(\sqrt[G]{1 - \alpha}). \tag{4.14}$$

In practice, the test is conducted by computing test statistic D for a given N and "best" k, and subsequently comparing to the critical value $C_{1-\alpha}(N, k)$. If $D > C_{1-\alpha}(N, k)$, then the test concludes in favor of the alternative hypothesis of k clusters. That is, it is unlikely that the groups detected by the clustering effort could have occurred by chance.

Note that edges in general, and the direction of edges in particular, are not part of the test's derivation. In fact, the only qualities of a network required to derive the test are N and k. As such, the test may be used without modification

for directed networks. The critical values of Table 5.1 in the Appendix may also be applied to directed networks.

4.7 Application to real-world networks

In this section, we cluster two real-world networks in an effort to demonstrate exactly how the proposed clustering approach is applied in practice. In particular, we consider the a friendship network of Hansell (1984) and network of Adamic and Glance (2005) consisting of liberal and conservative blogs during the 2004 presidential election.

4.7.1 Hansell's friendship network

A study was conducted by Hansell (1984) of 27 elementary school students. Each student was asked to rate their friendship by selecting a face with one of three expressions for each of their classmates: a big smile, a moderate smile, and no smile. A big smile was considered indicative of a strong friendship and recorded as a 1, and a moderate or no smile was considered indicative of a weak friendship and recorded as a 0. The gender of the students was also recorded. Students $1 \sim 13$ are male, and students $14 \sim 27$ are female The data collected from this experiment is reproduced in Figure 4.3 from Wang and Wong (1987).

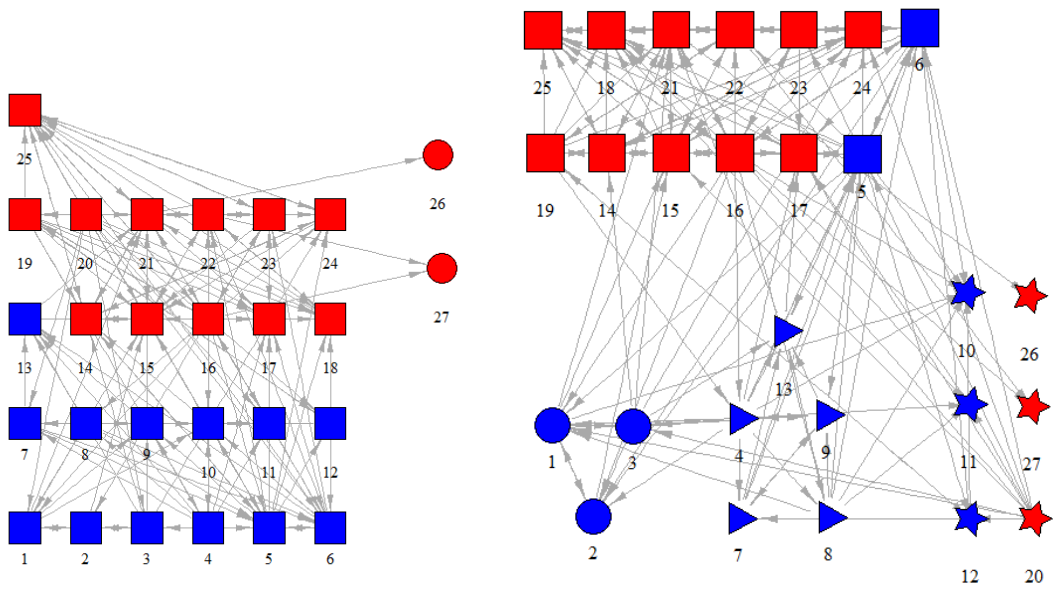
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	0	0	1	0	0	0
4	1	1	1	0	1	1	1	1	1	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
5	1	1	0	0	0	0	0	0	1	1	0	1	0	1	1	0	1	1	0	0	0	1	1	1	1	0	0
6	0	1	0	0	1	0	0	0	0	1	0	0	1	1	0	1	1	1	0	0	0	0	1	0	1	0	0
7	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	0	0	0	1	1	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
14	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	0	1	1	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0	1	0	0
16	1	1	0	1	1	1	0	1	0	1	1	0	0	1	1	0	1	1	1	0	1	1	1	1	1	0	1
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
19	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1	1	1	1	0	0	1	1	0	1	1	0	0
20	1	0	1	0	1	1	1	0	0	0	1	1	0	0	1	0	1	1	0	0	0	0	0	1	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	1	0
22	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	1	1	0	1
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	0	1	1	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	0	1	1	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.3: Data from Hansell's 1984 survey of elementary school children. A 1 indicates strong friendship and a 0 indicates weak friendship. Students $1 \sim 13$ are male and $14 \sim 27$ are female.

The directed Hansell network was clustered using both modularity and the proposed objective function (assuming a binomial density) in a simulated annealing algorithm[1] for values of $k \in \{1, ..., 10\}$. The clusterings that produced the maximum modularity and minimum BIC over the range of k were taken as the solutions for the modularity and proposed likelihood objective functions, respectively. Figure 4.4 shows the results for clustering using the modularity objective function and proposed objective functions.

Maximizing modularity results in a solution of $k = 2$ clusters, but the clusters do not correspond to gender. Rather, students {26,27} are in 1 cluster and the remainder in another. Referring to Figure 4.3 reveals that students {26,27} expressed no strong friendship with anyone in the class. While student 10 also expressed no strong friendship with anyone in the class, 6 male students

[1]Cooling schedule for maximizing both objective functions: Initial temp=1, Cooling Rate: 0.999, Temp Length: 100.



(a) Modularity (b) Likelihood

Figure 4.4: Clustering results for the Hansell network. Blue indicates males and red indicates females. The solution clusters are differentiated by their shapes.

and 1 female student expressed strong friendship with him. Only 3 students expressed strong friendship with students {26,27}, though. Thus it appears that modularity is only able to distinguish between students who who have virtually no friends in the class and all other students, but cannot distinguish between gender or along any other possible attributes.

Minimizing BIC results in a solution of $k = 4$ clusters. The solution is not split perfectly along gender, but does reveal some interesting associations. Clusters with students $\{1, 2, 3\}$ (circles) and $\{4, 7, 8, 9, 13\}$ (triangles) reveal potential male student cliques. The (square) clustering of students $\{5, 6, 14, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25\}$ is primarily female with 2 male students. Not only did $\{5, 6\}$ express strong friendship with more female students than male, the number of female students with which $\{5, 6\}$ expressed strong friendship was much higher than any of the other male students. Thus these 2 students were clearly more aligned with females than males. The $\{10, 11, 12, 20, 26, 27\}$ (triangle) clustering of students might be labeled the "non-social" cluster. $\{26, 27\}$, as before, express no friendship with their fellow

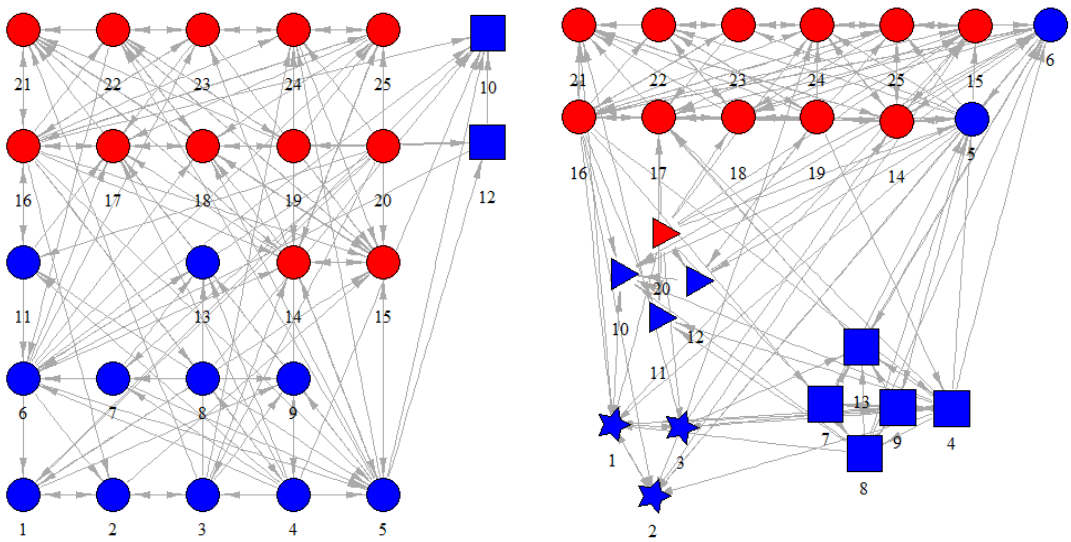
students. Now student 10 has joined them, along with students $\{11, 12, 20\}$. Students $\{11, 12\}$ expressed strong friendship with only a few classmates and received expressions of strong friendship from only a few classmates. Student 20 expressed strong friendship for 11 of the 27 total students but received no expressions of strong friendship in return. Thus the non-social cluster consists of those students who either don't feel strong friendship for most of their classmates, or for whom their classmates don't feel strong friendship.

