ON THE IDENTIFICATION OF STATISTICALLY SIGNIFICANT NETWORK TOPOLOGY

by

GREGORY VINCENT MICHAELSON

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

2010

ABSTRACT

Determining the structure of large and complex networks is a problem that has stirred great interest in many fields including mathematics, computer science, sociology, biomedical research, and epidemiology. Despite this high level of interest, though, there still exists no procedure for formal hypothesis testing to measure the significance of detected community structure in an observed network. First, this work proposes three, more general alternatives to modularity, the most common measure of community structure, which allow for the detection of more general structure in networks. An approach based upon the likelihood ratio test is shown not only to be as effective as modularity in detecting modular structure but also able to detect a wide variety of other network topologies. Second, this work proposes a general and novel test, the Likelihood Ratio Cluster (LRC) test, for assessing the statistical significance of the output of clustering algorithms. This technique is demonstrated by applying it to the sample partitions generated by both network and conventional clustering algorithms. Finally, a method for evaluating the capability of heuristic clustering techniques to detect the optimal sample partition is developed. This technique is used to evaluate several common community detection algorithms. Surprisingly, the most popular community detection algorithm is found to be largely ineffective at detecting the optimal partition of a random network. Also surprisingly, Clauset's fast algorithm (Clauset et al., 2004), which is commonly thought to be fast but inaccurate, is found to be the most effective of the algorithms examined at detecting the optimal partition in random networks.

ACKNOWLEDGEMENTS

I humbly acknowledge the support provided by my dissertation committee, my wife and family, and Calvary Baptist Church. Without these, this and many other goals could not have been reached.

CONTENTS

FIGURES

TABLES

CHAPTER 1

INTRODUCTION

Statistics is the science of transforming data into useful information. Statistics allows researchers and practitioners to describe a sample of data, to infer characteristics about the population from which that sample was taken, to identify and measure the strength of the relationships between variables, to mine information from massive data sets, and many other applications.

Commonly, data is displayed, in raw form, as a table of values in which the columns represent variables, and the rows represent observations. Modern data sets often have thousands of variables and millions of observations. Techniques have been developed to analyze such data, and increasingly powerful computers and increasingly sophisticated methods have made it possible to analyze even these very large data sets. Another data structure, network data, has generated intense interest in a wide variety of disciplines in recent years, though the study and analysis of this type of data is not new. This work is primarily concerned with the treatment of such data.

1.1 A Brief Introduction to Network Data

Network data is not to be described with a set of variables that are used to describe particular observations in a spreadsheet. Rather, network data consists

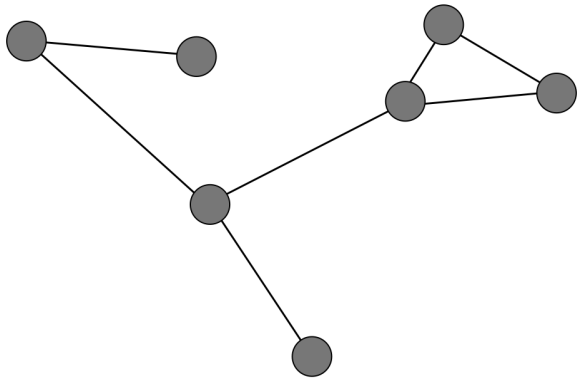

Figure 1.1. A sample graph. This network has 7 nodes and 7 edges.

of a set of nodes and the set of edges that connect those nodes. A simple example might be the e-mail correspondence between pairs of employees. Each employee, in this case, is a node, and a connection — also called an edge or a tie — between two nodes exists if those two employees have exchanged e-mails.

Network data is often displayed as a graph in which nodes are represented by points and the connections between those nodes are represented by lines. Figure 1.1 displays one such graph. Additionally, a network consisting of n nodes can also be represented as an $n \times n$ matrix, $\mathbf{X}$, in which the value of element x_{ij} represents the relationship going from node i to node j.

Stated symbolically, a graph $\mathcal{G} = (V, E)$ in which each edge $e_{ij} \in E$ denotes the connection or relationship from node v_i to node v_j. Networks can be further categorized based on the types of edges that exist between nodes. An *undirected* network is a network in which $e_{ij} = e_{ji}$; i.e., a relationship from node i to node j implies an equal relationship from node j to node i. In contrast, a *directed* network does not have this restriction. The above example of e-mail correspondence could be considered a directed network in which a link connecting one employee to

another employee implies that the former e-mailed the latter. Such a link does not imply that the latter has e-mailed the former.

The values taken by the elements of E define another category. 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.

Let a partition of a network, $\mathbf{s}$, be an $n \times 1$ vector of discrete values such that $s_i \in \{1, 2, \ldots, k\}$, $i = 1, 2, \ldots, n$, where n is the number of nodes in the network and k is the number of subgroups[1] into which the network is divided, and s_i denotes the group to which node v_i is assigned; that is, each node must belong to one and only one subgroup.[2]

1.2 The Problem of Determining Network Topology

One of the oldest problems in network analysis is the problem of finding meaningful groups of nodes within the global network.[3] The problem of identifying these groups is known by many names including graph clustering, blockmodeling, and, more recently, community detection. Since each of these terms carries with it some of the specificities associated with the discipline under which the term was

[1] While the term *subgroup* has a precise mathematical definition, this meaning is not intended. Unless otherwise noted, the term will be used to refer, generally, to any smaller subset of nodes within a network.

[2] There are methods, called fuzzy clustering methods, that allow a single node to be a member of more than one subgroup. These methods are largely outside the scope of this work, but represent an important area of continuing research.

[3] Define a *global* network to be the set of all nodes and links connecting those nodes. This is in contrast to a *local* network, which focuses on a smaller set of nodes in the same neighborhood, or an *ego* network, which focuses on a single node and all the ties connecting to that node.

developed,[4] this work will use the more general *network topology* to describe the structural features that characterize a network. While this term generally includes other features of a network — e.g., its degree distribution or evolution over time — the use of the term within this work shall include and place emphasis on the existence and description of any smaller, relationally distinct groupings of nodes within the global network.

The widespread interest in this problem has led to many different approaches. These approaches will be discussed in detail in Chapter 2. Despite this wide variety of approaches, however, the essential problem persists in each setting: How can the topology of a network or graph be searched for, detected, and measured? Within many networks, smaller "sub-networks" of nodes can often be identified. These sub-networks relate amongst themselves and with others in the network differently than members of other sub-networks. For example, in a high school, one expects to see many cliques that are tightly intraconnected, but sparsely interconnected; e.g., the football team might not hang out much with the math club. The cheerleading squad might avoid the band geeks, and so on.

The problem of detecting these sub-networks amounts to placing each node into the right group. Many different definitions of "the right group" have been suggested and will be discussed in Chapter 2, but given some definition of optimality, the combinatorial complexity of the problem is the chief obstacle to solving it. For a network of, say, 20 nodes, there are $542,284$ ways to partition the network into two groups. Given some way to measure the quality of a partition (and many *quality functions* have been suggested), modern computers could examine each of these and determine the optimal partition. There are, however, approximately

[4]For instance, *graph clustering* generally refers to a more restrictive mathematical approach to the problem, while *community detection* generally refers to the problem of finding densely intraconnected yet sparsely interconnected groups of nodes in a network.

$749,206,090,500$ ways to partition that network into five groups.[5] This is for a network of only 20 nodes. Add to that the additional difficulty added when the appropriate number of groups is unknown, the brute force approach quickly become untenable. Despite this complexity, however, interest in this problem has remained strong. Perhaps this is due to the diversity and importance of the problems that can be framed in this way.

For example, the journal *Parallel Computing* recently devoted an entire issue (Volume 26, Issue 12) to the problem of graph partitioning as it relates to parallel computing. Computational tasks can be described as graphs, in which computations are represented by vertices, and the links between these vertices represent data dependencies. Using many processors to accomplish a complex computational task requires coordination between the processors. By minimizing the communication between processors, the computations can be accomplished more efficiently; that is, by identifying densely intraconnected and sparsely interconnected subgroups of calculations (in this case, of approximately equal size), the optimal assignment of tasks to each processor can be determined.

Epidemiologists also find this problem meaningful as it relates to the spread of disease; e.g., Eubank (2005); Ganesh et al. (2005); Zhang et al. (2008) to name a few. Under some circumstances, diseases appear to spread much faster within dense clusters, but slower between clusters. By considering incubation periods, virulence, and other factors related to the disease in question as well as the topology of the host population, researchers are able to model disease spread. Similarly, such studies can identify subgroups at higher risk of infection during an outbreak.

There are many other relevant applications that could be mentioned, ranging

[5]The number of ways to divide a set into k subsets is known as Stirling's number of the second kind. See Chapter 5 for further details. Interested readers are additionally referred to Abramowitz and Stegun (1972) or other mathematical text.

from identifying the proteins responsible for and involved in cancer metastasis (Jonsson et al., 2006; Ruan et al., 2006) to detecting price fixing amongst stock traders (Palshikar and Apte, 2008). These examples could be greatly multiplied in order to demonstrate the relevance of and to motivate continued research in this area.

1.3 Organization of this Work

As a starting point, this work begins with the assumption that there exists randomness in observed networks. That randomness may result from missing nodes, incorrectly measured or missing links, networks sampled only over a particular period of time, errors in reporting and data collection, or a plethora of other sources. Additionally, in line with current thinking (Barabasi and Albert, 1999), in social, biological, ecological, and similar networks, the influences that govern the formation and evolution of those networks over time, as well as the effects of randomness, result in the peculiarities and characteristics of those networks. Hence, any particular realization of a network may be the result of some stochastic process (which may be largely unknown) that results in the observed network under consideration. While there are some applications that might exclude the effect of randomness — e.g., applications to parallel computing — this work will focus on those networks that may appropriately be considered random.

In order to overcome the combinatorial complexity associated with this problem, many heuristic approaches have been developed to detect particular types of network topology. Chapter 2 reviews relevant literature for the topics discussed in this work. Section 2.1 briefly describes relevant literature in regards to general clustering techniques. Section 2.2 discusses the different approaches to determin-

ing network topology in the various fields where it has been addressed. Section 2.3 describes the various degree distributions by which networks are often categorized. Section 2.4 discusses the various popular definitions of optimality that have been proposed for partitioning algorithms. Finally, because this work in part seeks to develop a methodology for identifying *statistically significant* community structure, Section 2.5 reviews existing techniques to apply statistical inference to this problem.

Chapter 3 describes the gaps in existing research and introduces the contribution this work makes. Chapter 4 proposes three, more general alternatives to modularity, the maximization of which allow for more general sorts of network topology to be detected. Chapter 5 describes a novel test for the statistical significance of the output of clustering algorithms. Finally, Chapter 6 uses the test developed in Chapter 5 to evaluate the effectiveness of four common network clustering algorithms.

CHAPTER 2

LITERATURE REVIEW

Several areas of research are relevant to this work. While this work will largely assume that a particular optimal (or approximate optimal) solution to the community detection problem has been determined, Section 2.1 will give a brief overview of important findings in general clustering techniques, and Section 2.2 will give a brief overview of some of the most common types of search techniques used for determining network topology.

Any of the search techniques described in Section 2.2 must include a definition of optimality. Such definition is often referred to as the *quality function*. Section 2.4 will describe various popular quality functions and models, from the more general to the more restricted cases. Also in this chapter, brief mention will be made of a few miscellaneous approaches, such as multi-modal clustering and detecting overlapping communities in which nodes may be members of more than one cluster. Though these miscellaneous approaches will not play a direct part in this work, they do represent one direction of continued research in this area.

Section 2.3 will identify relevant research related to the degree distribution — i.e., the distribution of the number of links connected to each node in a network — of various types of networks, including Erdős Rényi (ER) random graphs, scale-free networks, and small-world networks. Degree distribution is important in identifying the null model used in calculating the modularity measure.

Finally, Section 2.5 will describe efforts that have been made to apply principles of statistical inference to the problem of community detection. In particular, research has addressed, in varying degrees of thoroughness, the question of correct group membership as well as determining the statistical significance of a given partition.

2.1 Conventional Clustering

Clustering is a well-known unsupervised learning algorithm — i.e., a model is not fitted to a training set of data containing group labels — that groups together various observational units based on the similarity (or dissimilarity) that exists between them. Good clustering algorithms, then, are those that group together nodes of high similarity and separate nodes of high dissimilarity. One might use clustering algorithms to allow a computer to identify objects in a digital image, as an example.

Different approaches have been developed to address this problem, many of which are very well developed. Hierarchical clustering seeks to identify nested clusters in a data set. Either agglomerative or divisive, the algorithm either combines or separates observational units in order to produce the clusters. The user is left to determine the appropriate number of clusters for the particular data set.

Another approach that requires the user to set the number of groups is k-means clustering. In this approach, the user selects the number of groups and then randomly assigns each observational unit to one of those groups. The center of each of the groups is calculated, and each observational unit is reassigned to the nearest cluster. The centers of these new groups are recalculated and the observational units are again reassigned to the closest group. The process continues

until group membership stabilizes.

Unlike the previous examples, spectral clustering does not require the user to specify the number of groups. This approach requires the calculation of a matrix to describe the dissimilarity between each pair of observational units; i.e., the dissimilarity matrix. The eigenvectors and eigenvalues of this matrix — or one closely related to it — 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.

These and many other techniques have been developed for clustering conventional data. The interested reader is directed to Kaufman and Rousseeuw (2005) or similar textbook for additional details.

2.2 Search Techniques for Determining Network Topology

The identification of an[1] optimal partition of a network is a challenging problem because of the vast number of possible ways that the network can be partitioned. Even for relatively small networks, the number of possible partitions is large. In most cases, the number of possible partitions is too large to allow the optimality of every possible partition to be evaluated.

Consequently, many heuristic search algorithms have been devised by which the number of possible network partitions whose optimality must be evaluated can be greatly reduced. One spectrum on which to view these different approaches is their relative speed. In general, there appears to be an inverse relationship between the speed and the accuracy of the algorithms that have been developed;[2]

[1]The partition is non-unique in the sense that there are many definitions of *the* optimal partition.

[2]The reader will note that the discussion of speed is only relevant in the case of large networks (with the number of nodes exceeding, say, 10^4).

that is, in order to increase speed, the user must sacrifice accuracy. Similarly, to increase accuracy, speed must be sacrificed (Danon et al., 2005).

While the best-practices benchmarks and methods used to gauge the accuracy and speed of these algorithms are still largely unanswered questions (Fortunato, 2009), one common approach is known as the four-groups test (Newman and Girvan, 2004). The user simulates a network of 128 nodes containing four subgroups, each with 32 member nodes. The links are assigned at random, where the probability that two nodes in the same subgroup are tied is p_{in} and the probability that two nodes from different subgroups are tied is p_{out}. The user gradually increases the value of p_{out}, while keeping the expected degree constant. Larger values of p_{out} correspond to networks with increasingly interconnected (and therefore more difficult to detect) clusters. Danon et al. (2005) provides a good summary and graphical description of the performance of various common algorithms.

The fastest method in common use was developed by Clauset et al. (2004). Their work is a modification of Newman (2004).[3] Newman's algorithm is a greedy, agglomerative, hierarchical clustering algorithm that seeks to maximize modularity at each step.[4] The algorithm begins by assuming that each node represents an individual module, then merges the modules that lead to the greatest increase in modularity. The output of the algorithm is the ubiquitous dendrogram from which the user can choose the most appropriate partition of the network. These agglomerative methods work particularly well with networks that have hierarchical structure — i.e., larger subgroups that are themselves made up of smaller subgroups.

[3]Clauset et al. (2004) do not alter the general approach of this algorithm, but optimize its memory usage, data storage, and computational methods for use with sparse networks; i.e., the vast majority of networks of interest.

[4]For a detailed discussion of *modularity*, see Section 2.4.2.

Many have employed stochastic search methods such as simulated annealing or a genetic algorithm (Guimerà et al., 2004; Kü cükpetek et al., 2005). These methods are generally found to be slower but 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.

A third class of approaches involves examining the spectral properties of various matrices. Newman (2006), for example, denotes the assignment of nodes into two groups in terms of an $n \times 1$ vector $\mathbf{s}$ in which node i's membership in subgroup 1 (2) is denoted by $s_i = 1$ (-1). By choosing the assignment of group membership in such a way as to maximize the dot product of $\mathbf{s}$ and 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.

A fourth class of approaches uses extremal optimization (Duch and Arenas, 2005). Extremal optimization focuses on correcting those nodes with the worst fit. Kernigan and Lin (1970) propose a similar but more simplistic approach in which the graph is divided into equal parts, whereas more recent implementations of this approach allow the algorithm itself to determine the appropriate size and number of partitions.

Finally, 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; e.g., Girvan and Newman (2002); Newman and Girvan (2004).