The solutions under modularity and likelihood identifies $k = 2$ and $k = 4$ as the optimal number of clusters, respectively. We calculate the critical values at the 95% level for each solution as $C(27, 2) = 41.984$ and $C(27, 4) = 81.91$. Comparing the test statistics $D_{Mod} = 40.058 < C(27, 2)$ and $D_{Lik} = 121.222 > C(27, 4)$, we can conclude that the likelihood solution is statistically significant while the modularity solution is not. The p-values for the solution test statistics are $p_{Mod} = 0.126$ and $p_{Lik} = 0.0$.

Students $\{26, 27\}$ appear to be outliers in that they express no strong friendship for anyone in the class and the majority of the class express no strong friendship for them either. These students were removed from the network and the clustering repeated under the same conditions as before. The results of this second clustering are in Figure 4.5.

Maximizing modularity has again split the class into 2 clusters, putting students $\{10, 12\}$ into a single (square) cluster and the remaining students into the other (circle). With $\{26, 27\}$ removed, $\{10, 12\}$ appear to be the least social of the remaining students, expressing little strong friendship with their classmates and receiving few strong expressions in return.

Minimizing BIC produced the same clustering results even when students $\{26, 27\}$ were removed. The only difference is that the non-social cluster now consists only of $\{10, 11, 12, 20\}$. The male cliques $\{1, 2, 3\}$ and $\{4, 7, 8, 9, 13\}$ are still apparent and male students $\{5, 6\}$ are still clustered with the primarily female cluster $\{5, 6, 14, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25\}$ (circle).



(a) Modularity (b) Likelihood

Figure 4.5: Clustering results for the Hansell network without students {26,27}. Blue indicates males and red indicates females. The solution clusters are differentiated by their shapes.

Comparing the new solution test statistics $D_{Mod} = 18.074 < C(25, 2) = 39.211$ and $D_{Lik} = 98.782 > C(25, 4) = 76.226$, we again conclude that the likelihood solution is statistically significant while the modularity solution is not. The p-values for the solution test statistics are $p_{Mod} = 1$ and $p_{Lik} = 0.0$, the computable limits for statistical insignificance and significance, respectively.

Although modularity has been shown to produce results inferior to those of likelihood in the current example, it may simply be that modularity requires a more stringent simulated annealing cooling schedule (i.e., higher initial temperature, cooling rate and/or temperature length) to produce results competitive with likelihood. An exhaustive search for cooling schedules that produce equivalent results between the two methods in general or for a given network, however, is beyond the scope of this manuscript.

4.7.2 2004 Presidential election blogs

Adamic and Glance (2005) used online web directories to manually create a

citation network of 1,490 blogs in total, 758 liberal and 732 conservative. A directed edge was said to exist if one of these blogs contained a URL to another of the blogs. As expected, the authors reported dense connections between blogs of the same political orientation and sparse connections between them. The bifurcate nature of this real-world network and the expectation of a politically-oriented "ground truth" solution presents a good opportunity to apply both the modularity and proposed objective functions for clustering.

Before clustering, blogs that neither linked to, or were linked from, other blogs in the data set were removed, leaving 1,224 blogs in total. Although it is possible to cluster networks with singleton vertices, it wasn't felt that these are particularly informative. The resulting network appears in Figure 4.6 with "liberal" blogs colored blue and "conservative" blogs colored red. The blogs were clustered again using both the modularity objective function and the proposed objective function (assuming a binomial density) in a simulated annealing algorithm [2].

[2]Cooling schedule for maximizing both objective functions: Initial temp=1, Cooling Rate: 0.99, Temp Length: 100.

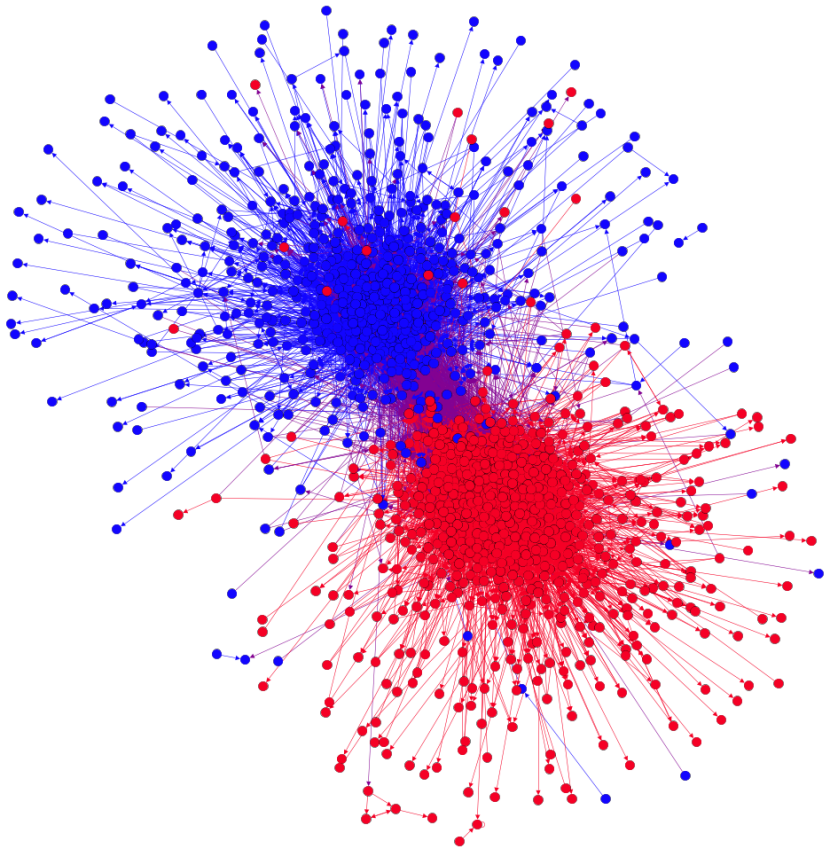


Figure 4.6: 2004 Presidential Election Blogs

The solution identified by the modularity and likelihood objective functions resulted in optimal number of clusters $k = 2$ and $k = 4$, respectively. The solution test statistics were $D_{Mod} = 1695.12$ and $D_{Lik} = 40902.0$. The required critical value for modularity is $C(1224, 2) = 1701.378$ but the critical value for the likelihood solution was too large for our current computing resources to calculate. The p-values for the solution test statistics were $p_{Mod} = 0.690$ and $p_{Lik} = 0.0$. The clustering derived under modularity is shown to be statistically insignificant while that of likelihood is significant.

The cooling rate was increased from 0.99 to 0.999 and the clustering results repeated. In this case, the optimal number of clusters identified by modularity and likelihood were $k = 8$ and $k = 6$ respectively. The solution test statistics were $D_{Mod} = 17842.9$ and $D_{Lik} = 48011.1$. In this case, the critical values for both solutions were too large for our current computing resources to calculate.

The p-values for the solution test statistics were $p_{Mod} = 0.0$ and $p_{Lik} = 0.0$. With the increased cooling rate, both objective functions were able to find statistically significant solutions.

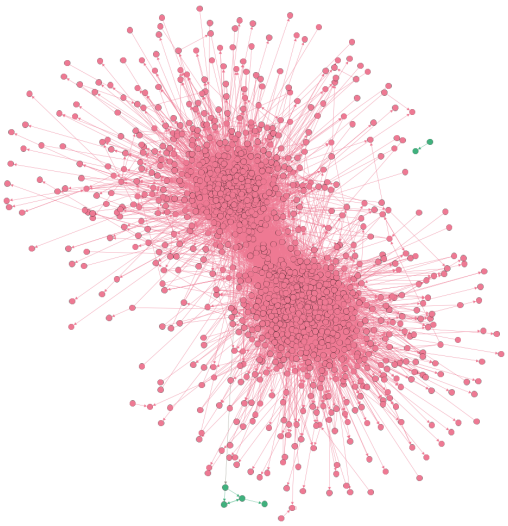
Although modularity identifies $k = 2$ clusters at cooling rate=0.99, Figure 4.7a shows that almost all vertices have been assigned to a single cluster, making the clustering largely uninformative. Likelihood, on the other hand, has identified a core group of blogs in Figure 4.7b. When the cooling rate was increased to 0.999, modularity was largely able to recover Adamic and Glance (2005)'s designation of "liberal" and "conservative" as shown in Figure 4.7c. In Figure 4.7d, likelihood has further identified the core blogs within each label and assigned the remaining blogs to peripheral clusters.