In closing, an approach to community detection that has not been discussed is the detection of *local* community structure (Clauset, 2005; Papadopoulos et al., 2009). Suppose there is significant missing data in a global network or, more

interestingly, suppose a particular community detection algorithm permits an interactive interface — i.e., if the process is user-directed — these local approaches allow for the detection of modular structure surrounding a particular node of interest. Since local detection methods do not consider the entire network, but rather only those within a few degrees of separation from the central node, these methods are very fast, but result in only a partial picture of the global network topology.

2.3 Regarding Degree Distribution

In an undirected network — i.e., $x_{ij} = x_{ji}$, where x_{ij} is an element of $\mathbf{X}$, the adjacency matrix of the network — the *degree* of a node in a network is the number of links that it has to other nodes. Specifically, $d_r = \sum_{i=1}^{n} x_{ij} = \sum_{j=1}^{n} x_{ij}$. In the case of a directed network, this can be further broken down into *indegree*, $d_{r,in}$, and *outdegree*, $d_{r,out}$; that is, the number of links from node i to any other node in the network is $d_{i,out} = \sum_{i=1}^{n} x_{ij}$ and the number of links going from any other node in the network to node i is $d_{i,in} = \sum_{j=1}^{n} x_{ij}$. Note also that $d = d_{in} + d_{out}$. For simplicity, even with directed networks, frequently only the degree is used.

Suppose Y_k represents the number of nodes with degree k. The distribution of this random variable is known as the degree distribution. Degree distribution plays an important part in categorizing the topology of social networks. The simplest type of graph that has been studied is the Erdős Rényi (ER) random graph. The ER random graph is defined by a stochastic process in which each node is connected to every other node with probability p. It has been shown (Bollobas, 2001) that the degree distribution in this case is asymptotically Poisson with $\lambda = \binom{n-1}{k} p^k (1-p)^{n-1-k}$, where n is the number of nodes in the network.

While these sorts of graphs are often poor models for many real-world networks, they do provide a good starting place for the study of degree distribution because the exact degree distribution can be determined analytically. One meaningful way in which many common networks differ from these ER random graphs is in their degree distribution.[5] For example, Handcock and Jones (2004) researches the degree distribution of a network of sexual partners. Showing that the degree distribution of these networks does not match any of the current models for degree distribution, they argue that the mechanism that brings about the formation of a network is key in understanding and modeling the degree distribution of that network.

Many large networks have been shown to demonstrate a power-law in the right tail of the degree distribution (i.e., a very thick right tail). Specifically, networks following this power-law have degree distribution of the form $P(Y_k = y_k) \propto {Y_k}^{\gamma}$, where γ is typically between 2 and 4. The World Wide Web (Albert et al., 1999) and the Internet (Faloutsos et al., 1999) are two examples of networks that appear to follow this power-law. Barabasi and Albert (1999) proposed a mechanism by which this degree distribution is formed. The Barabasi-Albert (BA) algorithm simulates scale-free networks by beginning with a small number of nodes and then adding new nodes over time. These nodes are connected to the rest of the network by means of a preferential scheme; that is, new nodes are more likely to be connected to those nodes with high degree than to nodes with lower degree. This evolution over time and preferential attachment produces degree distributions that are highly similar to observed scale-free networks. Leary et al. (2007) generalizes the BA algorithm by introducing a tuning parameter

[5]For a discussion of the exact degree distribution of ER and other simplified random graphs, the reader is directed to Barabasi et al. (1999).

that can be used, for example, to increase the level of modularity exhibited by the simulated network.

A third class of degree distributions of interest is exhibited by *small-world* networks. A small-world network is one in which very short paths between any two nodes can be found even for very large networks. This idea was first introduced in Milgram (1967), a psychologist who argued that most people in the United States can be connected, via their acquaintances, in six or fewer steps. Milgram randomly selected test subjects from a Nebraska telephone directory, asking them to send a letter to a particular stockbroker in Boston. If they did not know the stockbroker directly, test subjects were instructed to send the letter to someone that they did know who would be the most likely to know the stockbroker. Considering only the letters that reached the stockbroker, Milgram discovered that the average number of steps taken by each letter was approximately six. Additionally, Newman (2001), analyzing a coauthor network of scientists, also found the average degrees of separation amongst scientists to be around six.

A variety of models have been devised for these small-world networks. One popular model, the Watts-Strogatz (WS) model, has been devised to simulate these small-world networks (Watts and Strogatz, 1998). The algorithm starts with a ring of nodes in which each node is connected to its k nearest neighbors. For example, if $k = 4$, then each node would be connected to the two nodes on either side. The second step of the algorithm is to randomly rewire each edge with probability p. For large values of p, the network approaches the ER random graph and for small values of p, the network retains its ring structure. Values of p between these two extremes generate small-world networks of varying average degrees of separation.

By itself, the degree distribution is an important part of describing network topology, but it also plays a role in the search for subgroups, about which this work is concerned. For example, one of the more popular quality functions (which are discussed in Section 2.4), compares the observed graph with a *null model* that shares the characteristics of the observed graph — e.g., its degree distribution — absent any community structure. One way that has been proposed to preserve these characteristics is by holding the degree distribution constant in the null model.

For further reading, the reader is directed to Albert and Barabasi (2002); Newman (2003); Newman et al. (2001).

2.4 Specification of the Quality Function

In Section 2.2, this work described some of the recommended search techniques for detecting subgroups in networks. Each of these search techniques, however, requires a function to maximize (or minimize). In simulated annealing, this function is known as the objective function. In the genetic algorithm it is known as the fitness function. This work will refer, in general, to any characteristic that describes the extent to which a network can be partitioned into subgroups as the quality function. What follows is a discussion of several of the most popular quality functions.

2.4.1 Betweenness Centrality

Suppose the problem of detecting network topology is constrained to the more restrictive search for densely intraconnected, but sparsely interconnected subgroups of nodes. In this case, one logical approach is to look for those few links

that connect the subgroups to one another. If enough of those links are detected and removed, then the subgroups would appears as disconnected networks, rather than sparsely interconnected subgroups.

One logical approach to detecting these important bridges between subgroups is to calculate a measure of *edge centrality*. Centrality measures are well-known and can be applied either to edges or nodes. For example, a node with high centrality is a node that is important to the connectedness of the network. The same applies to edge centrality; i.e., edges with high centrality are important for maintaining the connectedness of the global network.

Of the many measures of centrality, Girvan and Newman (2002) proposes to use the geodesic edge betweenness (or simply edge betweenness) measure of centrality for determining which edges to remove. The edge betweenness of a given edge is the number of geodesics — i.e., shortest paths between two nodes — containing that edge. Hence, higher values of edge betweenness indicate an edge which connects many otherwise more distantly connected nodes. Girvan and Newman's paper also analyzes other measures of edge betweenness, including random walk edge betweenness and current-flow edge betweenness but find that geodesic edge betweenness is easier — i.e., computationally faster — to calculate and gives better results that other alternatives in practice.

The Girvan-Newman algorithm begins by (1) calculating the edge betweenness of every edge in the global network; (2) Removing the edge with the greatest edge betweenness; and (3) recalculating the edge betweenness for every edge in the resulting global network. The algorithm then repeats steps (2) and (3).

The stopping point for this algorithm is not clearly defined. Without setting a stopping criterion, the algorithm will run until every edge has been removed from

the network. One solution to this problem (Newman and Girvan, 2004) has been to evaluate the modularity (see Section 2.4.2) of each of the partitions that result from the process and choose the partition that results in the highest modularity.

2.4.2 Modularity

In a landmark paper, Newman and Girvan (2004) define a new measure of the extent to which a given partition of a network exhibits community structure. This *modularity* has become one of the most popular quality functions in use today due to its intuitive appeal, its versatility, and its simplicity. Roughly speaking, modularity is (a constant multiple of) the number of edges falling within the groups identified by a particular partition of a network minus the expected number of edges within the groups when the edges are placed in some random (or semi-random) way. More specifically, modularity, Q, can be written

$$Q = \frac{1}{2m} \sum_{ij} \left(x_{ij} - P_{ij} \right) I \left(C_i = C_j \right) \tag{2.1}$$

where m denotes the number of edges in the network, x_{ij} denotes the ijth element of the adjacency matrix, C_i denotes the group membership of node i, I denotes an indicator function that equals one when nodes $i = j$, and P_{ij} denotes the expected number of edges — i.e., the probability of an edge existing — between nodes i and j under the null model.

There are a variety of ways that the null model can be defined. The choice is largely arbitrary, but a good choice is one that preserves many of the characteristics of the network topology without the assumption of any community structure. One choice is to fix the degree distribution of the nodes, but to randomly rewire the connections between them. By cutting all of the connections between nodes,

one leaves $2m$ stubs in the entire graph. Under this approach, since the edges are rewired randomly, the probability of selecting a node i to be one of the pair that is rewired is the ratio of the number of stubs on that node — i.e., its degree — to the total number of stubs in the network ($2m$); that is, $\frac{d_i}{2m}$. Hence, the probability that a particular pair of nodes is connected together in the rewiring process is $\frac{d_i d_j}{4m_2}$. The expected value of the link between nodes i and j, then is $2m\frac{d_i d_j}{4m_2} = \frac{d_i d_j}{2m}$, and Q can be rewritten as

$$Q = \frac{1}{2m} \sum_{ij} \left(x_{ij} - \frac{d_i d_j}{2m} \right) I\left(C_i = C_j\right) \tag{2.2}$$

A variety of other quality functions have been proposed. For example, Costa (2004) proposes detecting communities by detecting the hubs by which those communities are tied together, and many others, e.g., Flake et al. (2000); Hwang, Kim, Ramanathan, and Zhang (Hwang et al.); Ino et al. (2005); Radicchi et al. (2004). Modularity, however, is by far the most popular — likely due to its intuitive appeal and ease of calculation.

2.4.3 Generalized Blockmodeling

Generalized blockmodeling also optimizes a quality function, referred to as the criterion function. The criterion function measures the equivalence between nodes in the network. Various types of equivalence have been identified. Two nodes that are structurally equivalent, for example, have the same neighbors. Blockmodeling attempts to simplify complex networks by grouping together equivalent nodes. Another common type of equivalence is regular equivalence. Two nodes are regularly equivalent if their neighbors are all regularly equivalent (Figure 2.1).

Choice of the criterion function is based on the type of equivalence that is being

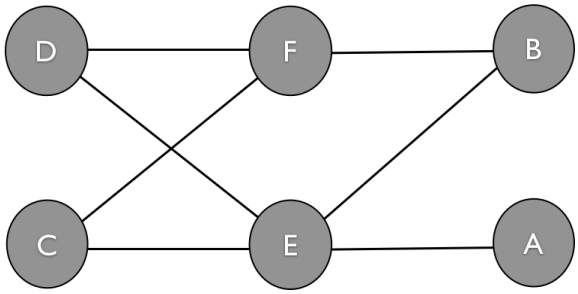

Figure 2.1. Illustrates the concepts of structural and regular equivalence. Nodes C and D are structurally equivalent because they have the same neighbors; i.e., E and F. Nodes C and D, nodes E and F, and nodes A and B, however, are regularly equivalent; that is, groups of regularly equivalent nodes connect to other groups of regularly equivalent nodes in the same way.

considered. The criterion function is chosen such that it is equal to 0 for pairs of nodes that exhibit that type of equivalence and increasing as nodes become less equivalent. There are a variety of methods for fitting these blockmodels. The indirect approach minimizes the dissimilarity between pairs of units based on the chosen criterion function. The direct approach measures the fit of the observed clustering to that of an ideal theoretical model where the types of relationships between groups of nodes are chosen from the set of types of relationships that are defined by the type of equivalence being considered; i.e., there are a finite number of relationships that can exist between groups of nodes — known as *ideal blocks* — given a particular equivalence. For example, for regular equivalence, there are nine ideal blocks. The direct approach appears to be NP-hard; that is, it is only feasible for small networks.

The reader is directed to Doreian et al. (2005) for a text on blockmodeling. While generalized blockmodeling originally treated only binary networks, the ap-

proach has been generalized to treat weighted networks (Žiberna, 2007a,b) and two-mode networks (Doreian et al., 2004)[6]

One interesting hybrid approach between the modularity approach of Section 2.4.2 and blockmodeling was made by Guimerà and Amaral (2005), which uses the modularity approach described above, but adds a second step to assign functional roles to the members of each detected module. They classify nodes as *hub nodes* and *non-hub nodes* based on a standardized measure of its internal degree and further into sub-categories based on how a given node connects to others outside of its own module.

2.4.4 Latent Position Cluster Model

The approaches previously discussed maximize (or minimize) some quality function that describes the optimality (inadequacy) of the partition under consideration. Latent space cluster modeling, which has roots in multi-dimensional scaling, presumes that the nodes of a network exist in a low-dimensional (compared to the number of nodes in the network) space, and seeks to estimate their positions by means of a logistic regression — i.e., the dependent variable is the presence or absence of a tie between two nodes — which includes a term for the positive of each node in an unobservable social space to be estimated. The latent space model was first introduced by Hoff et al. (2002). It does not include a means by which group membership can be estimated, but does seek to place the nodes in Euclidean space, whereby measures of distance between those nodes might be used to guess at the proper clustering. A nice feature of this model, unlike any of

[6]Two-mode networks are networks in which there are two different categories of nodes and links go from the first group to the second group. As an example, one standard two-mode data set is a network of socialites and the social events that they attended. Links never connect the socialites with one another or the events with one another, but links, rather, go from socialite to event indicating attendance.

the previous quality functions, is that it quite naturally provides a treatment for the most common correlations between the connections of the network; that is, nodes which are closer together in the social space are more likely to be connected to one another. Hence, similarity in nodal attributes and transivity[7] are both well modeled by this approach.

Handcock et al. (2007) extend this model to include the estimation of group membership, and Krivitsky et al. (2009) extend this model again to include both group membership and to account for the propensity of a given node to exhibit a higher degree than other nodes.

One advantage of these models is they provide a way to measure the variability in group assignment; that is, nodes which do not fit well into any particular group are easily identified. In Handcock et al. (2007), they demonstrate this uncertainty by labeling each node with a pie chart containing the probability that the node should be assigned to a particular group.

One limitation of these models is they require the user to set the number of groups into which the network is partitioned and the dimensionality of the space into which the network is projected. The authors recommend solving these problems in terms of Bayesian model selection.

2.4.5 Other Approaches to Clustering

Having mentioned the most common and mainstream approaches determining network topology in various forms, what follows is a very brief discussion of a few miscellaneous approaches that are not directly related to this work.

The first of these couch the problem in terms of statistical mechanics. Various

[7]Transivity refers to the effect on the chance that there exists a link between nodes j and k that results from the existence of a given link between nodes i and j and also between nodes i and k.

models, including the Potts model and the Ising model, can be used to find approximate optimal solutions to the problem of determining network topology. By parameterizing the problem in terms of statistical mechanics, finding the optimal cluster is equivalent to finding the ground state of these models. Under certain circumstances, this parameterization of the model is equivalent to maximizing the modularity. For a discussion of these approaches, the reader should see, as a start, Barabasi et al. (1999); Fortunato (2009); Guimerà et al. (2004); Massen and Doye (2008); Newman (2003); Reichardt and Bornholdt (2006, 2007).

The second allows nodes to have membership in more than one cluster — i.e., fuzzy clustering. These approaches argue that limiting group membership to only one cluster of individuals is too restrictive. From an intuitive perspective, this is a reasonable criticism of these approaches. Development of models to treat these groups are ongoing. Interested readers should see Fortunato (2009); Reichardt (2004).

2.5 Regarding Significance

2.5.1 Significance in Conventional Clustering Algorithms

Of interest in general and to this work are methods of measuring the statistical significance of the results of these clustering algorithms. One approach that is closely related to determining the statistical significance of clustering results is validation. Validation approaches, in general, evaluate the stability of the results. If clustering results are stable in some way, then the results must represent some "true" structure within the data set. For example, Levine and Domany (2001) propose an approach whereby repeated subsamples of the original data set are re-clustered by the same approach. The results of these m samples are then

summarized and compared to the results obtained when the entire data set is clustered. A high level of agreement between the original results and the validation results indicate that the clusters detected in the data are stable and, hence, valid in some way. A major drawback to this sort of resampling approach, however, is that resampling can often be computationally expensive (Greene and Cunningham, 2006).

Another stability-based approach involves constructing prediction models from a subset of the data set to be clustered. Tibshirani et al. (2001), e.g., uses a subset of the observed data, the training set, to construct a predictive model for the larger data set. If the centers of the clusterings in the training set are predictive of the clusters detected in the larger set, then the clustering is in some way valid. This approach is used, in particular, to determine the correct number of clusters.

While these validation-type approaches produce useful and informative results, they do not indicate the significance or lack of significance of detected clusters. While it is intuitive that significant clusters are likely stable, it is not clear that non-significant clusters are necessarily unstable.

Another broad category of methods to evaluate the output of clustering algorithms is a parametric approach. For example, McShane et al. (2002) use, as a global test for the presence of clusters, testing for multivariate normality in the data to be clustered. If the data can be reasonably thought to have come from the same multivariate normal distribution, then any detected clusters are not statistically significant. Liu et al. (2008) propose a similar approach.

Finally, some approaches involve resampling and bootstrapping procedures (Felsenstein, 1985; Kerr and Churchill, 2001; Suzuki and Shimodaira, 2004, 2006). These resampling approaches are computationally expensive, which makes them

infeasible for large data sets. Beyond these and parametric approaches, formal hypothesis testing for the output of clustering algorithms is not well developed.

2.5.2 Significance in Community Detection Algorithms

There are many potential applications of the technology of community detection. Due to the increased interest in combatting terrorism in recent years, much research has been done in the field of social network analysis to detect collusion and other covert groups within networks. These techniques might be applied in law enforcement settings as well. These applications illustrate the importance of being able to distinguish statistically significant network topology from that occurring as a result of the stochastic process by which the network developed — i.e., if community detection methods detect a covert group, determining the statistical significance of this group is vital.

What follows is a description of the various approaches that exist in the literature to measure the difference between the observed network and the modularity of the null model specified by the researcher. Many of these approaches, despite discussing *statistical significance*, treat this term rather loosely.

Like clustering conventional data, one approach to determining the "significance" of a given partition of the network is to determine the stability of a particular partition; that is, if group memberships remain the same given various perturbations of the network — e.g., removal/creation connections — then, in some sense, that partition of the network must be significant.

Gfeller et al. (2005) note that by design, many algorithms must assign every node to a cluster, regardless of how poor a fit that assignment is. They propose a method that can be implemented by any clustering algorithm which can be used

with weighted networks. The algorithm adds or subtracts a random quantity from the weights of the links between nodes. By observing which nodes are assigned to many different groups throughout many iterations of this procedure, the algorithm identifies those nodes that are unstable in their group assignment. Karrer et al. (2008) take a similar approach, but work with binary networks and perturbs them by rewiring, rather than by adding weights.

Another approach to inference asks a slightly different question. Roughly comparable to pairwise comparisons common with ANOVA and other global tests, some authors have sought to verify that their group assignments are accurate. Krivitsky et al. (2009) fit a latent cluster random effects model that includes estimates of group membership. They are able to measure the significance of their group membership estimates by means of MCMC simulation.

Rosvall and Bergstrom (2008) recommend a parametric resampling[8] approach to estimate the significance of group membership assignment. Specifically, by fitting a model to the observed network, they simulate additional networks from that model and examine the persistent members of those partitions. For example, in the case of a weighted network, one might resample each link from a Poisson distribution with mean equal to the weight of that link. For many iterations of this procedure, those nodes consistently appearing in the same networks are properly assigned, while those jumping from subgroup to subgroup are poor fits.

In addition to these approaches, some researchers have developed methods for measuring the significance of the detected community structure. Clauset et al. (2006) develop a rigorous definition of hierarchical network structure and use resampling methods to identify the hierarchical structure in a given network. Their approach allows them to determine the significance of this particular heirarchical

[8]The authors use the term "bootstrapping" in error.

structure under various simplifying assumptions including edge independence.

Guimerà et al. (2004); Reichardt and Bornholdt (2006, 2007) argue that any observed modularity must be compared to the expected modularity for an appropriate null model in order to conclude that the observed network is modular. They show that even random graphs — i.e., graphs without inherent community structure — can exhibit high values of modularity. For various network sizes, degree distributions, and cut sizes, these authors describe the expected value of modularity given various simplifications including equal group size. No mention in these works is made of the variability of the modularity of random graphs. Without this important information, statistical testing is not possible.

CHAPTER 3

PROPOSED CONTRIBUTION

This work seeks to address three missing elements in the current research:

First, the most popular quality function is Newman and Girvan's modularity, but even they admit that it is lacking in generality. They ask, "Could there be interesting and relevant structural features of networks that we have failed to find simply because we haven't thought to measure the right thing?" (Newman and Leicht, 2007). Many fast search methods use modularity — e.g., Clauset et al. (2004); Newman (2004) — for their quality function, and modularity has an intuitive appeal. In addition to that, modularity compares the observed links to the expected links in a null model in a way that is reminiscent of a goodness-of-fit test. Despite these features, however, it can be seen by inspection that modularity considers only the links within each given subgroup in its calculations; that is, by searching for partitions such that the within-group densities are increased, compared to the expected density, the between-group densities will be forced to fall.

In Chapter 4, this work suggests that the definition of community structure is too restrictive. Since modularity and other popular quality functions consider only the number/value of links within detected groups, increased versatility and power should be obtained by devising quality functions that use all of the information in the network. This work proposes three such statistics. The first is based

on a likelihood ratio test statistic. The second is based on a chi-square test of independence. The third is based on mean absolute deviation. These variations allow for the detection of the structural features that can be detected by block-modeling and other more general approaches without adding the computational complexity that makes these approaches infeasible with large networks. Additionally, the likelihood-based approach provides a means for hypothesis testing that will be discussed in later chapters.

Chapter 5 develops a general and novel statistical test for the significance of clustering algorithms. This test can easily (and without simulation) be applied to clustering algorithms that act on traditional as well as network data. This chapter demonstrates the effectiveness of the test by applying it to simulated and real-world networks.

Finally, the results in Chapter 5 provide an important breakthrough in evaluating the quality of clustering algorithms. Previously, the quality of clustering algorithms has been gauged based on how well they cluster data sets of known structure. This approach is inadequate for a variety of reasons. First, the number of real-world data sets of known structure is small. That a particular clustering algorithm performs well on a particular data set may not be a reliable indicator of its quality. Second, an appropriate method for selecting benchmark networks is not obvious. Third, simulated networks of known structure can be developed, but the high computational cost of many clustering algorithms limits the number of repetitions that can reasonably be performed. Finally, the optimal partition of a given data set into k groups is simply not known. Chapter 6 proposes a means of comparing the results of any clustering algorithm to the theoretical distribution of the optimal partition by means of the test statistic developed in Chapter 5.

CHAPTER 4

GENERALIZING MODULARITY

Modularity is a useful, intuitive, and effective statistic for describing the structure of complex networks. Introduced by Newman and Girvan (2004), modularity describes the extent to which a network is *modular*; i.e., the extent to which it is made up of densely intraconnected yet sparsely interconnected subgroups of nodes. In fact, it was formulated specifically for this purpose, and in many applications — e.g., parallel computing — this sort of structure is the only structure of interest.

It is often the case, however, that the structure of a particular network is unknown to the researcher or practitioner. In fact, community detection algorithms are frequently used as exploratory tools for discerning the structure of such unknown networks. Similarly, it is not hard to imagine other structures that might be of interest to the researcher.

Newman and Leicht (2007) suggest one such example: Consider a network consisting of four equally sized groups whose within-group ties occur with probability p_{in} and whose between-group ties occur with probability p_{out} — presumably with $p_{in} > p_{out}$, although this is not necessary. In addition to these four groups there are eight extra nodes, each of which are connected to the other four groups in unique ways (Figure 4.1). In this case, the first block of nodes connects to keystones $\{1, 2, 3, 4\}$; the second block connects with nodes $\{3, 4, 5, 6\}$, and so on.

The maximization of modularity has very little power to detect the structure of this network. Many other examples could be described to illustrate this sort of limitation.

The reason that the maximization of modularity is unable to detect other types of community structure is that it disregards much of the information contained in the network. It is, in essence, the sum of the differences between observed and expected *within* group links, disregarding the presence or absence of links *between* groups.

What follows is a description of three more general statistics, the maximization of which will allow the user to detect a much wider variety of structures. The first is based on the likelihood ratio test statistic, while the second and third are generalizations of modularity.

4.1 Three General Quality Functions

Recall the definition of modularity:

$$Q = \frac{1}{2m} \sum_{ij} \left(x_{ij} - \frac{d_i d_j}{2m} \right) I\left(\mathbf{c}_i = \mathbf{c}_j\right) \tag{4.1}$$

where m denotes the number of edges, x_{ij} denotes the ij^{th} element of the adjacency matrix $\mathbf{X}$, d_i denotes the degree of node i, and $\mathbf{c}_i$ denotes the i^{th} column of $\mathbf{C}$. Let $\mathbf{C}$ be an $n \times r$ binary matrix of group membership in which the rows sum to 1 and $c_{ij} = 1$ indicates that node i is a member of the j^{th} group.

Let o_{ij}, the ij^{th} element of the $r \times r$ matrix $\mathbf{O}$, denote the number of links connecting members of groups i and j. In matrix form, $\mathbf{O}$ can be written

$$\mathbf{O} = \frac{1}{2}\mathbf{C}^T\mathbf{X}\mathbf{C} \tag{4.2}$$

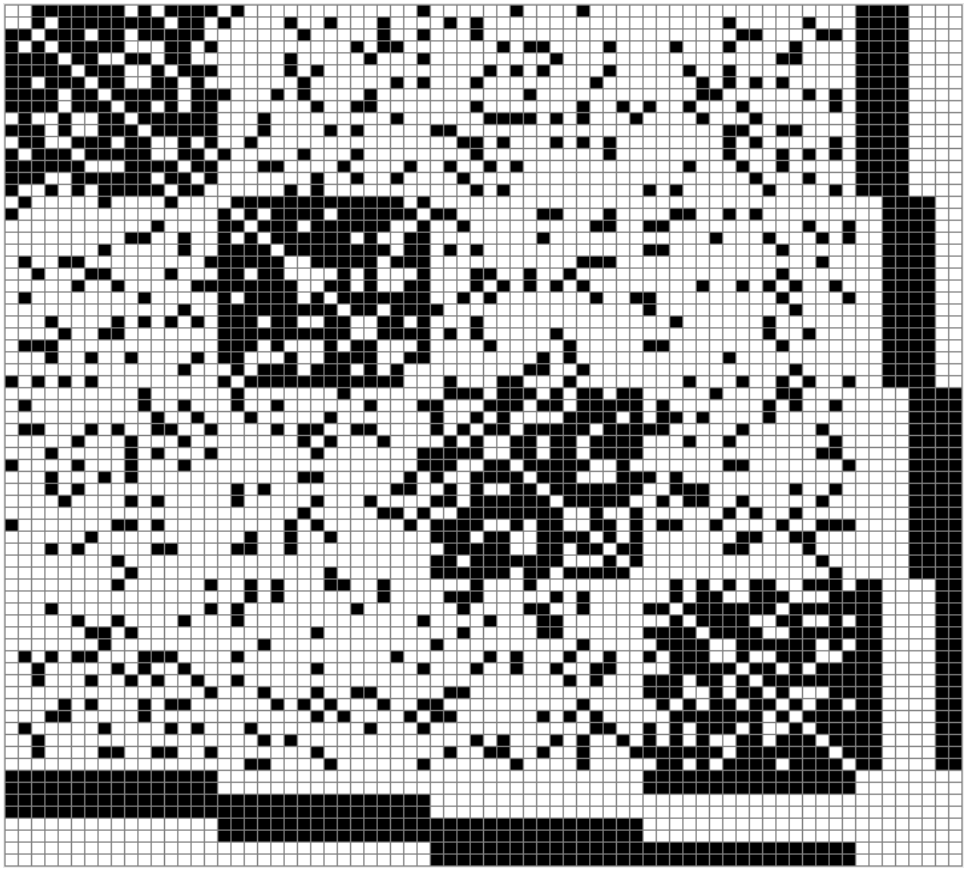

Figure 4.1. A simulated keystone graph based on those described in Newman and Leicht (2007). Black and white squares indicate the presence or absence of a link, respectively. Each of the larger blocks is uniquely connected to eight keystone nodes.

Furthermore, let e_{ij}, the ij^{th} element of the $r \times r$ matrix $\mathbf{E}$, denote the *expected* number of links between the members of groups i and j — as defined by Newman and Girvan in their definition of modularity. Noting that the degree of node i is the i^{th} element of $\mathbf{Xj}$, where $\mathbf{j}$ is an $n \times 1$ vector of 1's, $\mathbf{E}$ can be written

$$\mathbf{E} = \frac{1}{4m}\mathbf{C}^T(\mathbf{Xj})(\mathbf{Xj})^T\mathbf{C} = \frac{1}{4m}\mathbf{C}^T\mathbf{Xjj}^T\mathbf{X}^T\mathbf{C} = \frac{1}{4m}\mathbf{C}^T\mathbf{XJX}^T\mathbf{C} \tag{4.3}$$

where $\mathbf{J}$ is an $n \times n$ matrix of 1's. In matrix form, then, Q can be expressed as

$$Q = \frac{1}{m}tr(\mathbf{O} - \mathbf{E}) \tag{4.4}$$

Observing that only the diagonal elements of the matrix $\mathbf{O} - \mathbf{E}$ are included in modularity, the following, more general, statistics are proposed.

4.1.1 A Likelihood-Based Approach

The Likelihood-Ratio Test (LRT) is a well developed statistical procedure 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.

As a simple example, let $\mathbf{y_1}$ and $\mathbf{y_2}$ denote independent, simple random samples. Suppose the researcher believes that the population or populations from which these samples were drawn have the same shape and spread, but the researcher is unsure about the equality of their means. In the case that the means are equal, the two samples are drawn from identical populations; i.e., a simpler model is preferred. In the case that the two means are unequal, the more complex model is preferred.

Let $f_1(y)$ and $f_2(y)$ denote the density functions of observations from the first and second samples, respectively, where $f_1(y) \equiv f_2(y) \equiv f(y)$ in the simpler model. The likelihood function under the simpler model, then, is $L(\theta_0; \mathbf{y_1}, \mathbf{y_2}) = \prod_{ij} f(y_{ij})$ where θ_0 denotes the single unknown mean. Similarly, the likelihood function under the more complex model is $L(\theta_1, \theta_2; \mathbf{y_1}, \mathbf{y_2}) = \prod_i f_1(y_{1i}) \prod_j f_2(y_{2j})$, where θ_1 and θ_2 denote the means of each population.

The likelihood ratio, then, is

$$\Lambda = \frac{L(\hat{\theta}_0; \mathbf{y_1}, \mathbf{y_2})}{L(\hat{\theta}_1, \hat{\theta}_2; \mathbf{y_1}, \mathbf{y_2})} \tag{4.5}$$

where $\hat{\theta}$ is the maximum likelihood estimate of θ. The LRT statistic is

$$D = -2\ln(\Lambda) \tag{4.6}$$

which asymptotically follows a χ^2 distribution with degrees of freedom equal to the difference in the number of parameters between the two models; $df = 2 - 1 = 1$ in this case. For a review of the likelihood ratio test and related statistical procedures, the reader is referred to Hogg et al. (2005) or other mathematical statistics text.

The same approach can be applied to measuring the quality of a given partition of a network. First, let the observed edges represent random observations from a particular class of distributions. In the case of a binary network, the Bernoulli distribution is an obvious choice in which 1 and 0 indicate the presence or absence of a link, respectively. For integer-valued networks, the Poisson or negative binomial distributions are a natural fit. The distribution and type of edges in the network will dictate the choice of distribution.

The simplest case is that there are no subgroups in the network; i.e., there is one subgroup consisting of the entire network. Let $f(x_{ij};\boldsymbol{\theta}_0)$ denote the density function of the edges in the network, where $\boldsymbol{\theta}_0$ denotes the vector of parameters associated with f and the elements of the adjacency matrix $\mathbf{X}$, denoted x_{ij}, indicate the value of the link between nodes i and j. Under the assumption of independence, the likelihood function in the simpler model is $L(\boldsymbol{\theta}_0;\mathbf{X}) = \prod_{ij} f(x_{ij};\boldsymbol{\theta}_0)$.

As an example, suppose the observed network is both undirected and unweighted. Let f be the density function of the Bernoulli distribution with $\boldsymbol{\theta}_0 = p_0$. Concluding that this model is the most appropriate indicates that the observed network is an ER random graph with $p = p_0$.

On the other hand, suppose that the network is more complex. Consider a more complex model that indicates the presence of k subgroups within the network. Retaining the assumption of independence, let f_{rs} denote the density function of the edges connecting members of groups r and s with corresponding matrix of parameters $\boldsymbol{\theta} = [\boldsymbol{\theta}_{rs}]$. Denote the group membership as $C = \{C_1, C_2, \ldots, C_k\}$. The likelihood function under this more complex model is $L(\boldsymbol{\theta};\mathbf{X}) = \prod_{rs}\left[\prod_{i\in C_r, j\in C_s} f(x_{ij})\right]$.

Let f_{ij} be the density function of a Bernoulli distribution with $\boldsymbol{\theta}_{ij} = p_{ij}$. This model indicates that there are k distinct subgroups within the network and that members of groups i and j are connected with probability p_{ij}.

The LRT statistic associated with these two models is

$$D = -2\ln\left(\frac{L(\hat{\boldsymbol{\theta}}_0;\mathbf{X})}{L(\hat{\boldsymbol{\theta}};\mathbf{X})}\right) \tag{4.7}$$

where $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\theta}}_0$ are the maximum likelihood estimates of $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_0$, respectively. Large values of D indicate that the more complex model is more likely while small values of D indicate that the simpler model is more likely.