4.8 Performance evaluations

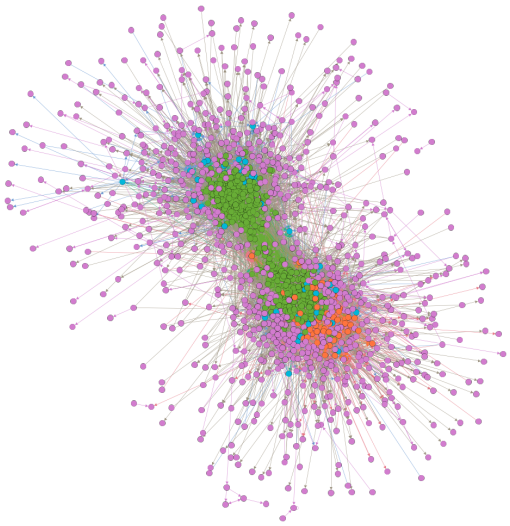
In this section, we evaluate the clustering performance of the proposed objective function when applied to the so called LFR benchmark graphs proposed by Lancichinetti et al. (2008). These graphs possess some properties of real-world networks, specifically, non-homogeneous degree distributions and community sizes. In the next subsection, we provide a brief description of the LFR benchmark graphs, to include the parameters involved in generating these graphs. We also discuss the results of a simulation study uses to assess the performance of the proposed clustering framework.

4.8.1 LFR benchmark graphs

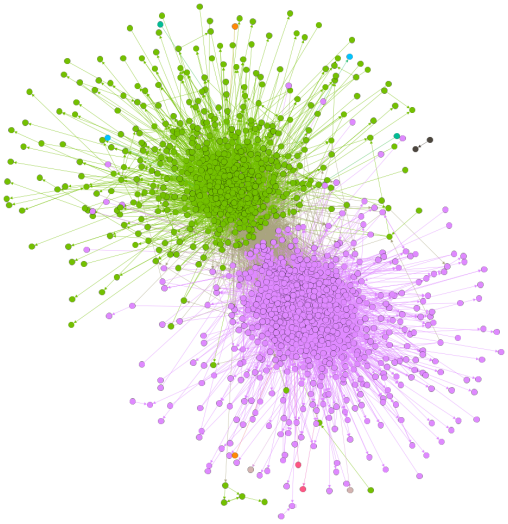
Lancichinetti et al. (2008) first proposed the undirected LFR benchmark networks as a means to test community detection algorithms. These networks possess some properties in common with many networks of interest, such as non-homogeneous degree distributions and non-homogeneous cluster sizes. For the LFR benchmark networks, the degree distribution follows a power law with parameter $2 \leq \gamma \leq 3$ and the cluster size distribution follows a power law with



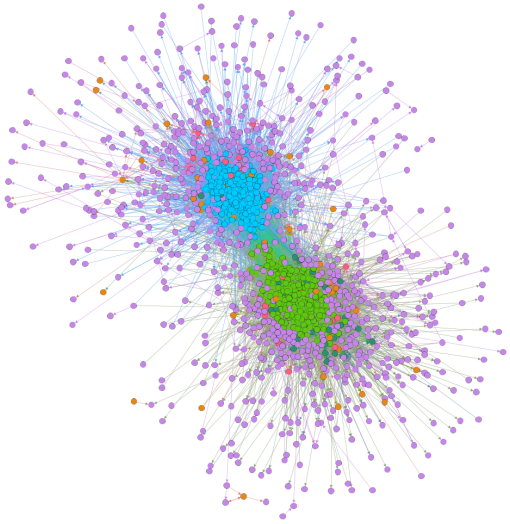
(a) Modularity



(b) Likelihood



(c) Modularity



(d) Likelihood

Figure 4.7: 2004 presidential election blogs clustered under modularity and likelihood

parameter $1 \leq \beta \leq 2$, which encompasses a vast array of networks with non-homogeneous degree and community size distributions. Further, there are three additional parameters: (1) the mixing parameter μ, which represents the fraction of a node's edges connected to other nodes not contained in the same group (and thus, $1 - \mu$ represents the fraction of a node's edges connection to other nodes contained in the same group), (2) the average degree of the network, or *Ave Degree*, and (3) the maximum degree of the network, or *Max Degree*. Undirected LFR benchmarks are detailed further in Section 2.6.1.

Lancichinetti and Fortunato (2009) later proposed a directed version of the LFR benchmark network generator. For directed LFR benchmark networks, the degree distribution, cluster size distribution and mixing parameter have parameters as before, $2 \leq \gamma \leq 3$, $1 \leq \beta \leq 2$ and $0 < \mu < 1$. However, γ and β are referred to as τ_1 and τ_2, respectively, in Lancichinetti and Fortunato (2009). Rather than specify *Ave Degree* and *Max Degree* as required in the undirected case, the directed LFR network generator requires *Ave In-Degree* and *Max In-Degree* parameters, the average and maximum number of edges, respectively, directed toward a particular vertex. In addition to mimicking real-world properties, both the directed and undirected LFR network generators produce "ground truth" cluster assignments that can be compared to the output of clustering algorithms.

In the next subsection we discuss the results of a Monte Carlo simulation study used to evaluate the clustering performance of the proposed clustering framework, relative to that achieved by maximizing modularity.

4.8.2 Clustering performance of the proposed clustering framework when applied to the directed LFR benchmark graphs

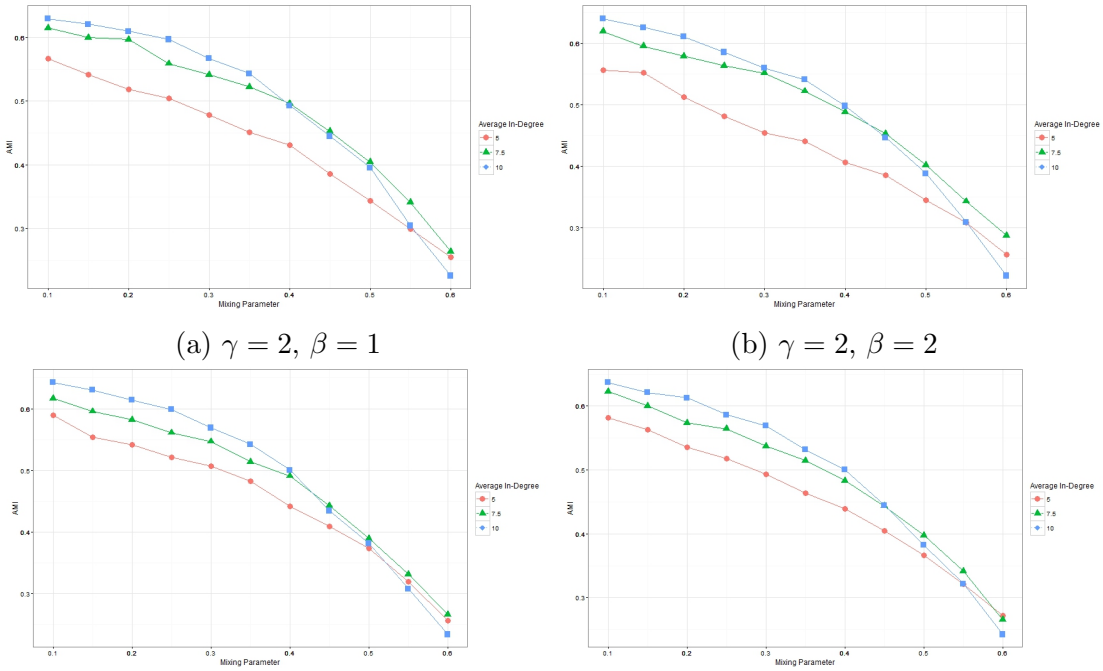
In this subsection we report the results of a Monte Carlo simulation study where we applied our proposed clustering framework to the LFR benchmark graphs to assess its expected performance. We investigated clustering

performance as a result of maximizing the proposed likelihood objective function, and for comparison purposes, we also investigated that achieved by maximizing the modularity metric. In what follows, details of the simulation model are discussed.

We consider networks of size $N = 100$ nodes with *Max In-Degree* $= 10$, and *Ave In-Degree* $= 5$, 7.5 and 10. For each value of *Ave In-Degree*, we considered four different combinations for the parameters γ and β; namely $(\gamma, \beta) = (2, 1), (2, 2), (3, 1), (3, 2)$, which encompasses the extremes of the ranges of these parameters. Finally, we considered the values of the mixing parameter μ between 0.1 and 0.6, in increments of 0.05. For any given combination of the LFR benchmark parameter settings studied[3], we generated 100 independent benchmark network for a total of 13,200 networks. For each simulated network we used simulated annealing to find the group membership vectors that maximize the modularity metric and the proposed objective function, respectively, for each $k \in \{10, 11, ...20\}$. Once these vectors were found, we selected the "best" k for each network by retaining the solutions for each objective function that minimized BIC.

To judge the quality of the solutions, we used adjusted mutual information (AMI), outlined in Section 2.5. AMI is bound in [0,1], with 0 indicating two cluster membership vectors have nothing in common and 1 indicating that two cluster membership vectors are perfect matches. We computed the AMI between the known "ground-truth" cluster membership vector (which was given) and the estimated cluster membership vectors obtained by maximizing the two objective functions. The average of the AMI values obtained via maximizing modularity, as well as that obtained by maximizing the proposed likelihood objective function, was then computed over the independently simulated graphs. Figures 4.8 and 4.9 show the estimated AMI curves over the parameter space μ corresponding to modularity and the proposed method, respectively.

[3] i.e., γ, β, μ, *Ave In-Degree*, *Max In-Degree*



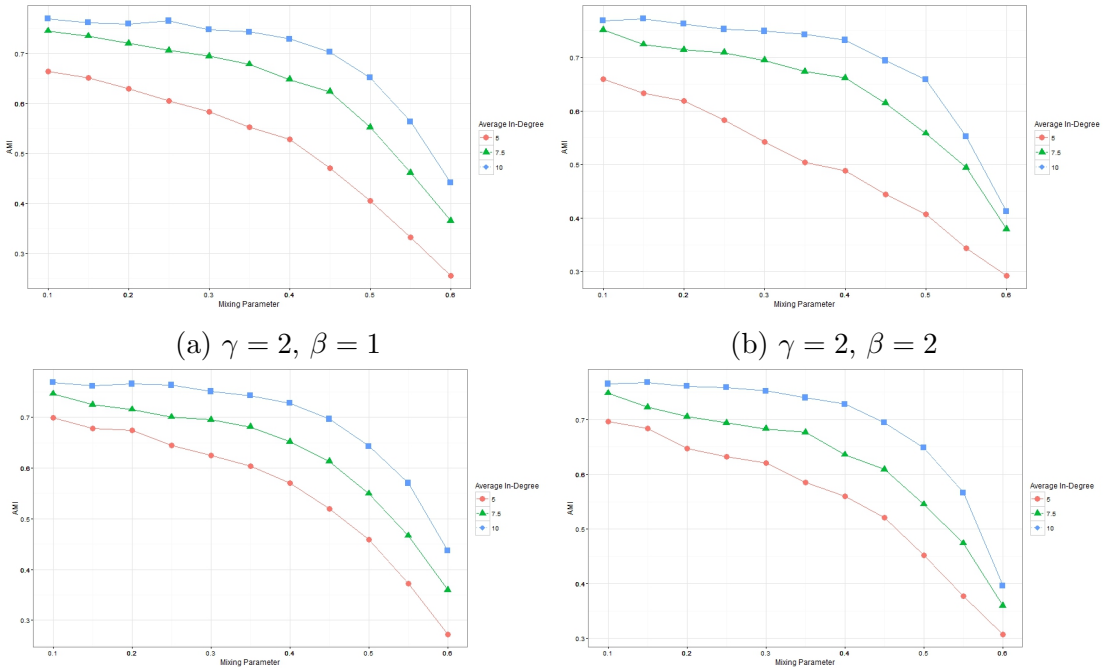
(a) $\gamma = 2$, $\beta = 1$ (b) $\gamma = 2$, $\beta = 2$

(c) $\gamma = 3$, $\beta = 1$ (d) $\gamma = 3$, $\beta = 2$

Figure 4.8: Clustering performance evaluation using the modularity objective function. Each point is the average adjusted mutual information over 100 simulated LFR benchmark networks. The lines represents average in-degree 5 (circle), 7.5 (triangle), or 10 (square).

Similar to the results seen in the undirected case of Section 2.6.2, we can observe several effects common to both the proposed likelihood objective function and modularity: (1) AMI decreases as the mixing parameter μ increases. This is intuitive since, for small μ, the clusters are more densely intra-connected and sparsely inter-connected. (2) An increase in AMI is observed as *Ave In-Degree* approaches *Max In-Degree* $= 10$, suggesting that clustering performance improves as the degree distribution becomes more homogeneous. (3) A slight increase in AMI can be observed as the parameter γ increases. (4) No significant effect on AMI due to changes in the parameter β is observed.

In Figures 4.10 and 4.11, we have averaged AMI for the 1,200 simulated LFR benchmark networks generated at each value of the mixing parameter μ, for both modularity and the proposed likelihood objective function. Unlike the previous study of directed networks in Section 2.6.2, the algorithms were *not* given the true number of clusters in advance. Rather, as described above, each



(a) $\gamma = 2$, $\beta = 1$ (b) $\gamma = 2$, $\beta = 2$

(c) $\gamma = 3$, $\beta = 1$ (d) $\gamma = 3$, $\beta = 2$

Figure 4.9: Clustering performance evaluation using the proposed objective function. Each point is the average adjusted mutual information over 100 simulated LFR benchmark networks. The lines represents average in-degree 5 (circle), 7.5 (triangle), or 10 (square).

objective function was allowed to find a solution for each value of $k \in \{10, 11, ..., 20\}$, and then the solutions corresponding to the "best" k was selected from among those solutions for each of the objective functions. The AMIs averaged in these figures were then derived by comparing the derived "best" solutions with the ground-truth clusterings output by the LFR network generator.

As differences increase between the solution clustering and the ground-truth clustering, AMI will decrease. Clusterings that are inaccurate despite a correct k value, or clusterings into an incorrect k will have a lower AMI. Therefore, these figures represent measurements of not only the quality of each objective function's solution clusterings, but also of the ability of an objective function to select the correct number of clusters via BIC.

Figure 4.10 clusters were produced using a relatively lax simulated

annealing cooling schedule[4]. 100 LFR benchmark networks were created and analyzed for each LFR parameter combination[5]. On average, the proposed likelihood objective function produced solution clusters of higher-quality than modularity over the entire range of mixing parameter μ values considered. Both objective functions' quality suffered as μ grew large. This is a natural and intuitive result as when μ exceeds 0.5, the inter-cluster edges become more dense than the intra-cluster edges. As both the likelihood and modularity objective functions seek to cluster vertices such that their intra-cluster edges are dense and inter-cluster edges are sparse, it is not surprising that cluster quality rapidly decreases as μ approaches and exceeds 0.5.

Data for Figure 4.11 were produced in the same manner as Figure 4.10, except that the cooling rate was increased to give the simulated annealing algorithm more time to find better solutions[6]. Further, only 50 LFR benchmark networks were generated for each parameter setting combination as the stronger cooling schedule significantly increased computation time required for this analysis.

In this figure, we can see that the quality of clusters produced by both objective functions increase with the strengthened cooling schedule. However, average AMI values for modularity immediately begin to decrease as μ increases and rapidly drop once μ exceeds 0.4. The proposed likelihood objective function, on the other hand, on average identifies clusters with AMI approximately equal to 0.9, a very good match, until μ reaches 0.5. Only at that point does the average AMI for likelihood begin to fall but is still above 0.8 when $\mu = 0.6$ while the average AMI for modularity has already fallen below 0.1. The stark difference in clustering performance between the two objective functions could be due to the proposed likelihood objective function's superior ability to identify clusters in the benchmark networks, even at high μ levels, relative to that of

[4]Initial temp=1, Cooling Rate: 0.9, Temp Length: 100

[5]*Max In-Degree* $= 10$, *Ave In-Degree* $\in \{5, 7.5, 10\}$, $(\gamma, \beta) \in \{(2,1), (2,2), (3,1), (3,2)\}$

[6]Initial temp=1, Cooling Rate: 0.99, Temp Length: 100

modularity, or it may be an indicator that modularity requires a significantly stronger cooling schedule than likelihood to find clusters of equivalent quality. It may be instructive to identify at which simulated annealing cooling schedules the two objective functions produce clusters of approximately equal quality, but we have not made such a comparison in this manuscript.