The parameters in this case are estimated by means of the maximum likelihood estimator, $\hat{p} = \frac{\sum x_{ij}}{n}$, where x_{ij} is the value (1 or 0) of the observed link between nodes i and j and n is the total number of possible links. The density function of the Bernoulli distribution is $f(x) = p^x(1-p)^{1-x}$ when $x = 0, 1$ and 0 elsewhere. Hence the LRT statistic for the examples in the following analysis is

$$D = -2\ln\left(\frac{\hat{p}_0^m(1-\hat{p}_0)^{N-m}}{\prod_{rs}\hat{p}_{rs}^{m_{rs}}(1-\hat{p}_{rs})^{N_{rs}-m_{rs}}}\right) \tag{4.8}$$

where m denotes the number of edges in the network, m_{rs} denotes the number of edges between nodes in groups r and s, N denotes the number of possible edges in the network, and N_{rs} denotes the number of possible edges between nodes in groups r and s.

4.1.2 Generalized Squared Modularity

When modularity is written as in Equation 4.4, the resemblance between it and the well-known χ^2 test statistic is apparent. The elements of matrix $\mathbf{O}$ simply display counts of links between subgroups. Loosely, $\mathbf{O}$ is similar to a two-way contingency table that displays the relationship between two categorical variables.[1] In some (nontechnical) sense, the identification of distinct groups can be considered to be a test of independence between group membership and the formation of links; i.e., is the assigned group membership of a particular node independent of the way in which that node relates to the rest of the network.

In line with this conceptualization, define *Generalized Squared Modularity*, S,

[1] In a typical two-way table, each observation is counted exactly one time. In this case, however, since the degree for each node is allowed to exceed 1, each node may contribute more than one entry into the table.

as

$$S = \sum_{i=1}^{r} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}} \tag{4.9}$$

where r denotes the number of groups into which the network is divided, and o_{ij} and e_{ij} are defined as above.

4.1.3 Generalized Absolute Modularity

Intuitively, the information in the off-diagonal elements of the matrix ($\mathbf{O} - \mathbf{E}$) that are disregarded in Equation 4.4 should contain information about the community structure of the network being studied. In the case that group r is more densely connected to group s, o_{ij} will exceed e_{ij} and in the case that group i is more sparsely connected to group j, e_{ij} will exceed o_{ij}. If the absolute value of these differences, then, are included in the measure of community structure, the result should be increased sensitivity not only to standard modularity — densely intraconnected subgroups groups implies sparsely interconnected subgroups — but also to a variety of different types of structures. Define *Generalized Absolute Modularity*, H, as

$$H = \sum_{i=1}^{s} \sum_{j=1}^{r} |o_{ij} - e_{ij}| \tag{4.10}$$

where r denotes the number of groups into which the network is divided, and o_{ij} and e_{ij} are defined as above.

4.2 Comparison of the Proposed Statistics

The purpose of this chapter is to develop more general quality functions for use with community structure search algorithms that allow for the detection of a wider variety of network structures than has previously been possible. Section 4.2.1

compares the effectiveness of maximizing the three proposed quality functions with that of modularity to detect specific types of community structure in various simulated networks. Section 4.2.2 applies the same search algorithm using each of these four quality functions to several real-world networks and compares the results.

4.2.1 Comparison Based on Simulated Networks

There are a variety of heuristic techniques that have been proposed for detecting a particular optimum partition. In comparing these different algorithms there are various procedures that have been suggested; e.g., the four-groups test (Newman and Girvan, 2004) and the keystone test (Newman and Leicht, 2007). These and other approaches have largely been used to compare the effectiveness of various search algorithms. In this case, however, these benchmark graphs will be used to compare the effectiveness of the same search algorithm employing different quality functions. The choice of search algorithm is somewhat arbitrary. In this case, the four quality functions under consideration will be maximized using an agglomerative hierarchical technique similar to that used by Newman (2004).

Three simulated benchmark graphs will be used to compare the four quality functions. A modified four-groups test, based on Newman and Girvan (2004); a modified keystone test, based on Newman and Leicht (2007); and the periphery test, a new benchmark (Figure 4.2).

The four-groups test simulates random graphs that demonstrate traditional modularity. In this simulation, each graph consists of four groups of ten nodes each. The nodes are connected within groups with probability, p_{in} and between groups with probability p_{out}. The value of p_{in} is increased in a stepwise fashion

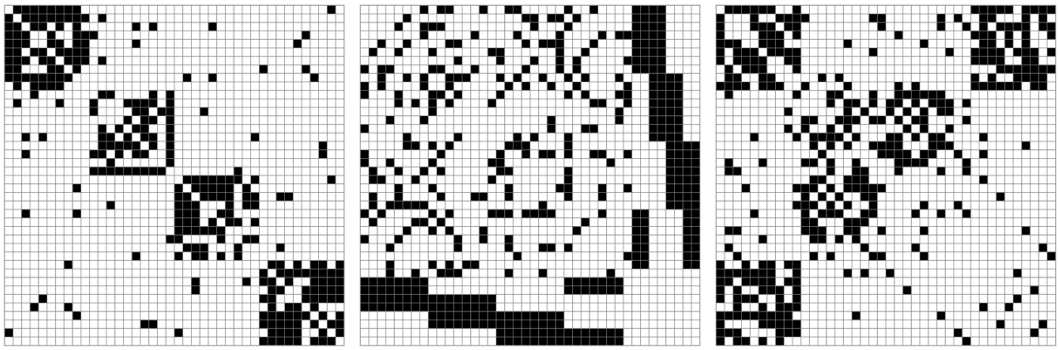

Figure 4.2. Three sample networks for the evaluation of proposed quality functions. (a) A simulated network for the four-groups test. (b) A simulated network for the keystone test. (c) A simulated network for the periphery test.

and p_{out} is set such that the expected degree remains constant throughout the simulation; that is, let d denote the degree of a node in the network. Then, $E(d) = E(d_{within}) + E(d_{between}) = \frac{n}{k}p_{in} + \frac{n(k-1)}{k}p_{out}$. So, $p_{out} = \frac{kE(d)-np_{in}}{n(k-1)}$.

The keystone test was first proposed by Newman and Leicht (2007) to demonstrate that simply maximizing modularity will often miss the key features of a network's structure. The keystone test presents a challenge to standard algorithms because most of the nodes in the network are connected to one another in precisely the same way. There are two different types of nodes in the keystone test: There are eight keystone nodes and four groups of main nodes which relate within and between groups in the same way; i.e., apart from the keystone nodes, the rest of the nodes form an ER random graph. Each group, however, uniquely relates to two of the eight keystone nodes (Figure 4.2). The keystone test is designed to determine whether the maximization of the various quality functions will allow the user to distinguish between the four groups of main nodes based mainly upon

their links to the eight keystones.

The periphery test is designed to determine whether the quality function under consideration can distinguish between different types of nodes within a larger cluster. As an example, consider an influential leadership team and the people surrounding that team in a large corporation. Consider a network drawn from the e-mail correspondence of this group of people. The leaders of that team would likely trade frequent e-mails, maintaining constant communication. One layer beyond that might be people like personal assistants, analysts, vendors, and other individuals that also correspond heavily with the leadership core, but have very little communication amongst themselves. In a study of the network of people involved in this project, distinguishing between the core planners and the peripheral support staff is one vital aspect of community structure that is of great interest. This sort of structure likely exists in criminal and terrorist networks, as well.

The periphery test is designed to determine if the maximization of the various quality functions will allow the user to correctly distinguish the core from the periphery. In this particular benchmark, there are four groups. Two of them are core groups and two are peripheral groups in the sense described above. Just as in the four-groups test, the nodes within each core group are connected with probability p_{in} and to the nodes in one of the peripheral groups also with probability p_{in} (Figure 4.2).

For each data point and for each of the quality function, 20 networks were simulated and clustered using an agglomerative hierarchical clustering algorithm similar to what was used by Newman (2004). Denote the groups identified by the algorithm as $X_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,r_i}\}$, $i = 1, 2, \ldots, k$, and the true groups as

$Y_i = \{y_{i,1}, y_{i,2}, \ldots, y_{i,\frac{n}{k}}\}$. Each experimental group might contain as many as $\frac{n}{k}$ members of the same true group, where n is the number of nodes in the network and k is the number of groups. Define the number of correctly categorized nodes as $\sum_{j=1}^{k}\left[\max_{k}\left[\sum_{i=1}^{r_j} I(x_{j,i} \in Y_k)\right]\right]$, where I is a simple indicator function.

4.2.1.1 The Four-Groups Test

Since this simulation generates graphs that exhibit precisely the sort of structure that modularity was developed to detect, modularity is expected to perform well in this exercise. Figure 4.3 shows, however, that not only does modularity perform well but also that most of the other quality functions perform at least as well over a particular range of values of p_{in}.

At the left side of the charts displayed in Figure 4.3 the value of p_{in} is 0. This indicates that the members of subgroup i, say, are not connected to the other members of subgroup i. Since the expected degree is held constant, this results in a higher value for p_{out}; that is, subgroups are more densely interconnected.

As the value of p_{in} increases (moving to the right in Figure 4.3), the value of p_{out} decreases. This decrease is not linear and depends on the expected degree and the number and size of groups in the simulation. At the right side of each plot in Figure 4.3, p_{in} is at its highest possible value and p_{out} is at its lowest possible value given the expected degree and the size and number of groups. In other words, the left side of each plot shows the effectiveness of the search algorithm using each of the quality functions when the four groups are sparsely intraconnected and densely interconnected (the opposite of what modularity was designed to detect), and the right side of each plot shows the effectiveness of the search algorithm using each of the quality functions when the four groups are densely intraconnected and

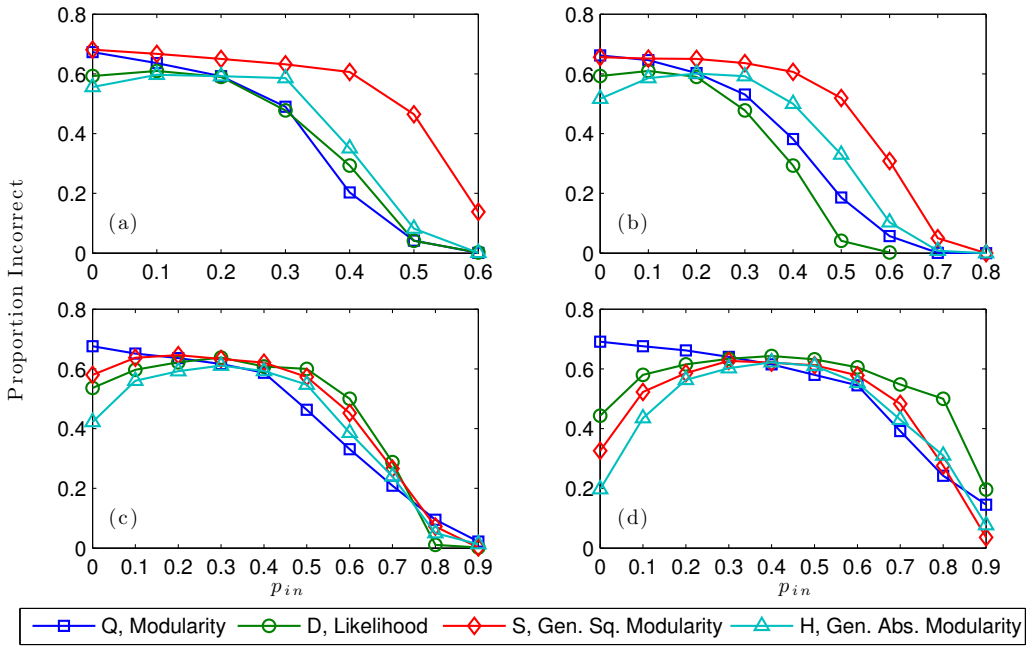

Figure 4.3. A comparison of the proportion of incorrect classifications obtained in the four-groups test upon maximizing each of the four quality functions under consideration. The number nodes in each simulation is 40, with four equally sized groups. The probability, p_{in}, of within-group connections (horizontal axis) and the probability, p_{out}, of between-group connections are set to hold the expected degree constant. Plots (a), (b), (c), and (d) have expected degree of 6, 8, 12, and 16 respectively. Results are averaged over 20 simulated graphs.

sparsely interconnected (precisely what modularity was designed to detect).

Search algorithms that maximize modularity show very little ability to correctly classify nodes in four-groups networks with low values of p_{in}. In fact, the proportion of incorrectly classified nodes associated with the maximization of modularity is strictly decreasing in each of the four cases studied here.

In contrast, the other three measures of quality demonstrate an arcing pattern; that is, when each is maximized, they are able not only to detect modular networks, but also these "anti-modular" networks. As p_{in} increases and p_{out} decreases, there comes a point when the nodes connect within-groups in exactly the same way as they connect between-groups. This point can be seen in Figure 4.3 as the apex of the arc associated with the use of these more general quality functions.

Generalized Squared Modularity, S, appears to be ineffective in sparse networks, while the maximization of the LRT statistic, D, and Generalized Absolute Modularity, H, both appear to outperform the other measures when considering the range of possibilities.

4.2.1.2 The Keystone Test

Figure 4.4 shows that the maximization of modularity was ineffective at distinguishing between the groups of main nodes. The likelihood approach produced the best results with fewer than 5% incorrect classification for all levels of expected degree in networks of 32 nodes. The likelihood approach performed well in larger networks with 56 nodes as well. The maximization of Generalized Squared Modularity, S, and Generalized Absolute Modularity, H, produced opposite results with S performing better for more sparse networks and H performing better in more dense networks. Hence, the likelihood approach is more robust, in this case.

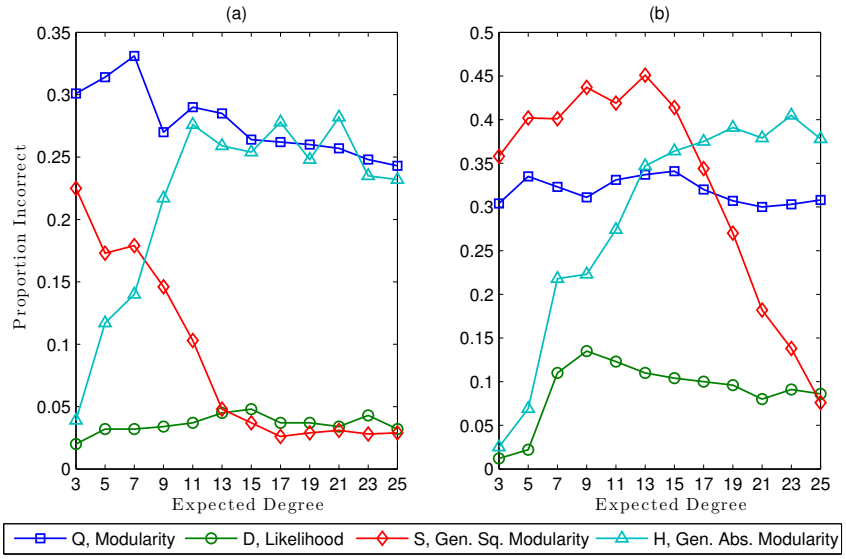

Figure 4.4. A comparison of the proportion of incorrect classifications obtained in the keystone test upon maximizing each of the four quality functions. Plots (a) and (b) display results from simulated networks containing 32 and 56 nodes, respectively. The plots show the expected degree of the main nodes on the horizontal axis. Results are averaged over 20 simulated graphs.

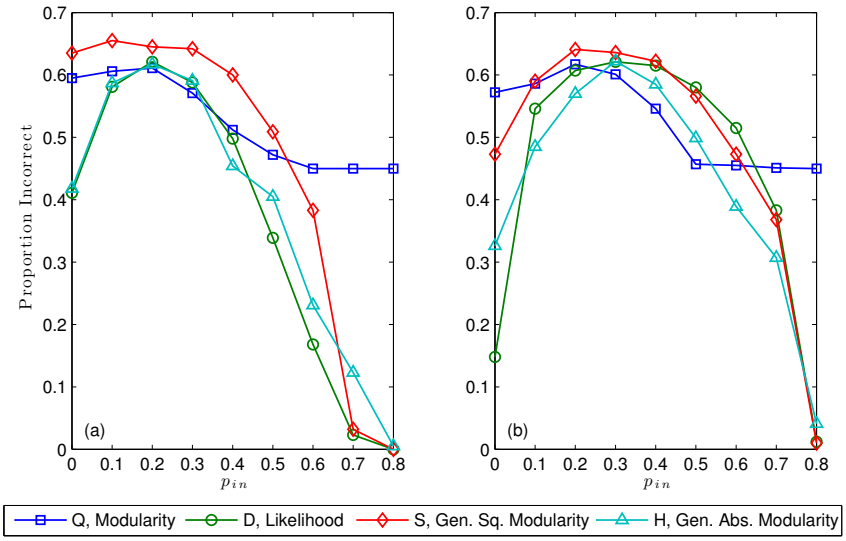

Figure 4.5. A comparison of the proportion of incorrect classifications obtained in the periphery test upon maximizing each of the four quality functions under consideration. The number nodes in each simulation is 40, with four equally sized groups, and the nodes connect as described in the text. Plots (a) and (b) were generated from simulated networks with expected degree of 8 and 12, respectively. Results are averaged over 20 simulated graphs.