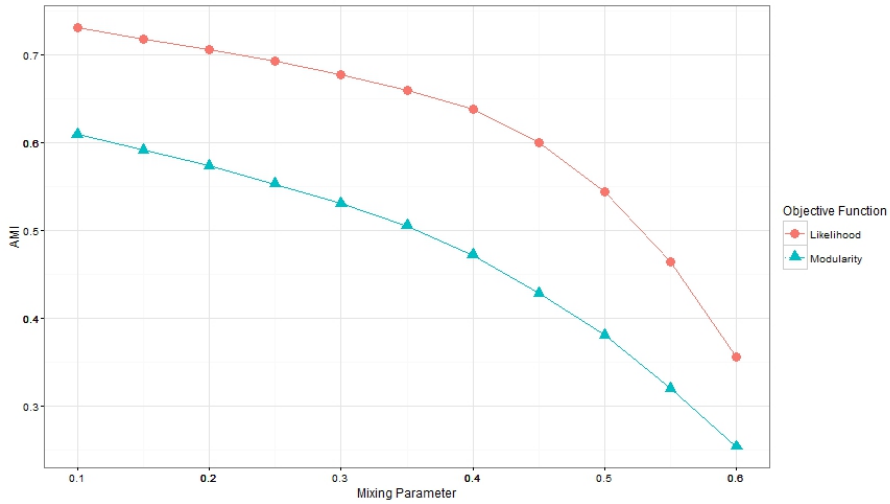


Figure 4.10: Clustering performance for proposed objective function and modularity. Each point is the average adjusted mutual information over 1200 simulated LFR benchmark networks. Likelihood is indicated by circles and modularity by triangles.

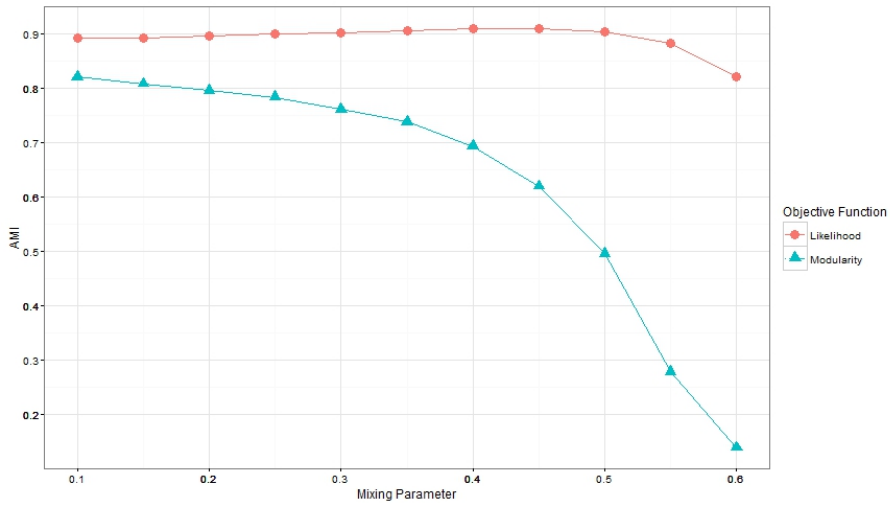


Figure 4.11: Clustering performance for proposed objective function and modularity. Each point is the average adjusted mutual information over 1200 simulated LFR benchmark networks. Likelihood is indicated by circles and modularity by triangles.

4.9 Summary and discussion

When clustering a network, whether undirected or directed, we seek to identify clusters of nodes that are densely intra-connected to each other and sparsely inter-connected to nodes of other clusters. The undirected case has a rich selection of methods available for this problem, and many of these techniques have been extended to the directed case, including the popular modularity metric.

Perry et al. (2013) proposed a likelihood objective function for use in clustering undirected networks and demonstrated that it outperformed modularity over a range of simulated networks designed to be similar to real-work networks. Further, an approximation to the distribution of the proposed likelihood ratio test statistic under the null hypothesis of a single cluster (i.e., $k = 1$) was proposed and used in an approximate test of the statistical significance of identified clusterings.

In this chapter, we have extended the likelihood objective function to directed networks and compared its performance against that of directed modularity. As in the undirected case, our simulations on simulated directed networks indicate that maximizing the likelihood objective function will generally yield superior clustering performance than that achieved by maximizing modularity. We have also extended the theorems derived in Chapter 3 to the directed network case, making likelihood maximization via heuristic algorithms such as simulated annealing more efficient. Finally, we demonstrated the use of likelihood in clustering real-world networks of Hansell (1984) and Adamic and Glance (2005), and the application of the statistical significance test, which required no modification for use with directed networks.

5 CONCLUSION

The research presented in the previous Chapters 2∼4 proposed likelihood objective functions for clustering undirected and directed networks and derived a statistical significance test for evaluating the resultant clusterings.

Theorems were derived for both undirected and directed networks to reduce computation complexity when evaluating proposed clusterings against the current clustering in an optimization context. It was further demonstrated that by iteratively updating the sufficient statistics and cluster counts required to estimate the parameters in the likelihood function, rather than naively calculating parameter estimates for each proposed clustering, the time required to find a final solution for a network with 2,000 vertices could be reduced from 20 minutes to less than 5 seconds. A network of 100,000 was clustered using the parameter update techniques in 39 seconds but did not reach a solution after running for more than 24 hours when the parameters were naively calculated.

Additionally, the efficacy of the likelihood objective function was demonstrated against modularity, a competitor objective function, on both simulated and real-world networks. The performance of likelihood was found to be frequently superior to that of modularity. Likelihood is an effective objective function for use in network clustering, has statistical properties that are both useful and familiar to most researchers, and is flexible in that a researcher can tailor the objective function to a network of interest by selecting an appropriate density for the likelihood expression.

REFERENCES

Adamic, L., Glance, N., 2005. The political blogosphere and the 2004 u.s. election: divided they blog. LinkKDD 05 Proceedings of the 3rd international workshop on Link discovery , 36–43.

Akaike, H., 1974. A new look at the staistical model identification. IEEE Transactions on Automatic Control 19 (6), 716–723.

Barabasi, A., 2001. The physics of the web. Physics World 14 (7) 33.

Basu, A., 2005. Cooperative groups, weakties, and the integration of peer friendships. Institute for Defence Studies and Analysis (IDSA) .

Bianconi, G., Pin, P., Marsili, M., 2009. Assessing the relevance of node features for network structure. Proceedings of the National Academy of Sciences of the United States of America 106 (28), 11433–11438.

Blondel, V., Guillaume, J., Lambiotte, R., Lefevre, E., 2008. Fast unfolding of communities in large networks. Journal of Statistical Mechanics 10, 10008. doi:10.1088/1742-5468/2008/10/P10008.

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., Wiener, J., 2000. Graph structure in the web. Computer Networks: The International Journal of Computer and Telecommunications Networking 33, 309–320.

Cerny, V., 1985. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. Journal of Optimization Theory and Applications 45, 41–51.

Clauset, A., Newman, M., Moore, C., 2004. Finding community structure in very large networks. Physical Review E 70 (066111), 1–6.

Danon, L., Diaz-Guilera, A., Duch, J., Arenas, A., 2005. Comparing community structure identification. Journal of Statistical Mechanics: Theory and Experiment 9 (P09008).

Duch, J., Arenas, A., 2005. Community detection in complex networks using extremal optimization. Physical Review E 72 (027104).

Dugue, N., Perez, A., 2015. Directed louvain : maximizing modularity in directed networks doi:10.13140/RG.2.1.4497.0328.

Fortunato, S., 2010. Community detection in graphs. Physics Reports 486, 75–174.

Girvan, M., Newman, M., 2002. Community structure in social and biological networks. Proceedings of the National Academy of Sciences of the United States of America 99 (12), 7821–7826.

Gleich, D., 2006. Hierarchical directed spectral graph partitioning. Tech. rep. Stanford University .