4.2.1.3 The Periphery Test

Figure 4.5 shows that the maximization of modularity is ineffective at distinguishing between the core and peripheral nodes in the periphery test. Values of p_{in} are set as above, and the maximization of modularity leads to the correct classification of just over 50% of nodes at its very best. In addition, as the density within the core group and between the core and periphery groups increases, the maximization of the other three quality functions produces much better results. Furthermore, the same arcing pattern associated with these more general quality functions can be seen, the apex of which coincides with the point at which the

levels of p_{in} and p_{out} make the probability of links between any two nodes equal, regardless of group membership. As before, the maximization of S appears to produce poorer results in more sparse networks. Also as before, the maximization of D and H produce similar results. Overall, the results shown in Figure 4.5 agree with what was shown in Figure 4.3 except that the maximization of modularity has proven ineffective in this case.

4.2.2 Comparison Based on Real-World Networks

The application of these approaches to real-world networks is quite different from the application to simulated networks. When applying different quality functions to networks of known structure, the test of effectiveness is as simple as determining which approaches adequately identify that structure. In the case of real-world networks, on the other hand, the structure is not known at the beginning (or even at the end) of the process. The maximization of different quality functions by means of some search algorithm (in this case hierarchical agglomeration) can and should produce different results depending on the quality function used. More general quality functions such as the ones described herein allow the network, rather than the quality function itself, to dictate what sort of structure is detected.

Two networks were analyzed using each of the quality functions under consideration. Zachary's Karate Club is a well-known benchmark used with community detection algorithms (Zachary, 1977). It consists of 34 members of a karate club that were studied for two years. At some point a conflict between two prominent members in the club led to the club splitting. The links in this network represent interactions outside of regular club activities.

The second network is an unpublished but well-known network of political books compiled by Krebs. The nodes represent 105 political books that can be purchased on amazon.com, and the links between them indicate books that were frequently purchased by the same buyer. Additionally, Newman (2006) classified these books according to ideology; i.e., conservative, neutral, and liberal.

These two data sets were chosen for two reasons. First, they illustrate the limitations of the more restrictive quality functions and emphasize the strengths of more general quality functions. Secondly, the two data sets are small enough (34 nodes and 105 nodes, respectively) to allow the easy visualization of the results.

When using hierarchical clustering methods, one question is how to determine the correct number groups; i.e., the stopping point of the algorithm. For modularity, there is a natural stopping point where the modularity is at a maximum. In the other three cases, however, the value of the quality function is generally monotonic. In those cases, the stopping point was chosen by inspection; that is, the number of groups was chosen that seemed to give the most informative and most natural division of the network under consideration. Model selection approaches are outside the scope of this work, but could easily be applied here.

Figures 4.6 and 4.7 display the results of the simulation. Each pane of these plots represents the reordered adjacency matrix of the clustered network. The ith row and column correspond to the ith row and column of the adjacency matrix with black squares indicating the presence of a link, and white squares indicating the absence of a link. On the right and bottom of each pane are true group membership bands.

4.2.2.1 Zachary's Karate Club

The maximization of modularity, Q, and Generalized Squared Modularity, S, produced similar results (Figure 4.6). Maximizing modularity produced three groups. The first and second of these groups are related to one another in that their combined membership correctly identifies one of the factions into which the karate club split (all but node 10). The third group correctly identifies the remainder of the second faction in the karate club. As expected, within group densities are higher than the global density of 0.133 (Table 4.1).

The maximization of S produced similar results with the same mis-categorization of node 10. The process resulted in only 2 groups. The maximization of S by this method appears to be prone to "snowballing"; that is, once groups form, it is likely that each step of the algorithm simply adds singletons onto the already formed group.

The maximization of the likelihood ratio test statistic and Generalized Absolute Modularity produced more informative results. Not only did both approaches correctly categorize every node (including node 10), but these approaches also provide much finer relational detail about the nodes. Figure 4.6D, for example, shows seven distinct groups within the larger network. The first (node 1) and seventh groups (nodes 33 and 34) represent the leaders of each of the two factions. Groups 2 and 3 represent two modular clusters (densities of 0.6 and 0.9, respectively) of nodes that are also well connected to the leader of faction 1. Groups 2 and 3 also differ in that group 2 is isolated except for its connections to the leader of the first faction, while group 3 is well connected to other groups in the network. Group 6 is similar to groups 2 and 3. Group 6 is a dense cluster of nodes (density = 0.321) connected to the leaders of faction 2. Groups 2, 3, and 6 follow the traditional

definition of modularity (Table 4.1).

TABLE 4.1

Within-group densities for Zachary's karate club data set. The global density is 0.133. Asterisks indicates the group contains only one member.

Group	Modularity	Likelihood	Gen. Sq. Modularity	Gen. Abs. Modularity
1	0.429	0.000*	0.25	0.000*
2	0.361	0.600	0.243	0.600
3	0.243	0.900	-	0.289
4	-	0.000	-	0.048
5	-	0.036	-	0.250
6	-	0.321	-	1.000
7	-	1.000	-	-

Groups 4 and 5, on the other hand, are very sparse (densities of 0.0 and 0.036) but distinct in that they are all connected to the leaders of their respective factions; that is, the members of groups 4 and 5 are peripheral nodes (Section 4.2.1.3). Were they not connected to the faction leaders, they would be almost totally unconnected to the network.

With a few variations, the maximization of Generalized Absolute Modularity, H, produces results similar to that of D, but with poorer resolution. Maximizing H produces fewer groups, with singletons occurring at preceding steps of the clustering algorithm.

4.2.2.2 Kreb's Political Books

The results for Kreb's political books data set also show that the maximization of D and H give increased resolution and increased information about the structure of the network. Maximizing Q and S yield fewer groups, but with similar rates of correct categorization. In fact, the maximization of each of the quality functions produce similar rates of correct classification. Maximizing Q and S both produce groups of modular structure with density higher than that of the global network, as expected (Table 4.2).

Of greater interest is the information revealed when D and, to a lesser extent, H are maximized. Figure 4.7D shows there is more than just modular structure in this network. Groups 2 and 5 contain conservative and liberal books, respectively. Table 4.2 shows these books are distinct in that they are very frequently purchased together (densities of 0.733 and 0.727), and people who purchase these books are very likely to purchase any in a wide variety of other books of similar ideology. Groups 1 and 3 (group 4), however, do not share that same structure. People who buy books in these groups are not likely to purchase other books in their group. They are much more likely to purchase books from group 2 (group 5).

This information cannot be determined from the results shown in Figure 4.7Q or 4.7S, yet this information has considerable importance for, say, amazon.com. People who purchase a book from groups 1, 3 (group 4) should not be offered

TABLE 4.2

Within-group densities for Kreb's political books data set. The global density is 0.08.

Group	Modularity	Likelihood	Gen. Sq. Modularity	Gen. Abs. Modularity
1	0.171	0.174	0.155	0.279
2	0.276	0.733	0.158	0.163
3	0.213	0.130	-	0.129
4	-	0.086	-	0.146
5	-	0.727	-	0.582

books from those categories. Rather, they should be shown books from group 2 (group 5) because they are far more likely to purchase books in those groups, according to the data. Conversely, purchasers of books from groups 2 (group 5) should be shown books from groups 1 and 3 (group 4) as well as books from their own group; that is, groups 2 and 5 are roughly analogous to the keystones discussed in Section 4.2.1.2.

4.2.3 Conclusions

The use of modularity as the quality function in network clustering algorithms was effective at detecting modular structure; i.e., densely intraconnected, sparsely interconnected groups of nodes. The maximization of modularity, however, did not reliably lead to the detection of other types of community structure. Search

algorithms that rely on modularity for detecting structure are restricted to this very specific definition of community structure.

The proposed quality functions, on the other hand, demonstrated effectiveness at detecting a variety of network topologies. The worst performing among these was Generalized Squared Modularity. The maximization of S did not produce uniform results, performing very poorly under some circumstances and well under others. Generalized absolute modularity performed better, sometimes outperforming all of the other quality functions. The likelihood approach, however, produced the most robust and more informative results.

In applying these different quality functions to real-world networks, the maximization of D produced the highest resolution and the most informative groups of all the quality functions examined. The maximization of D allowed the user to identify new structures in the data such as peripheral groups and keystone groups without increasing the mis-classification rate of the algorithm. The maximization of D was followed by H, Q, and S, in order of effectiveness. The maximization of these quality functions produced groups of increasing size and decreasing variety.

In summary, the maximization of D produced the most refined and informative results in the identification of network topology followed by the maximization of Generalized Absolute Modularity, modularity, and Generalized Squared Modularity.

Continued research is needed to compare the resolution limits of these new approaches to that of modularity. Fortunato and Barthélemy (2007) show that the resolution of modularity is limited. These preliminary results suggest the likelihood approach does not suffer from this condition, since the likelihood ratio test statistic is generally at its highest when the each node represents a single

group. Additionally, these new approaches need to be tested on large networks to demonstrate that these promising results are scalable.

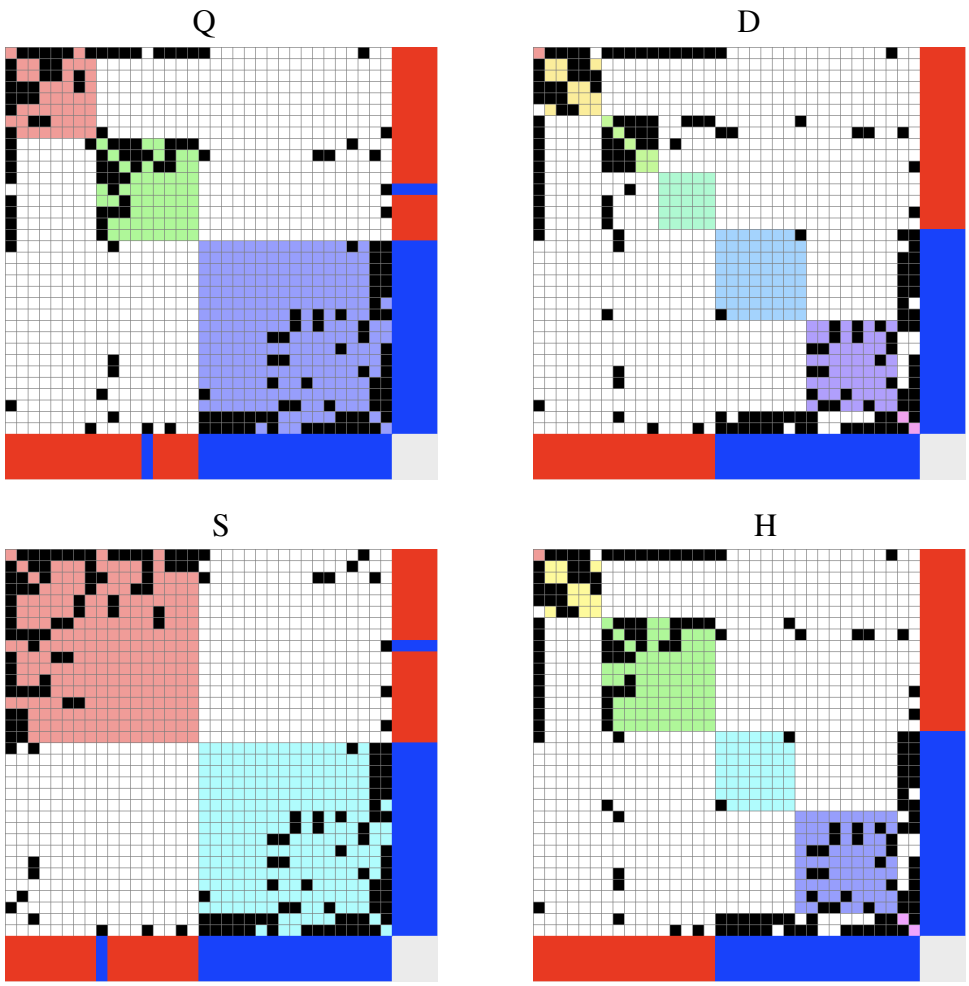

Figure 4.6. Ordered graphical adjacency matrices of Zachary's karate club data that result from the use of each of the discussed quality functions. A black square in the ijth position indicates a link between nodes i and j. A white square indicates the absence of a link. The colored bars at the right and bottom of each pane represent the correct true group membership of nodes.

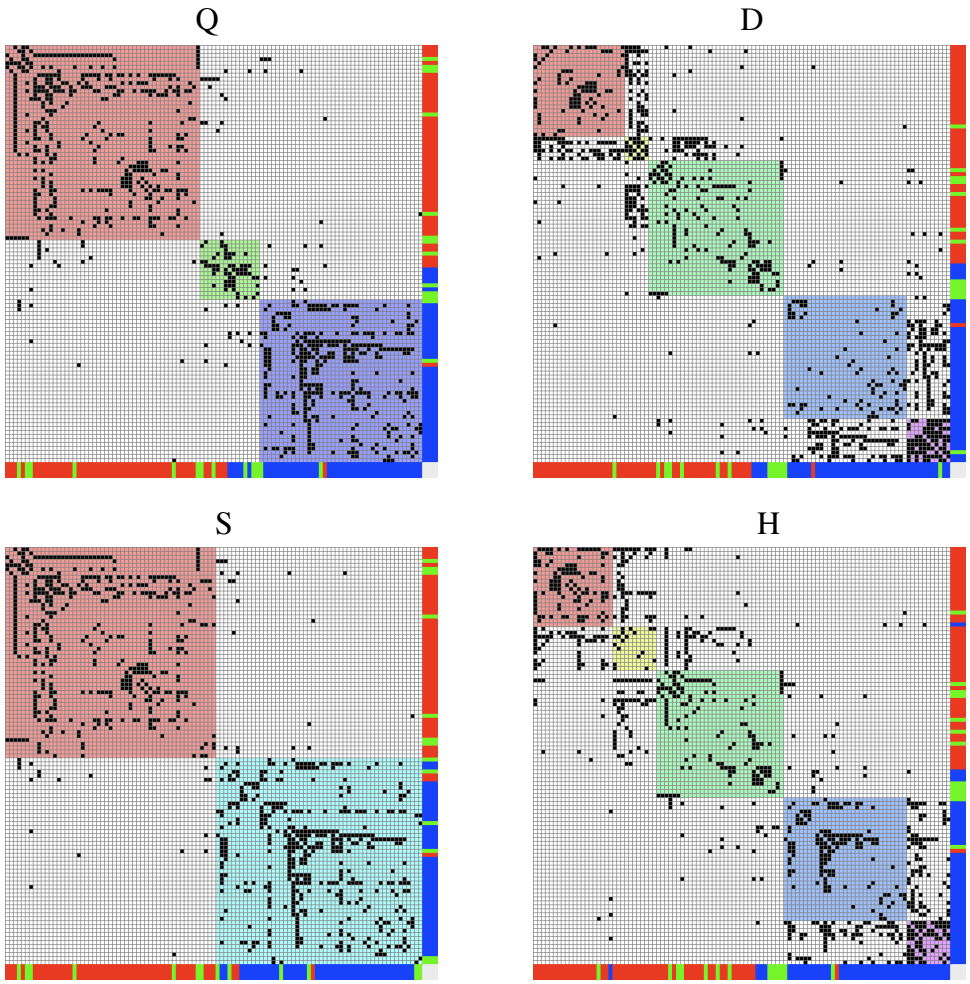

Figure 4.7. Ordered graphical adjacency matrices of Kreb's political book data that result from the use of each of the discussed quality functions. A black square in the ijth position indicates a link between nodes i and j. A white square indicates the absence of a link. The colored bars at the right and bottom of each pane represent the correct true group membership of nodes.

CHAPTER 5

A LIKELIHOOD APPROACH TO DETERMINING THE SIGNIFICANCE OF DETECTED CLUSTERS

Despite the large variety of algorithms, all clustering algorithms have one thing in common: Clustering algorithms detect clusters. Some algorithms — e.g., hierarchical clustering algorithms — provide no guidance to the user with regard to the actual number of clusters in the data. Other algorithms — e.g., k-means clustering — require the number of clusters as an input parameter to the algorithm. Various model selection procedures — e.g., Akaike (1974); Schwarz (1978) — have been developed that can be applied to this problem. Still other algorithms — e.g., 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. An equally important and related problem, though, is determining if the output of these clustering algorithms is simply the result of randomness or if it displays the actual structure of the data; that is, what if the actual number of clusters in the data is 1? This section develops a test for the significance of detected clusters that is based on the LRT statistic proposed as a quality function in Chapter 4.