Gordon, A., 1987. A review of heirarchical classification. Journal of the Royal Statistical Society Series A 150 (2) , 119–137.

Guimera, R., Sales-Pardo, M., Amaral, L., 2004. Modularity from fluctuations in random graphs and complex networks. Physical Review E 72 (027104).

Hansell, S., 1984. Cooperative groups, weakties, and the integration of peer friendships. Social Psychology Quarterly 47, 316–328.

Hogg, R., McKean, J., Craig, A., 2005. Introduction to Mathematical Statistics. 6 ed., Pearson Prentice Hall, Upper Saddle River, New Jersey.

Huang, J., Zhu, T., 2006. Web communities identification from random walks. PKDD '06: Proceedings of the 10th European conference on Principle and Practice of Knoledge Discovery in Databases , 187–198.

Kernigan, B., Lin, S., 1970. An efficient heuristic procedure for partitioning graphs. Bell System Technical Journal 49, 291–307.

Kirkpatrick, S., Gelatt, C., Vecchi, M., 1983. Optimization by simulated annealing. Science 4598 (220), 671–680.

Ku cukpetek, S., Polat, F., Oguztuzun, H., 2005. Multilevel graph partitioning: an evolutionary approach. The Journal of the Operational Research Society 56 (5), 549–562.

Lancichinetti, A., Fortunato, S., 2009. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. Physical Review E 80 (016118).

Lancichinetti, A., Fortunato, S., Radicchi, F., 2008. Benchmark graphs for testing community detection algorithms. Physical Review E 78 (046110). doi:`10.1103/PhysRevE.78.046110`.

Lancichinetti, A., Radicchi, F., Ramasco, J., 2010. Statistical significance of communities in networks. Physical Review E 81 (046110).

Lancichinetti, A., Radicchi, F., Ramasco, J., Fortunato, A., 2011. Finding statistically significant communities in networks. PLoS ONE 6 (e18961).

Leicht, E., Newman, M., 2008. Community structure in directed networks. Phys. Rev. Lett. 100 (118703).

von Luxburg, U., 2007. A tutorial on spectral clustering. Statistics and Computing 17 (4).

Malliaros, F., Vazirgiannis, M., 2013. Clustering and community detection in directed networks: A survey. Physics Reports 533, 95–142.

MapleSoft, 2011. Maple.

Matsumoto, M., Nishimura, T., 1998. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation 8, 3–30. doi:10.1145/272991.272995.

Newman, M., 2004. Fast algorithm for detecting community structure in networks. Physical Review E 69 (066133).

Newman, M., 2006. Modularity and community structure in networks. Proceedings of the National Academy of Sciences of the United States of America 103, 8577–582. doi:10.1073/pnas.0601602103.

Newman, M., 2010. Networks, An Introduction. 1 ed., Oxford University press, Great Clarendon Street, Oxford OX2 6DP.

Newman, M., Girvan, M., 2004. Finding and evaluating community structure in networks. Physical Review E 69.

Newman, M.E.J., Leicht, E.A., 2007. Mixture models and exploratory analysis in networks. Proceedings of the National Academy of Sciences of the United States of America 104, 9564–9569.

Perry, M., Michaelson, G., Ballard, A., 2013. On the statistical detection of clusters in undirected networks. Computational Statistics and Data Analysis 68, 170–189. doi:10.1016/j.csda.2013.06.019.

Press, W., Teukolsky, S., Vetterling, W., Flannery, B., 2007. Numerical Recipes: The Art of Scientific Computing. 3 ed., Cambridge University Press.

Reichardt, J., Bornholdt, S., 2006. Statistical mechanics of community detection. Physical Review E 74 (016110).

Rohe, K., Qin, T., Yu, B., 2006. Co-clustering for directed graphs; the stochastic co-blockmodel and a spectral algorithm. Tech. rep., ArXiv .

Sageman, M., 2004. Understanding Terror Networks. University of Pennsylvania Press.

Schwarz, G., 1978. Estimating the dimension of a model. The Annals of Statistics 6, 461–464.

Snijders, T., Nowicki, K., 1997. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. Journal of Classification 14, 75–100.

Steinley, D., 2006. K-means clustering: a half-century synthesis. British Journal of Mathematical and Statistical Psychology 59, 1–34.

Vinh, N., Epps, J., 2010. Information theoretic measures for clustering comparison: variants, properties, normalization and correction for chance. Journal of Machine Learning Research 11, 2837–2854.

Wang, Y., Wong, G., 1987. Stochastic blockmodels for directed graphs. Journal of the American Statistical Association 82, 8–19.

Zachary, W., 1977. An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33, 452–473.

Zanghi, H., Ambroise, C., Miele, V., 2007. Fast on-line graph clustering via erosrenyi mixture. Pattern Recognition 41, 3591–3599.

Zhang, B., Liu, R., Massey, D., Zhang, L., 2005. Collecting the internet as-level topology. SIGCOMM Comput. Commun. Rev. 35 (1).

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

APPENDIX

k	α			
	0.1	0.05	0.01	0.001
$n = 100$				
2	141.74	143.18	146.44	151.06
5	331.59	333.04	336.34	340.99
8	427.82	429.28	432.58	437.26
10	472.20	473.66	476.98	481.67
15	548.24	549.71	553.05	557.77
$n = 200$				
2	280.37	281.81	285.07	289.69
5	655.52	656.96	660.24	664.87
8	847.79	849.25	852.53	857.18
10	938.19	939.65	942.93	947.59
15	1098.89	1100.35	1103.65	1108.32
$n = 300$				
2	419.00	420.44	423.70	428.32
5	655.52	980.05	983.32	987.95
8	847.79	1267.53	1270.81	1275.44
10	938.19	1403.37	1406.64	1411.28
15	1098.89	1647.20	1650.48	1655.13
$n = 400$				
2	557.63	559.07	562.33	566.95
5	1301.35	1302.79	1306.06	1310.68
8	1683.68	1685.13	1688.40	1693.03
10	1864.71	1866.16	1869.43	1874.07
15	2191.07	2192.52	2195.80	2200.44
$n = 500$				
2	696.26	697.70	700.96	705.58
5	1623.90	1625.34	1628.60	1633.23
8	2100.90	2102.34	2105.61	2110.24
10	2327.00	2328.45	2331.72	2336.35
15	2735.57	2737.01	2740.29	2744.92

Table 5.1: Table of critical values for test of significance of k clusters

Proof of Theorem 3.4.1.

Let $\hat{\theta}_s^1 = \hat{\theta}_s^0, s \notin \{\ell, j\}$. Then, $f_s(y_s|\hat{\theta}_s^1, \mathbf{z}_{(k)}^1) = f_s(y_s|\hat{\theta}_s^0, \mathbf{z}_{(k)}^0)$. Finally, for space purposes, let $\Lambda_t^0 = y_t|\hat{\theta}_t^0, \mathbf{z}_{(k)}^0$ and $\Lambda_t^1 = y_t|\hat{\theta}_t^1, \mathbf{z}_{(k)}^1$, $t \in \{1, 2, ...k, b\}$. Then, $f_s(\Lambda_s^1) = f_s(\Lambda_s^0), s \notin \{\ell, j\}$ and,

$$
\begin{aligned}
\Delta \ln L &= ln[\{\prod_{s \neq \{\ell,j\}} f_s(\Lambda_s^1)\} f_\ell(\Lambda_\ell^1) f_j(\Lambda_j^1) f_b(\Lambda_b^1)] \\
&- ln[\{\prod_{s \neq \{\ell,j\}} f_s(\Lambda_s^0)\} f_\ell(\Lambda_\ell^0) f_j(\Lambda_j^0) f_b(\Lambda_b^0)] \\
&= ln\left[\frac{\{\prod_{s \neq \{\ell,j\}} f_s(\Lambda_s^1)\} f_\ell(\Lambda_\ell^1) f_j(\Lambda_j^1) f_b(\Lambda_b^1)}{\{\prod_{s \neq \{\ell,j\}} f_s(\Lambda_s^0)\} f_\ell(\Lambda_\ell^0) f_j(\Lambda_j^0) f_b(\Lambda_b^0)}\right] \\
&= ln\left[\frac{\{\prod_{s \neq \{\ell,j\}} f_s(\Lambda_s^0)\} f_\ell(\Lambda_\ell^1) f_j(\Lambda_j^1) f_b(\Lambda_b^1)}{\{\prod_{s \neq \{\ell,j\}} f_s(\Lambda_s^0)\} f_\ell(\Lambda_\ell^0) f_j(\Lambda_j^0) f_b(\Lambda_b^0)}\right] \\
&= ln\left[\frac{f_\ell(\Lambda_\ell^1) f_j(\Lambda_j^1) f_b(\Lambda_b^1)}{f_\ell(\Lambda_\ell^0) f_j(\Lambda_j^0) f_b(\Lambda_b^0)}\right]
\end{aligned}
$$

□

Proof of Theorem 3.4.2.

Assuming $s, t \notin \{\ell, j\}$,

1. $obs_{s,t}^1 = {\omega_s^1}^T \mathbf{A} \omega_t^1 = {\omega_s^0}^T \mathbf{A} \omega_t^0 = obs_{s,t}^0$

2. $obs_{s,s}^1 = \frac{1}{2} {\omega_s^1}^T \mathbf{A} \omega_s^1 = \frac{1}{2} {\omega_s^0}^T \mathbf{A} \omega_s^0 = obs_{s,s}^0$

3. $$
\begin{aligned}
obs_{\ell,s}^1 - obs_{\ell,s}^0 &= {\omega_\ell^1}^T \mathbf{A} {\omega_s}^1 - {\omega_\ell^0}^T \mathbf{A} \omega_s^0 \\
&= {\omega_\ell^1}^T \mathbf{A} {\omega_s}^0 - {\omega_\ell^0}^T \mathbf{A} {\omega_s}^0 \\
&= -[\omega_\ell^0 - \omega_\ell^1]^T \mathbf{A} {\omega_s}^0 \\
&= -[\mathbf{1}_{l^*}]^T \mathbf{A} \omega_s^0 = -Z_s^0
\end{aligned}$$

4. $$
\begin{aligned}
obs_{j,s}^1 - obs_{j,s}^0 &= {\omega_j^1}^T \mathbf{A} \omega_s^1 - {\omega_j^0}^T \mathbf{A} \omega_s^0 \\
&= {\omega_j^1}^T \mathbf{A} \omega_s^0 - {\omega_j^0}^T \mathbf{A} \omega_s^0 \\
&= [\omega_j^1 - \omega_j^0]^T \mathbf{A} {\omega_s}^0 \\
&= [\mathbf{1}_{l^*}]^T \mathbf{A} \omega_s^0 = Z_s^0
\end{aligned}$$

5. $obs^1_{\ell,j} = {\omega^1_\ell}^T \mathbf{A} \omega^1_j = [\omega^0_\ell - \mathbf{1}_{l^*}]^T \mathbf{A} [\omega^0_j + \mathbf{1}_{l^*}]$

$$= {\omega^0_\ell}^T \mathbf{A} {\omega_j}^0 - [\mathbf{1}_{l^*}]^T \mathbf{A} \omega^0_j + {\omega^0_\ell}^T \mathbf{A} \mathbf{1}_{l^*}$$

$$= obs^0_{\ell,j} - Z^0_j + Z^0_\ell$$

6. $obs^1_{\ell,\ell} = \frac{1}{2} {\omega^1_\ell}^T \mathbf{A} \omega^1_\ell = \frac{1}{2} [\omega^0_\ell - \mathbf{1}_{l^*}]^T \mathbf{A} [\omega^0_\ell - \mathbf{1}_{l^*}]$

$$= \frac{1}{2} [{\omega^0_\ell}^T \mathbf{A} \omega^0_\ell - [\mathbf{1}_{l^*}]^T \mathbf{A} \omega^0_\ell - [\mathbf{1}_{l^*}]^T \mathbf{A} \omega^0_\ell]$$

$$= \frac{1}{2} [2obs^0_{\ell,\ell} - 2[\mathbf{1}_{l^*}]^T A \omega^0_\ell] = obs^0_{\ell,\ell} - Z^0_\ell$$

7. $obs^1_{j,j} = \frac{1}{2} {\omega^1_j}^T \mathbf{A} \omega^1_j = \frac{1}{2} [\omega^0_j + \mathbf{1}_{l^*}]^T \mathbf{A} [\omega^0_j + \mathbf{1}_{l^*}]$

$$= \frac{1}{2} [{\omega^0_j}^T \mathbf{A} \omega^0_j + [\mathbf{1}_{l^*}]^T \mathbf{A} \omega^0_j + [\mathbf{1}_{l^*}]^T \mathbf{A} \omega^0_j]$$

$$= \frac{1}{2} [2obs^0_{j,j} + 2[\mathbf{1}_{l^*}]^T \mathbf{A} \omega^0_j] = obs^0_{j,j} + Z^0_j$$

□

Proof of Theorem 3.4.3.

$$\begin{aligned}
obs^1_b &= \sum obs^1_{s,t} + \sum obs^1_{\ell,t} + \sum obs^1_{j,t} + obs^1_{\ell,j}, \\
&= \sum obs^0_{s,t} + \sum (obs^0_{\ell,t} - Z^0_t) \\
&+ \sum (obs^0_{j,t} + Z^0_t) + (obs^0_{\ell,j} + Z^0_\ell - Z^0_j), \text{ by Theorem 3.4.2} \\
&= \sum obs^0_{s,t} + \sum obs^0_{\ell,t} + \sum obs^0_{j,t} + (obs^0_{\ell,j} + Z^0_\ell - Z^0_j) \\
&= (\sum obs^0_{s,t} + \sum obs^0_{\ell,t} + \sum obs^0_{j,t} + obs^0_{\ell,j}) + Z^0_\ell - Z^0_j \\
&= obs^0_b + Z^0_\ell - Z^0_j, s, t \notin \{\ell, j\},\ s \neq t \quad \square
\end{aligned}$$

□

Proof of Theorem 3.4.4.

Assuming $s, t \notin \{\ell, j\}$,

1. $pos^1_{s,t} = n^0_s n^0_t = pos^0_{s,t}$

2. $pos^1_{s,s} = \frac{n^0_s(n^0_s - 1)}{2} = pos^0_{s,s}$

3. $pos^1_{\ell,s} = n^0_s(n^0_\ell - 1)$

4. $pos^1_{j,s} = n^0_s(n^0_j + 1)$

5. $pos^1_{\ell,j} = (n^0_\ell - 1)(n^0_j + 1)$

6. $pos^1_{\ell,\ell} = \dfrac{(n^0_\ell - 1)((n^0_\ell - 1) - 1)}{2} = \dfrac{(n^0_\ell - 1)(n^0_\ell - 2)}{2}$

7. $pos^1_{j,j} = \dfrac{(n^0_j + 1)((n^0_j + 1) - 1)}{2} = \dfrac{(n^0_j + 1)n^0_j}{2}$

□

Proof of Theorem 3.4.5.

$$
\begin{aligned}
pos^1_b &= \sum pos^1_{s,t} + \sum pos^1_{\ell,t} + \sum pos^1_{j,t} + \sum pos^1_{\ell,j} \\
&= \sum pos^0_{s,t} + \sum n^0_t(n^0_\ell - 1) + \sum n^0_t(n^0_j + 1) \\
&+ (n^0_\ell - 1)(n^0_j + 1), \text{ by Theorem 3.4.4} \\
&= \sum pos^0_{s,t} + \sum n^0_t(n^0_\ell) + \sum n^0_t(n^0_j) + ((n^0_\ell)(n^0_j) \\
&+ n^0_\ell - n^0_j - 1) \\
&= (\sum pos^0_{s,t} + \sum pos^0_{\ell,t} + \sum pos^0_{j,t} + pos^0_{\ell,j}) \\
&+ n^0_\ell - n^0_j - 1 \\
&= pos^0_b + n^0_\ell - n^0_j - 1,\ s,t \notin \{\ell, j\},\ s \neq t
\end{aligned}
$$

□

Proof of Theorem 4.5.1.

The proof for this theorem is identical to Theorem 3.4.1. □

Proof of Theorem 4.5.2.