The proposed test, the likelihood ratio cluster (LRC) test, is applicable to the output from any clustering algorithm. The only inputs to the test are the output

of the clustering algorithm — i.e., the proposed group membership assignments — the observed data set, and the distributional assumptions. Section 5.1 gives the derivation and asymptotic distribution of the test statistic under the null hypothesis that the most appropriate model for the data is a single partition consisting of all observations. This is significant because it provides a means of formal hypothesis testing and also a means of evaluating the quality of clustering algorithms (Chapter 6). Section 5.2 describes the application of the test to various simulated and real-world data sets (Section 5.2.2).

5.1 The Likelihood Ratio Cluster Test

The Likelihood Ratio Test (LRT) is a natural fit for this sort of problem. The 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 two models are nested; the more complex model includes more parameters, while the less complex model includes fewer parameters.

As a simple example, let $\mathbf{x_1}$ and $\mathbf{x_2}$ denote independent, simple random samples. Suppose the researcher believes that the population or populations from which these samples were drawn have the same shape and spread, but the researcher is unsure about the equality of their means. In the case that the means are equal, the two samples are drawn from identical populations; i.e., a simpler model is preferred. In the case that the two means are unequal, the more complex model is preferred.

Let $f_1(x;\theta_1)$ and $f_2(x;\theta_2)$ denote the density functions of observations from the first and second samples, respectively, where $f_1(x) \equiv f_2(x) \equiv f(x)$ in the simpler model, and θ_i denotes the unknown mean associated with the ith sample.

The likelihood function under the simpler model, then, is $L(\theta_0; \mathbf{x_1}, \mathbf{x_2}) = \prod_{ij} f(x_{ij})$ where θ_0 denotes the single unknown mean. Similarly, the likelihood function under the more complex model is $L(\theta_1, \theta_2; \mathbf{x_1}, \mathbf{x_2}) = \prod_i f_1(x_{1i}; \theta_1) \prod_j f_2(x_{2j}; \theta_2)$.

The likelihood ratio, then, is

$$\Lambda = \frac{L(\hat{\theta}_0; \mathbf{x_1}, \mathbf{x_2})}{L(\hat{\theta}_1, \hat{\theta}_2; \mathbf{x_1}, \mathbf{x_2})} \tag{5.1}$$

where $\hat{\theta}$ is the maximum likelihood estimator of θ. The LRT statistic, then, is

$$D = -2 \ln(\Lambda) \tag{5.2}$$

which asymptotically follows a χ^2 distribution with degrees of freedom equal to the difference in the number of parameters to be estimated between the two models; $df = 2 - 1 = 1$, in this case. For a review of the LRT 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, let $\mathbf{X}$ denote an $n \times 1$ vector of observations to be clustered. Let $\mathbf{Z}$ denote an $n \times 1$ group membership vector such that, independent of $\mathbf{X}$, z_i takes integer values between 1 and k with equal probability. In this way, z_i indicates the group membership of x_i, and this group membership is uniformly and randomly assigned.

The null hypothesis that the data are independently and identically distributed — let $f(x_i; \boldsymbol{\theta}_0, \mathbf{Z} = \mathbf{1})$ denote the density function of this distribution — 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 member-

ship. Let $f_i(x_{ij}; \boldsymbol{\theta}_i, \mathbf{Z} = \mathbf{z})$ denote the density function of the ith cluster, where x_{ij} denotes the jth observation in cluster i, $\boldsymbol{\theta}_0$ denotes the vector of parameters associated with the null hypothesis, and $\boldsymbol{\theta}_i$ denotes the vector of parameters associated with the ith cluster specified in the alternative hypothesis.

By assigning group membership independently of the sample, the likelihood ratio can be written

$$\Lambda = \frac{\prod_{i=1}^{n} f(x_i; \boldsymbol{\theta}_0, \mathbf{Z} = 1)}{\prod_{j=1}^{k} \prod_{i=1}^{n_j} f_j(x_{ij}; \boldsymbol{\theta}_j, \mathbf{Z} = \mathbf{z})} \tag{5.3}$$

where n_j denotes the number of observations in cluster j. As described above, it is well known that $D = -2 \ln \Lambda$ is asymptotically χ^2 with degrees of freedom equal to the difference in dimensionality between the two models.

Since the number of possible values of $\mathbf{Z}$ is finite for a fixed $\mathbf{X}$ — large, but finite nonetheless — the population of possible values of D is also finite for a fixed $\mathbf{X}$. Let $\mathcal{P}$ denote the population of all possible values of D, for a fixed $\mathbf{X}$. The most interesting member of $\mathcal{P}$ is D_{max}, where $D_{max} \geq D_i$ for all possible values of i, $i = 1, 2, \ldots, S(n, k)$, where $S(n, k)$ denotes the Stirling number of the second kind; i.e., the number of ways to partition n elements into k subsets. Explicitly, $S(n, k) = \frac{1}{k!} \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} j^n$.

The practitioner could simply generate random values of Z in order to stumble upon D_{max}. As the number of generated values of Z increases, the probability of having selected the correct group membership vector, and hence D_{max}, increases. This approach is obviously problematic, however, for even moderately sized data sets. For example, suppose the practitioner suggests that a sample of size $n = 100$ forms $k = 5$ clusters. The number of possible partitions is $S(100, 5) = 6.57 \times 10^{67}$. Sampling enough times to have even a chance at selecting D_{max} is clearly infeasible.

Nonetheless, there is hope for producing an estimate for D_{max}. Let the vector $\mathbf{D}$ denote a sample *with replacement* from $\mathcal{P}$. If D_{max} is included in this sample, then the sample maximum, $D_{(N)} = D_{max}$; if not, $D_{(N)} < D_{max}$. 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 — i.e., the size of the sample — 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 the probability that D_{max} has been selected at least once after M draws is $P(D_{max} \in \mathbf{D}) = 1 - \left(1 - \frac{1}{S(n,k)}\right)^M$. It is important to note that this estimate is biased; that is, $D_{(N)} \leq D_{max}$, with equality achieved when $D_{max} \in \mathbf{D}$.

Since the asymptotic distribution of D_i is known, actually simulating values of $\mathbf{Z}$ is unnecessary; that is, the *pdf* of the maximum of a sample of size N from a χ^2 distribution is

$$f(D_{(N)}) = N[F_Y(D_{(N)})]^{N-1} f_Y(D_{(N)}) \tag{5.4}$$

where $F_Y(y)$ and $f_Y(y)$ are the *cdf* and *pdf* of the χ^2 distribution, respectively. Because the support of the χ^2 distribution is unbounded on the right, increasing values of N produce increasing values of $D_{(N)}$. By choosing an appropriate value for N a (downwardly) biased estimate of the distribution of D_{max} under the null hypothesis is obtained. Recalling that $E(M) = S(n,k) - 1$ is the expected number of draws required to select the maximum at random, set $N = E(M)$.

Hence, the $100(1-\alpha)th$ percentiles of the distribution of $D_{(N)}$ provide critical values for a one-tailed test for the significance of detected clusters (see Appendix C). The size of the test, however, may be somewhat less than *alpha* in the case

that the clustering algorithm used fails to detect the partition of the data set into k clusters that maximizes the test statistic. The size and power of the test for various algorithms will be described in Section 5.2.1.

Based on Equation (5.4), the critical values are calculated as

$$D_{(N),1-\alpha} = F_Y^{-1}\left(\sqrt[N]{1-\alpha}\right)$$

In most cases, $(\sqrt[N]{1-\alpha})$ is exceedingly close to 1, often too close to be easily computed. In fact, N is often so large that calculating it is infeasible. A logarithmic transformation, too, is not useful in making the problem any more computationally tractable. The critical values displayed in Appendix C were calculated using Maple (MapleSoft, 2009), 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).

These values and others were used to fit a model to predict the $(1-\alpha)$th quantile of the distribution of $D_{(N)}$. The number of degrees of freedom (r), the logarithm of the sample size, $\ln N$, and α produced a very predictive model with $R^2 \approx 1$ (Table 5.1). The fitted model is

$$\hat{C} = -2.755 + 3.945r + 2.027\ln(N) - 81.58\alpha \tag{5.5}$$

where $\hat{C}$ is the predicted critical value. If the number of observations in the original sample is too large to make the direct computation of the critical values feasible, Equation 5.5 provides a reliable approximation. Table 5.2 compares a few large sample cases' calculated values with approximations made in this way.

TABLE 5.1

Coefficients and their significance in the model of the $(1-\alpha)$th quantile of the distribution of $D_{(N)}$.

Predictor	**Coefficient**	**Standard Error**	**t**	**p**
Constant	-2.755	1.112	-2.48	0.013
r	3.94457	0.01107	356.44	0.000
$\ln N$	2.02706	0.00075	2686.14	0.000
α	-81.58	11.71	-6.97	0.000

5.2 Results

The data sets in this section are clustered by means of two algorithms. Conventional data sets are clustered with k-means clustering and network data sets are clustered by Newman and Girvan's edge betweenness community detection algorithm as implemented in R (R Development Core Team, 2009). These results, then, must be viewed in that light; that is, the power of the test is directly related to the ability of the algorithm in use to detect the partition of the data set which produces the largest value of the LRC test statistic. Insofar as the algorithm does not find this partition, the power of the test will be reduced. As will be seen in Chapter 6, there is great variability in the quality of common clustering algorithms. To obtain an unbiased description of the power of the LRC test, would require a brute force detection algorithm; i.e., one that is guaranteed to detect the desired partition. As it is, all heuristic algorithms necessarily produce an underestimate of the power of the test in that they fail to detect the true maximum

TABLE 5.2

A comparison of several approximate and exact critical values for the LRC test.

n	df	k	α	**Exact**	**Lower PI**	**Approximate**	**Upper PI**
2000	7	48	0.001	8059.0676	8024.1531	8058.1624	8092.1716
2000	12	7	0.05	7845.7040	7878.1327	7912.1604	7946.1880
2000	44	9	0.05	9031.9298	9014.4835	9048.5748	9082.6661
2500	2	2	0.1	3468.8504	3474.3632	3508.1928	3542.0225
2500	6	3	0.1	5524.2889	5542.5956	5576.4961	5610.3965
3000	8	3	0.01	6642.3573	6671.2569	6705.2020	6739.1472

LRC test statistic.

5.2.1 Size and Power

As mentioned above, the practical size of the LRC test may be somewhat less than α due to a particular algorithm's failure to detect the partition of the data into k clusters that maximizes the test statistic. This decrease in power is algorithm specific and will vary according to the algorithm that is being used. While this results in a decreased probability of committing a Type I error, it needs to be shown that this failing does not lead to an unacceptable increase in the probability of committing a Type II error.

To show this, two types of data will be simulated. In both cases, let $\alpha = 0.05$. First, two samples of size 100 will be simulated from normal distributions with equal variance. The mean of the second group will be increased in stepwise fashion

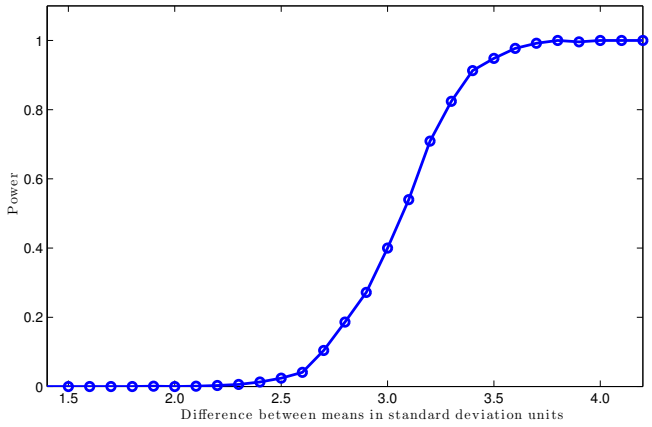

Figure 5.1. Simulated power of the LRC test on two independent normally distributed samples that were clustered via k-means clustering. Each point represents 1000 sets of samples.

to generate a power curve. Each step will include the output from 1000 simulated samples. The LRC test allows any clustering algorithm to be used. In this case, the k-means clustering algorithm described by Hartigan and Wong (1979) is used to cluster the data.

For the second simulation, the four-groups test described in Section 4.2.1.1 will be modified. In this case, the simulated networks will consist of 100 nodes divided into three groups of sizes 33, 33, and 34. Let p_{in} denote the probability that a node connects to another node within its group, and let p_{out} denote the probability that a node connects to a node outside of its group. The power curve will be generated by increasing the values of p_{in} in stepwise fashion while holding the expected degree — in this case set to 20 — constant. Each step will include the output from 200 simulated networks. The networks are clustered via the algorithm described by Newman and Girvan (2004).

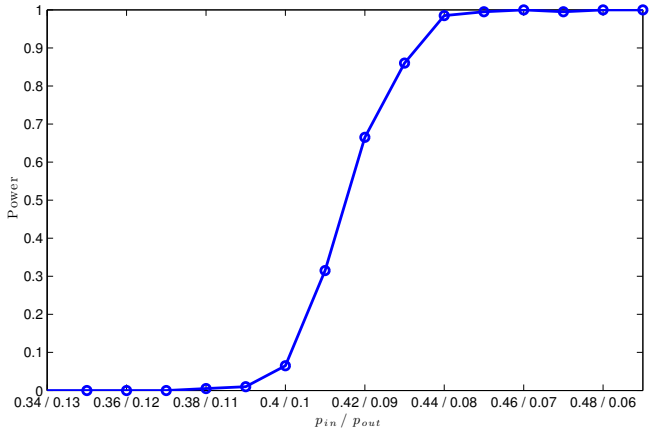

Figure 5.2. Simulated power of the LRC test on simulated, directed, binary networks with three equally sized groups. Networks were partitioned with the algorithm of Newman and Girvan (2004) with p_{in} indicating the probability of within group connections and p_{out} indicating the probability of between group connections. Each point represents 200 simulated, clustered networks.

As expected, the empirical size of the test is less than α due to the ineffectiveness of the algorithms to detect the optimal partition. In both cases, the simulated size of the test is close to 0. This indicates that neither of the algorithms used consistently detects the optimum partition of the data when the samples are from identical populations. In the first example, the power of the LRC is 0.8 when the means of the two distributions are approximately 3 standard deviations apart (Figure 5.1). In the second example, the LRC test had simulated power of 0.8 when $p_{in} \approx 0.42$ and $p_{out} \approx 0.09$ (Figure 5.2).

5.2.2 Network Data

While previous sections have proposed different and arguably more versatile and informative approaches, the LRC test can be applied to the results of any clustering algorithm. In order to demonstrate the usefulness of this test, previously detected clusters and clusters resulting from the most familiar algorithms will be tested for significance. Unless otherwise noted, all clusters are detected via the algorithm proposed in Newman and Girvan (2004) because of its ubiquity and familiarity.

TABLE 5.3

Evaluation of the significance of detected clusters in several well studied networks.

	data set	n	k	D	p
(a)	Lusseau's Dolphin Network, Arenas et al	62	2	159.1366	< 0.001
(b)	Lusseau's Dolphin Network, Newman and Girvan	62	2	131.7061	< 0.001
(c)	Kreb's Political Books, three groups	105	3	174.9011	> 0.1
(d)	Kreb's Political Books, two groups	105	2	161.7061	< 0.001

Lusseau et al. (2003) made a study of 62 dolphins living off Doubtful Sound in New Zealand. Both Newman and Girvan (2004) and Arenas et al. (2008) studied the community structure of this network. Arenas proposed a modification

of modularity that allows the practitioner to adjust the resolution of the search method and consequently produce a different partition of the node set. Table 5.3 shows that Arenas' clustering of the network is in fact more significant than Newman and Girvan's by a slight amount.

Kreb's unpublished data set of political blogs has also been closely analyzed. The books were classified by Newman (2006) according to their ideology; i.e., conservative, neutral, and liberal. Consequently, community detection algorithms have attempted to cluster the network into three or more groups. Table 5.3, however, shows that the partition produced by the Newman and Girvan algorithm does not produce significant results, nor does the algorithm produce significant results for four or five groups (not shown). Clustering the network into two groups, however, produces a significant partition.

5.3 Conclusions

The LRC test is unique in that it allows the practitioner to have an idea of the amount of community structure — as measured by the LRT statistic — that might occur as a result of randomness in data that does not exhibit any cluster-like structure. This provides the ability to perform formal hypothesis testing for the significance of detected clusters. This approach is applicable to both traditional and network data, though this work primarily focuses on network data.

By means of simulation, it has been shown that the LRC test likely has a much lower probability of Type I error than specified due to the limitations of the search algorithm employed. This decrease, however, does not appear to decrease the power of the test so as to limit its usefulness. The test was still able to detect clusters at acceptable levels. It is important to note — and this will be shown in

Chapter 6 — that this decrease in power and size is due to the sub-optimality of the algorithm being used and not due to a lack of power by the LRC test. By adjusting the value of N in the test statistic, perhaps an adjustment for this lack of power can be determined; that is, by lowering the value of N, the LRC test can be adjusted to have lower critical values that take into account the limitations imposed upon the test by the algorithm being used. Further research is needed to develop this tuning variable.