Assuming $s, t \notin \{\ell, j\}$,

1. $obs^1_{s,t} = {\omega^1_s}^T \mathbf{A} \omega^1_t = {\omega^0_s}^T \mathbf{A} \omega^0_t = obs^0_{s,t}$

2. $obs^1_{t,s} = obs^0_{t,s}$, by 1 above

3. $obs^1_{s,s} = {\omega^1_s}^T \mathbf{A} \omega^1_s = {\omega^0_s}^T \mathbf{A} \omega^0_s = obs^0_{s,s}$

4. $$
\begin{aligned}
obs^1_{\ell,s} - obs^0_{\ell,s} &= {\omega^1_\ell}^T \mathbf{A} {\omega_s}^1 - {\omega^0_\ell}^T \mathbf{A} \omega^0_s \\
&= {\omega^1_\ell}^T \mathbf{A} {\omega_s}^0 - {\omega^0_\ell}^T \mathbf{A} {\omega_s}^0 \\
&= -[\omega^0_\ell - \omega^1_\ell]^T \mathbf{A} {\omega_s}^0 \\
&= -[\mathbf{1}_{l^*}]^T \mathbf{A} \omega^0_s = -Z^0_{\text{to s}}
\end{aligned}
$$

5. $obs^1_{s,\ell} - obs^0_{s,\ell} = {\omega_s^1}^T \mathbf{A} {\omega_\ell}^1 - {\omega_s^0}^T \mathbf{A} \omega_\ell^0$

$$= {\omega_s^0}^T \mathbf{A} {\omega_\ell}^1 - {\omega_s^0}^T \mathbf{A} {\omega_\ell}^0$$

$$= -{\omega_s^0}^T \mathbf{A} [\omega_\ell^0 - \omega_\ell^1]$$

$$= -{\omega_s^0}^T \mathbf{A} [\mathbf{1}_{l^*}] = -Z^0_{\text{from s}}$$

6. $obs^1_{j,s} - obs^0_{j,s} = {\omega_j^1}^T \mathbf{A} \omega_s^1 - {\omega_j^0}^T \mathbf{A} \omega_s^0$

$$= {\omega_j^1}^T \mathbf{A} \omega_s^0 - {\omega_j^0}^T \mathbf{A} \omega_s^0$$

$$= [\omega_j^1 - \omega_j^0]^T \mathbf{A} {\omega_s}^0$$

$$= [\mathbf{1}_{l^*}]^T \mathbf{A} \omega_s^0 = Z^0_{\text{to s}}$$

7. $obs^1_{s,j} - obs^0_{s,j} = {\omega_s^1}^T \mathbf{A} \omega_j^1 - {\omega_s^0}^T \mathbf{A} \omega_j^0$

$$= {\omega_s^0}^T \mathbf{A} \omega_j^1 - {\omega_s^0}^T \mathbf{A} \omega_j^0$$

$$= {\omega_s^0}^T \mathbf{A} [\omega_j^1 - \omega_j^0]$$

$$= {\omega_s^0}^T \mathbf{A} [\mathbf{1}_{l^*}] = Z^0_{\text{from s}}$$

8. $obs^1_{\ell,j} = {\omega_\ell^1}^T \mathbf{A} \omega_j^1 = [\omega_\ell^0 - \mathbf{1}_{l^*}]^T \mathbf{A} [\omega_j^0 + \mathbf{1}_{l^*}]$

$$= {\omega_\ell^0}^T \mathbf{A} {\omega_j}^0 - [\mathbf{1}_{l^*}]^T \mathbf{A} \omega_j^0 + {\omega_\ell^0}^T \mathbf{A} \mathbf{1}_{l^*}$$

$$= obs^0_{\ell,j} - Z^0_{\text{to j}} + Z^0_{\text{from } \ell}$$

9. $obs^1_{j,\ell} = {\omega_j^1}^T \mathbf{A} \omega_l^1 = [\omega_j^0 + \mathbf{1}_{l^*}]^T \mathbf{A} [\omega_\ell^0 - \mathbf{1}_{l^*}]$

$$= {\omega_j^0}^T \mathbf{A} {\omega_\ell}^0 + [\mathbf{1}_{l^*}]^T \mathbf{A} \omega_\ell^0 - {\omega_j^0}^T \mathbf{A} \mathbf{1}_{l^*}$$

$$= obs^0_{j,\ell} + Z^0_{\text{to } \ell} - Z^0_{\text{from j}}$$

10. $obs^1_{\ell,\ell} = {\omega_\ell^1}^T \mathbf{A} \omega_\ell^1 = [\omega_\ell^0 - \mathbf{1}_{l^*}]^T \mathbf{A} [\omega_\ell^0 - \mathbf{1}_{l^*}]$

$$= {\omega_\ell^0}^T \mathbf{A} \omega_\ell^0 - [\mathbf{1}_{l^*}]^T \mathbf{A} \omega_\ell^0 - {\omega_\ell^0}^T \mathbf{A} [\mathbf{1}_{l^*}]$$

$$= obs^0_{\ell,\ell} - Z^0_{\text{to } \ell} - Z_{\text{from } \ell}$$

11. $obs^1_{j,j} = {\omega_j^1}^T \mathbf{A} \omega_j^1 = [\omega_j^0 + \mathbf{1}_{l^*}]^T \mathbf{A} [\omega_j^0 + \mathbf{1}_{l^*}]$

$$= {\omega_j^0}^T \mathbf{A} \omega_j^0 + [\mathbf{1}_{l^*}]^T \mathbf{A} \omega_j^0 + {\omega_j^0}^T \mathbf{A} [\mathbf{1}_{l^*}]$$

$$= obs^0_{j,j} + Z^0_{\text{to j}} + Z^0_{\text{from j}}$$

□

Proof of Theorem 4.5.3.

$$
\begin{aligned}
obs_b^1 &= \sum obs_{s,t}^1 + \sum obs_{t,s}^1 + \sum obs_{\ell,t}^1 + \sum obs_{t,\ell}^1 \\
&+ \sum obs_{j,t}^1 + \sum obs_{t,j}^1 + obs_{\ell,j}^1 + obs_{j,\ell}^1 \\
&= \sum obs_{s,t}^0 + \sum obs_{t,s}^0 + \sum (obs_{\ell,t}^0 - Z_{\text{to t}}^0) + \sum (obs_{t,\ell}^0 - Z_{\text{from t}}^0) \\
&+ \sum (obs_{j,t}^0 + Z_{\text{to t}}^0) + \sum (obs_{t,j}^0 + Z_{\text{from t}}^0) \\
&+ (obs_{\ell,j}^0 - Z_{\text{to j}}^0 + Z_{\text{from }\ell}^0) + (obs_{j,\ell}^0 + Z_{\text{to }\ell}^0 - Z_{\text{from j}}^0), \text{ by Theorem 4.5.2} \\
&= (\sum obs_{s,t}^0 + \sum obs_{t,s}^0 + \sum obs_{\ell,t}^0 + \sum obs_{t,\ell}^0 + \sum obs_{j,t}^0 + \sum obs_{t,j}^0 \\
&+ obs_{\ell,j}^0 + obs_{j,\ell}^0) - Z_{\text{to j}}^0 - Z_{\text{from j}}^0 + Z_{\text{to }\ell}^0 + Z_{\text{from }\ell}^0 \\
&= obs_b^0 - Z_{\text{to j}}^0 - Z_{\text{from j}}^0 + Z_{\text{to }\ell}^0 + Z_{\text{from }\ell}^0, s, t \notin \{\ell, j\},\ s \neq t \quad \square
\end{aligned}
$$

$\square$

Proof of Theorem 4.5.4.
Assuming $s, t \notin \{\ell, j\}$,

1. $pos_{s,t}^1 = n_s^0 n_t^0 = pos_{s,t}^0$
2. $pos_{t,s}^1 = n_t^0 n_s^0 = pos_{t,s}^0$
3. $pos_{s,s}^1 = 2[\frac{n_s^0(n_s^0 - 1)}{2}] = n_s^0(n_s^0 - 1) = pos_{s,s}^0$
4. $pos_{\ell,s}^1 = (n_\ell^0 - 1)n_s^0$
5. $pos_{s,\ell}^1 = n_s^0(n_\ell^0 - 1)$
6. $pos_{j,s}^1 = (n_j^0 + 1)n_s^0$
7. $pos_{s,j}^1 = n_s^0(n_j^0 + 1)$
8. $pos_{\ell,j}^1 = (n_\ell^0 - 1)(n_j^0 + 1)$
9. $pos_{j,\ell}^1 = (n_j^0 + 1)(n_\ell^0 - 1)$
10. $pos_{\ell,\ell}^1 = 2[\frac{(n_\ell^0 - 1)((n_\ell^0 - 1) - 1)}{2}] = (n_\ell^0 - 1)(n_\ell^0 - 2)$
11. $pos_{j,j}^1 = 2[\frac{(n_j^0 + 1)((n_j^0 + 1) - 1)}{2}] = (n_j^0 + 1)n_j^0$

$\square$