Application of the LRC test to existing data sets and known results provided additional insight into those results. In some cases the test showed that modifications to modularity, did, in fact, produce more significant results (Table 5.3a and b). In other cases, the test provided a means to specify the appropriate number of groups in the data set (Table 5.3c and d).

In addition to allowing the practitioner to test for the significance of detected clusters, this approach allows not only for the comparison of clustering algorithms with one another, but also for the objective evaluation of a clustering algorithm *in isolation*. Because the asymptotic distribution of the test statistic under the null hypothesis is known, the level of community structure that occurs as a result of randomness can be described without knowing the optimal partition of the data set; that is, by means of simulation, the quality of any clustering algorithm can be determined by the magnitude of the difference between the observed LRC test statistic and the expected LRC test statistic under the null hypothesis in random graphs. The following chapter demonstrates this new procedure on several network partitioning algorithms.

CHAPTER 6

EVALUATING THE EFFECTIVENESS OF CLUSTERING ALGORITHMS

Since detecting the optimal partition of a network is not feasible for networks of even moderate size — i.e., every partition cannot be evaluated — there has been a proliferation of clustering algorithms. Unfortunately, a reliable method for comparing the quality of these clustering algorithms has not been forthcoming (Fortunato, 2009). The benchmarks that have been developed for comparing network clustering algorithms consist of inputting a network of known structure into multiple clustering algorithms to determine how quickly, reliably, and accurately those algorithms partition those networks.

This is not a particularly satisfactory solution for two reasons: First, each evaluation depends heavily on the structure of the graph that is being supplied. For instance, a given algorithm might perform well on networks with hierarchical structure but poorly on small-world graphs or vice versa. Second, the approach does not provide an objective standard by which to measure a single approach and is mainly useful in comparing the effectiveness of two search algorithms. The test described in Chapter 5, however, provide a means of evaluating the quality of a clustering algorithm without requiring a comparison and without *a priori* knowledge the structure of the input network. This chapter describes a methodology for evaluating the effectiveness of a particular algorithm at detecting the unknown optimum partition of a network under the null hypothesis.

6.1 A General Definition of Community Structure

No precise definition of community structure has been agreed upon by the many researchers from the different disciplines that have studied community detection in networks such as in Ling (1972, 1973) which defined a cluster in conventional clustering approaches. This difficulty is compounded by results that demonstrate that even simulated networks into which no community structure is intentionally designed can be partitioned such that they demonstrate at least some level of structure. The test developed in Chapter 5 largely solves this problem. In other words, it is not unexpected to be able to detect "cliques" within social networks, regardless of the process by which they developed. The expected amount of structure that results from randomness can, in fact, now be quantified.

Hence, this work proposes the following definition of community structure:

Definition 6.1. *The community structure of a network is the way in which and the extent to which some nodes connect differently than other nodes in a network.*

In other words, a network which possesses no community structure whatsoever would be arranged such that every node has exactly same degree and exactly the same neighbors as every other node; that is, a network without community structure is either completely saturated or completely empty. Of course, theses trivial cases are not of interest. Due to randomness, nodes that have similar theoretical behavior — i.e., nodes in the same subgroup — may exhibit different observed behavior.

This definition is much more general than others that have been proposed. Consider, for example, the graph displayed in Figure 6.1. This network, under many definitions of network structure, would not be considered to have community structure. It is clear, however, that nodes 1 and 3 behave much differently than the

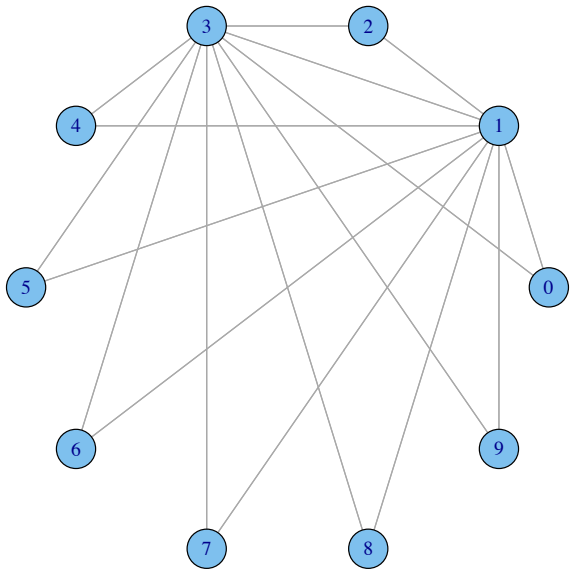

Figure 6.1. Even though some definitions of community structure would exclude the above graph, nodes 1 and 3 clearly behave differently than the other nodes in the network. They are connected to every other node in the network while the other nodes only connect to them.

other nodes in the network. In this network every node connects to nodes 1 and 3. No other connections exist. Failing to recognize this and other non-modular structures indicates a definition of network structure that is too restrictive.

This work proposes, then, that any difference in the way that nodes connect to the rest of the network be described as network structure. The LRC test can then be used to evaluate and quantify the significance of that structure.

6.2 Methods

Since heuristic clustering algorithms provide only approximately optimal partitions of a network — i.e., possibly sub-optimal partitions — it is of great interest

to the practitioner to determine how close a particular algorithm tends to be to the true optimum. The true optimum, though, is not known and cannot be easily determined in most cases. The results of Chapter 5, however, provide the asymptotic distribution of the LRC test statistic under the null hypothesis. This result suggests a direct method for evaluating the effectiveness of a given clustering algorithm.

Let f denote an arbitrary clustering algorithm and $\mathcal{G}$ denote an ER random graph. Let $f(\mathcal{G}) \to \hat{\mathcal{P}}$ denote the partitioning of $\mathcal{G}$ by f, where $\hat{\mathcal{P}}$ is the resulting partition. Let $D_{\mathcal{P}_i}$ denote the value of the LRC test statistic that results from the partition $\mathcal{P}_i$, and let $\mathcal{P}$ denote the true optimal partition; i.e., the partition of $\mathcal{G}$ such that $D_{\mathcal{P}} \geq D_{\mathcal{P}_i}$ for all i.

For randomly generated realizations of $\mathcal{G}$, the asymptotic distribution of $D_{\mathcal{P}}$ is approximated by Equation (5.4). The distribution of $D_{\hat{\mathcal{P}}}$, however, depends on f and is not currently well-known for any existing clustering algorithm, though since $D_{\mathcal{P}} \geq D_{\hat{\mathcal{P}}}$, the distribution is, at the least, shifted down. The magnitude of that shift is of interest; i.e., the larger the shift, the less adequate is f.

Because a multitude of search algorithms exist, the algorithms chosen in this study is somewhat arbitrary. Three algorithms were chosen for analysis in this section based on their common usage and on their capability to allow the user to specify the number of groups. The clustering algorithms studied are:

1. The Fast-Greedy (FG) algorithm of Clauset et al. (2004).

2. The Walktrap (WT) algorithm of Pons and Latapy (2005).

3. The Edge-Betweenness (EB) algorithm of Newman and Girvan (2004).

The reader will note that, unlike Section 4, this section does not seek to evaluate

the effectiveness of maximizing a particular quality function. The comparison, rather, is *between* algorithms.

Let $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_N$ denote a random sample of simulated ER random graphs with constant probability of a tie, p. Each algorithm, $f_{EB}(\mathcal{G}_i) \to \hat{\mathcal{P}}_{EB,i}$, $f_{FG}(\mathcal{G}_i) \to \hat{\mathcal{P}}_{FG,i}$, and $f_{WT}(\mathcal{G}_i) \to \hat{\mathcal{P}}_{WT,i}$ is used to calculate the random variables $D_{\hat{\mathcal{P}}_{EB,i}}$, $D_{\hat{\mathcal{P}}_{FG,i}}$, and $D_{\hat{\mathcal{P}}_{WT,i}}$. The distribution of these random variables will be compared to the approximate distribution of $D_{\mathcal{P}}$ graphically and based on two metrics: First, let $\Delta_j = E(D_{\mathcal{P}}) - \bar{D}_{\hat{\mathcal{P}}_j}$. Where $\bar{D}_{\hat{\mathcal{P}}_j}$ is the sample average of the LRC test statistics from algorithm j. Second, the sample variance of the LRC test statistic from algorithm j, s_j, will be compared to the approximate variance of $D_{\mathcal{P}}$, σ. The theoretical values were calculated using Equation (5.4) by means of numerical integration.

A brute force search algorithm that searches every possible partition would, theoretically, have $E(\Delta) = 0$ and $E(s) = \sigma$. Heuristic algorithms will have $E(\Delta) \geq 0$ with smaller values of Δ indicating better algorithms. In this simulation, $N = 1000$ ER random graphs of size $n = 100$ and $p = 0.12$ were generated. Each graph was partitioned into two groups using each of the algorithms being evaluated.

6.3 Results

The results of the evaluation are striking. Table 6.1 shows that the performance of the algorithms is markedly different. The FG algorithm, which is often said to be useful more for speed than for accuracy, performed the best of all the algorithms tested. The WT algorithm was more variable than the FG algorithm and did not perform as well. Surprisingly, one of the most popular clustering techniques in use,

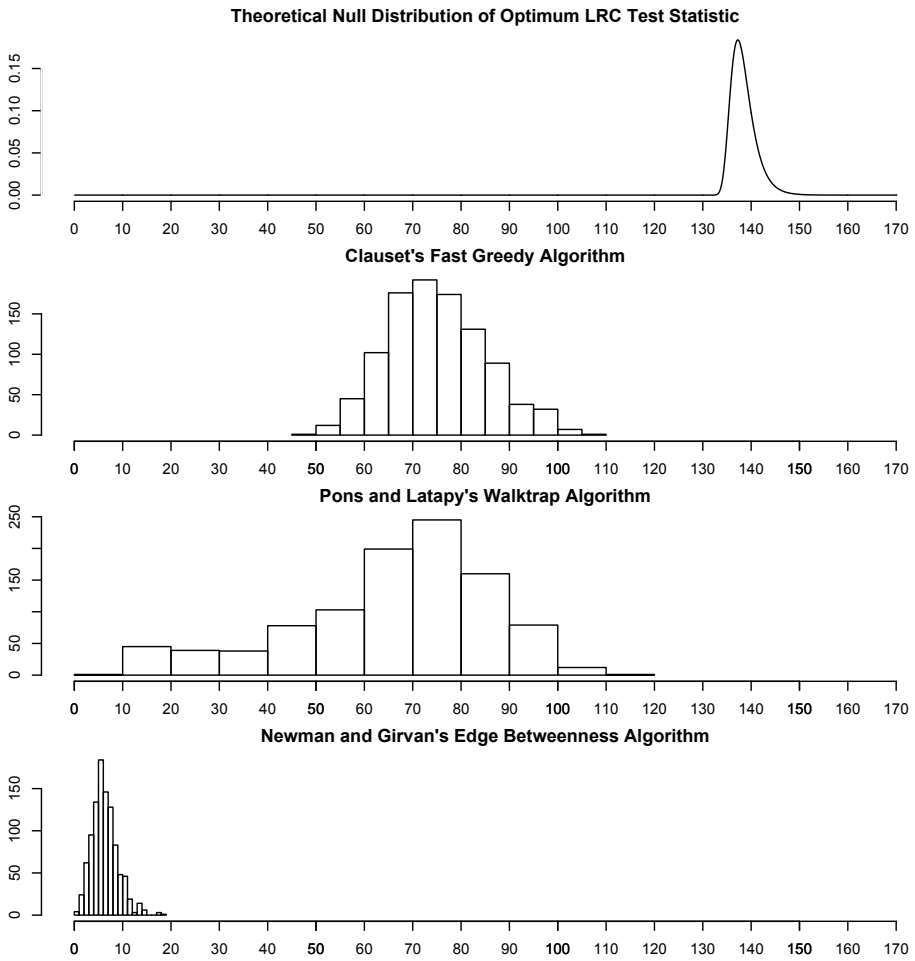

Figure 6.2. The distribution of test statistics resulting from three common clustering algorithms when applied to 1000 simulated ER random graphs. The theoretical distribution of the test statistic is shown in the first plot.

the EB algorithm performed abysmally with an average test statistic value of 23.22 and $\Delta = 132.0$. The FG and WT algorithms also had much higher variability than the theoretical test statistic. The shape of the distribution of the EB test statistic was essentially the same as that of the theoretical test statistic, but shifted down. Figure 6.2 shows the theoretical distribution of the test statistic in the top pane, followed by histograms of the simulated values of the test statistics for each of the four algorithms being studied.

TABLE 6.1

Performance of various algorithms as compared to theoretical results.

Algorithm	Mean	St. Dev.	95% CI	Δ
Theoretical values	138.30	2.56	-	-
Fast-Greedy Clustering (FG)	74.96	10.13	$(74.3, 75.6)$	63.34
Walktrap Clustering (WT)	65.95	20.73	$(64.5, 67.2)$	72.35
Edge-Betweenness Clustering (EB)	6.30	2.64	$(6.1, 6.5)$	132.0

6.4 Conclusions

The identification of the optimal partition of an ER random graph is not an easy task. The above results showed that even algorithms which have been

shown quite useful in identifying modular structure in various test networks — e.g., the EB algorithm — are not necessarily effective at identifying the optimal partition when the network has no particular community structure of interest. Conversely, clustering algorithms which are, in general, employed for speed rather than accuracy — e.g., the FG algorithm — can perform well at identifying these partitions.

Nonetheless, none of the clustering algorithms under consideration came close to detecting the optimal partition of the network even when they performed at their best; i.e., the highest test-statistics in the simulation were lower than the lowest reasonable theoretical values. Until this point, clustering algorithms could only be evaluated based on their performance on a limited number of test networks; e.g., Zachary's Karate Club; that is, the algorithms that have been developed to date have been optimized for their performance on these few data sets. These results show that continued research is necessary to develop heuristics that perform better in the general, rather than only in the specific case. Such evaluations are now possible.

The algorithms included in this analysis were chosen, in part, because they allow the user to specify the number of groups into which the network is partitioned. Other algorithms have stopping criteria included in the algorithm and were omitted from this analysis for that reason. This omission is due to the change in the number of degrees of freedom involved in calculating the test statistic when the number of groups change. The distribution of the maximum test statistic for an unknown number of groups (possibly with an upper bound) could likely be determined using mixture distributions. Further research is needed to explore this approach.

CHAPTER 7

CONCLUSIONS

Clustering algorithms and other unsupervised learning algorithms are important tools in the mining of information from data. Particularly in network data, effective clustering algorithms should provide key information about the structure of the network and function and behavior of the nodes within that network. Many approaches are plagued by overly restrictive definitions of structure and have, consequently, produced algorithms that are only effective in particular situations. The quality functions developed in Chapter 4 provide a new means of measuring the most general type of community structure. The nature of this structure is not dictated by the algorithm or by the form of the quality function, but rather by the data itself.

Chapter 5 proposes a statistical framework under which the results of any clustering algorithm can be evaluated by means of formal hypothesis testing. This section developed an asymptotic, approximate null distribution for the value of the quality function proposed in Chapter 4. Because of the vast number of ways to partition a network of even moderate size, even simulated networks with no inherent structure can be partitioned such that they demonstrate at least some level of structure. The LRC test allows the user determine whether the structure identified by the clustering algorithm in use exceeds that which might have occurred in an ER random graph due to randomness.

Another useful feature of the LRC test it that provides a means of evaluating the performance of any given clustering algorithm. Since the clustering algorithms are heuristic — i.e., not exact — clustering techniques, they are not guaranteed to detect the optimal partition. Chapter 6, in fact, shows that none of the algorithms studied can reliably detect the optimal partition of a network. Some algorithms performed poorly, while others were somewhat better. None, though, came close to the theoretical optimum. More important than the performance of any single clustering algorithm, though, is the new capability of comparing clustering results to the theoretical optimum, even when the specific optimal partition is not known.

Opportunities exist for further research exist in several areas: First, the quality functions developed in Chapter 4, while they generalize existing approaches, have not been extended for application to multi-modal networks or to networks in which nodes can belong to multiple groups. Secondly, the test developed in Chapter 5 is a first step in applying formal statistical methods to clustering algorithms. Additional research is necessary to determine the distribution of the test statistic when the number of groups is not specified in advance, perhaps by employing mixture distributions. Additionally, in order to account for the inability of a particular clustering algorithm, the value N in the calculation of the LRC test statistic can be treated as a tuning parameter in order to adjust the theoretical distribution of the test statistic to be in line with the, now known, performance of the algorithm in use. The size and type of adjustment is an object of future research.

Furthermore, the LRC test assumes that the links in the network are independent of one another. It is well-known that most networks violate this assumption. Additional work is needed to demonstrate the effectiveness and robustness of the

LRC test to networks which violate these assumptions.

Finally, the evaluations of clustering algorithms described in Chapter 6 should be extended to include more of the common clustering algorithms in use. Since these results show that heuristic algorithms likely do not detect the optimal partition, additional research should also focus on adjusting the parameters — the value of N in particular — to increase the power of the test and to increase the size of the test to the desired level.

APPENDIX A

GROUP ASSIGNMENTS FROM VARIOUS QUALITY FUNCTIONS: ZACHARY'S KARATE CLUB DATA SET

TABLE A.1

Group assignments resulting from the maximization of modularity on Zachary's Karate Club data set.

Group	Node
1	1, 5, 6, 7, 11, 12, 17, 20
2	2, 3, 4, 8, 10, 13, 14, 18, 22
3	9, 15, 16, 19, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34

TABLE A.2

Group assignments resulting from the maximization of the LRC test statistic on Zachary's Karate Club data set.

Group	Node
1	1
2	5, 6, 7, 11, 17
3	2, 3, 4, 8, 14
4	12, 13, 18, 20, 22
5	9, 15, 16, 19, 21, 23, 31
6	10, 24, 25, 26, 27, 28, 29, 30, 32
7	33, 34

TABLE A.3

Group assignments resulting from the maximization of Generalized Squared Modularity on Zachary's Karate Club data set.

Group	Node
1	1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 17, 18, 20, 22
2	9, 15, 16, 19, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34

TABLE A.4

Group assignments resulting from the maximization of Generalized Absolute Modularity on Zachary's Karate Club data set.

Group	Node
1	1
2	5, 6, 7, 11, 17
3	2, 3, 4, 8, 12, 13, 14, 18, 20, 22
4	9, 15, 16, 19, 21, 23, 31
5	10, 24, 25, 26, 27, 28, 29, 30, 32
6	33, 34

APPENDIX B

GROUP ASSIGNMENTS FROM VARIOUS QUALITY FUNCTIONS: KREB'S POLITICAL BOOKS DATA SET

TABLE B.1

Group assignments resulting from the maximization of modularity on Kreb's political books data set.

Group	Node
1	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, 29, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 53, 54, 55, 56, 105
2	48, 49, 50, 51, 52, 57, 58, 64, 65, 67, 68, 69, 85, 103, 104
3	28, 30, 31, 59, 60, 61, 62, 63, 66, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102

TABLE B.2

Group assignments resulting from the maximization of the LRC test statistic on Kreb's political books data set.

Group	Node
1	10, 13, 15, 16, 17, 19, 23, 27, 32, 33, 34, 35, 36, 37, 38, 39, 41, 42, 43, 44, 46, 54, 55
2	3, 8, 11, 12, 40, 47
3	1, 2, 4, 5, 6, 7, 9, 14, 18, 20, 21, 22, 24, 25, 26, 29, 45, 48, 49, 50, 51, 52, 53, 56, 57, 58, 64, 65, 67, 68, 69, 103, 104, 105
4	28, 59, 60, 61, 62, 63, 70, 77, 78, 79, 80, 81, 82, 83, 85, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102
5	30, 31, 66, 71, 72, 73, 74, 75, 76, 84, 86

TABLE B.3

Group assignments resulting from the maximization of Generalized Squared Modularity on Kreb's political books data set.

Group	Node
1	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, 29, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 53, 54, 55, 56, 57, 105
2	28, 30, 31, 51, 52, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104

TABLE B.4

Group assignments resulting from the maximization of Generalized Absolute Modularity on Kreb's political books data set.

Group	Node
1	8, 23, 24, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 46, 47, 54, 61
2	9, 10, 12, 13, 20, 21, 22, 25, 26, 27, 45
3	1, 2, 3, 4, 5, 6, 7, 11, 14, 15, 16, 17, 18, 19, 29, 48, 49, 50, 51, 52, 53, 55, 56, 57, 58, 64, 65, 68, 69, 103, 104, 105
4	28, 30, 31, 59, 60, 62, 63, 67, 70, 71, 72, 75, 76, 77, 78, 79, 80, 81, 82, 83, 85, 90, 91, 92, 94, 95, 96, 97, 98, 101, 102
5	66, 73, 74, 84, 86, 87, 88, 89, 93, 99, 100

APPENDIX C

TABLES OF CRITICAL VALUES FOR THE LIKELIHOOD RATIO CLUSTER TEST

TABLE C.1

Table of critical values for test of significance of k clusters using a single parameter distribution. $df = k - 1$

		$\boldsymbol{\alpha}$		
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 100$		
3	220.64	222.08	225.34	229.95
		$n = 200$		
3	440.36	441.8	445.06	449.68
5	650.28	651.72	654.99	659.62
7	788.35	789.79	793.07	797.71
9	890.81	892.26	895.54	900.19

(Continued on next page)

TABLE C.1 – continued from previous page

	α			
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 300$		
3	660.08	661.52	664.78	669.4
5	972.97	974.41	977.68	982.3
7	1179.13	1180.58	1183.85	1188.48
9	1332.67	1334.11	1337.39	1342.02
11	1454.6	1456.05	1459.33	1463.97
13	1555.37	1556.82	1560.1	1564.74
		$n = 400$		
3	879.81	881.25	884.51	889.12
5	1295.43	1296.87	1300.13	1304.75
7	1569.46	1570.9	1574.17	1578.8
9	1773.83	1775.27	1778.54	1783.17
11	1936.47	1937.91	1941.19	1945.82
13	2071.22	2072.66	2075.94	2080.58
15	2185.95	2187.4	2190.68	2195.32

(Continued on next page)

TABLE C.1 – continued from previous page

	α			
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 500$		
3	1099.53	1100.97	1104.23	1108.84
5	1617.76	1619.2	1622.46	1627.08
7	1959.53	1960.97	1964.24	1968.86
9	2214.6	2216.05	2219.31	2223.94
11	2417.82	2419.27	2422.54	2427.17
13	2586.43	2587.87	2591.15	2595.78
15	2730.23	2731.68	2734.95	2739.59
		$n = 600$		
3	1319.25	1320.69	1323.95	1328.57
5	1940.01	1941.45	1944.71	1949.33
7	2349.43	2350.88	2354.14	2358.76
9	2655.13	2656.58	2659.84	2664.47
11	2898.85	2900.29	2903.56	2908.19
13	3101.23	3102.68	3105.95	3110.58
15	3274.02	3275.46	3278.74	3283.37

(Continued on next page)

TABLE C.1 – continued from previous page

	α			
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 700$		
3	1538.97	1540.41	1543.67	1548.29
5	2262.2	2263.64	2266.91	2271.52
7	2739.23	2740.67	2743.94	2748.56
9	3095.5	3096.94	3100.21	3104.83
11	3379.66	3381.1	3384.37	3388.99
13	3615.76	3617.2	3620.47	3625.09
15	3817.47	3818.91	3822.18	3826.81
		$n = 800$		
3	1758.7	1760.14	1763.4	1768.01
5	2584.36	2585.8	2589.06	2593.68
7	3128.94	3130.39	3133.65	3138.27
9	3535.74	3537.18	3540.45	3545.07
11	3860.3	3861.74	3865.01	3869.63
13	4130.07	4131.52	4134.79	4139.41
15	4360.68	4362.12	4365.39	4370.02

(Continued on next page)

TABLE C.1 – continued from previous page

	α			
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 900$		
3	1978.42	1979.86	1983.12	1987.73
5	2906.48	2907.92	2911.18	2915.8
7	3518.6	3520.04	3523.3	3527.92
9	3975.89	3977.33	3980.6	3985.22
11	4340.82	4342.26	4345.52	4350.15
13	4644.24	4645.68	4648.95	4653.57
15	4903.69	4905.14	4908.41	4913.03
		$n = 1000$		
3	2198.14	2199.58	2202.84	2207.46
5	3228.58	3230.02	3233.28	3237.9
7	3908.2	3909.64	3912.9	3917.52
9	4415.96	4417.41	4420.67	4425.29
11	4821.23	4822.68	4825.94	4830.56
13	5158.28	5159.72	5162.98	5167.61
15	5446.56	5448.01	5451.27	5455.9

TABLE C.2

Table of critical values for test of significance of k clusters using a two parameter distribution. $df = 2k - 2$

		$\boldsymbol{\alpha}$		
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 100$		
2	141.74	143.18	146.44	151.06
3	230.15	231.6	234.89	239.54
4	294.01	295.47	298.77	303.45
		$n = 200$		
2	280.37	281.81	285.07	289.69
3	451.21	452.65	455.93	460.56
4	573.93	575.38	578.66	583.31
5	670.02	671.47	674.76	679.42
6	749.12	750.57	753.87	758.53
7	816.38	817.84	821.14	825.81
8	874.91	876.37	879.67	884.35
9	926.71	928.17	931.48	936.16

(Continued on next page)

TABLE C.2 – continued from previous page

		α		
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 300$		
2	419	420.44	423.7	428.32
3	671.72	673.17	676.44	681.07
4	852.77	854.21	857.49	862.12
5	994.27	995.72	999	1003.64
6	1110.61	1112.06	1115.35	1119.99
7	1209.49	1210.94	1214.22	1218.88
8	1295.5	1296.95	1300.24	1304.9
9	1371.63	1373.08	1376.38	1381.04
10	1439.92	1441.38	1444.68	1449.34
11	1501.84	1503.29	1506.59	1511.26
12	1558.45	1559.91	1563.21	1567.89
13	1610.6	1612.06	1615.36	1620.04
14	1658.91	1660.37	1663.67	1668.36

(Continued on next page)

TABLE C.2 – continued from previous page

		α		
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 400$		
2	557.63	559.07	562.33	566.95
3	892.01	893.46	896.72	901.35
4	1131.15	1132.6	1135.87	1140.5
5	1317.85	1319.29	1322.57	1327.2
6	1471.21	1472.66	1475.94	1480.57
7	1601.47	1602.92	1606.2	1610.84
8	1714.75	1716.2	1719.48	1724.13
9	1814.99	1816.44	1819.73	1824.38
10	1904.91	1906.36	1909.65	1914.3
11	1986.44	1987.9	1991.19	1995.84
12	2061.02	2062.47	2065.76	2070.42
13	2129.73	2131.18	2134.47	2139.14
14	2193.41	2194.87	2198.16	2202.83
15	2252.76	2254.21	2257.51	2262.18

(Continued on next page)

TABLE C.2 – continued from previous page

	α			
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 500$		
2	696.26	697.7	700.96	705.58
3	1112.17	1113.62	1116.88	1121.51
4	1409.29	1410.73	1414	1418.63
5	1641.05	1642.49	1645.76	1650.39
6	1831.31	1832.76	1836.03	1840.67
7	1992.84	1994.28	1997.56	2002.2
8	2133.25	2134.7	2137.98	2142.62
9	2257.49	2258.94	2262.22	2266.86
10	2368.91	2370.36	2373.65	2378.29
11	2469.94	2471.39	2474.67	2479.32
12	2562.35	2563.8	2567.09	2571.74
13	2647.5	2648.95	2652.24	2656.89
14	2726.44	2727.89	2731.18	2735.84
15	2800.02	2801.47	2804.76	2809.42

(Continued on next page)

TABLE C.2 – continued from previous page

	α			
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 600$		
2	834.89	836.33	839.59	844.2
3	1332.26	1333.7	1336.96	1341.59
4	1687.27	1688.71	1691.98	1696.6
5	1964.01	1965.46	1968.73	1973.35
6	2191.1	2192.54	2195.81	2200.44
7	2383.81	2385.25	2388.53	2393.16
8	2551.29	2552.73	2556.01	2560.64
9	2699.43	2700.88	2704.16	2708.8
10	2832.29	2833.74	2837.01	2841.65
11	2952.73	2954.18	2957.46	2962.1
12	3062.9	3064.35	3067.63	3072.27
13	3164.41	3165.86	3169.14	3173.79
14	3258.53	3259.98	3263.26	3267.91
15	3346.26	3347.71	3350.99	3355.64

(Continued on next page)

TABLE C.2 – continued from previous page

	α			
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 700$		
2	973.52	974.96	978.22	982.83
3	1552.29	1553.73	1556.99	1561.61
4	1965.14	1966.58	1969.84	1974.47
5	2286.81	2288.25	2291.52	2296.15
6	2550.66	2552.11	2555.38	2560.01
7	2774.51	2775.95	2779.22	2783.86
8	2968.99	2970.44	2973.71	2978.34
9	3141	3142.44	3145.72	3150.35
10	3295.22	3296.67	3299.95	3304.58
11	3435.03	3436.48	3439.76	3444.39
12	3562.9	3564.35	3567.63	3572.27
13	3680.72	3682.17	3685.45	3690.09
14	3789.96	3791.41	3794.69	3799.34
15	3891.79	3893.24	3896.52	3901.17

(Continued on next page)

TABLE C.2 – continued from previous page

	α			
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 800$		
2	1112.15	1113.59	1116.85	1121.46
3	1772.27	1773.71	1776.98	1781.6
4	2242.92	2244.36	2247.63	2252.25
5	2609.49	2610.93	2614.2	2618.83
6	2910.07	2911.51	2914.78	2919.41
7	3165.01	3166.45	3169.72	3174.35
8	3386.46	3387.9	3391.18	3395.81
9	3582.28	3583.73	3587	3591.63
10	3757.84	3759.29	3762.56	3767.2
11	3916.97	3918.42	3921.69	3926.33
12	4062.51	4063.95	4067.23	4071.87
13	4196.6	4198.04	4201.32	4205.96
14	4320.92	4322.37	4325.65	4330.29
15	4436.81	4438.26	4441.54	4446.18

(Continued on next page)

TABLE C.2 – continued from previous page

	α			
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 900$		
2	1250.78	1252.22	1255.48	1260.09
3	1992.23	1993.67	1996.93	2001.55
4	2520.65	2522.09	2525.35	2529.98
5	2932.08	2933.52	2936.79	2941.41
6	3269.35	3270.79	3274.06	3278.69
7	3555.35	3556.79	3560.06	3564.69
8	3803.74	3805.19	3808.46	3813.08
9	4023.35	4024.8	4028.07	4032.7
10	4220.22	4221.66	4224.93	4229.56
11	4398.64	4400.08	4403.36	4407.99
12	4561.8	4563.25	4566.53	4571.16
13	4712.13	4713.58	4716.86	4721.49
14	4851.51	4852.96	4856.23	4860.87
15	4981.43	4982.87	4986.15	4990.79

(Continued on next page)

TABLE C.2 – continued from previous page

	α			
k	**0.1**	**0.05**	**0.01**	**0.001**
	$n = 1000$			
2	1389.41	1390.85	1394.11	1398.72
3	2212.16	2213.6	2216.86	2221.48
4	2798.32	2799.77	2803.03	2807.65
5	3254.59	3256.03	3259.3	3263.92
6	3628.54	3629.98	3633.25	3637.87
7	3945.57	3947.02	3950.28	3954.91
8	4220.88	4222.32	4225.59	4230.22
9	4464.25	4465.7	4468.97	4473.6
10	4682.39	4683.84	4687.11	4691.74
11	4880.09	4881.53	4884.8	4889.43
12	5060.86	5062.31	5065.58	5070.21
13	5227.41	5228.85	5232.13	5236.76
14	5381.81	5383.26	5386.53	5391.17
15	5525.73	5527.18	5530.45	5535.09

TABLE C.3

Table of critical values for test of significance of k clusters using a three parameter distribution. $df = 3k - 3$

		$\boldsymbol{\alpha}$		
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 100$		
3	238.41	239.87	243.19	247.88
		$n = 200$		
3	460.75	462.2	465.49	470.15
5	687.56	689.02	692.32	697
7	841.31	842.78	846.1	850.81
9	958.62	960.09	963.43	968.15
		$n = 300$		
3	682.04	683.49	686.76	691.41
5	1013.31	1014.77	1018.06	1022.72
7	1236.65	1238.1	1241.41	1246.08
9	1406.48	1407.95	1411.26	1415.95
11	1544.06	1545.53	1548.85	1553.55
13	1659.92	1661.39	1664.72	1669.43

(Continued on next page)

TABLE C.3 – continued from previous page

	α			
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 400$		
3	902.88	904.33	907.6	912.23
5	1337.97	1339.42	1342.71	1347.36
7	1630.24	1631.7	1634.99	1639.65
9	1851.97	1853.43	1856.73	1861.4
11	2031.31	2032.77	2036.07	2040.75
13	2182.21	2183.67	2186.98	2191.67
15	2312.63	2314.09	2317.41	2322.1
		$n = 500$		
3	1123.47	1124.92	1128.19	1132.82
5	1662.03	1663.48	1666.76	1671.4
7	2022.87	2024.32	2027.61	2032.26
9	2296.14	2297.6	2300.89	2305.55
11	2516.88	2518.34	2521.64	2526.3
13	2702.46	2703.92	2707.22	2711.89
15	2862.77	2864.23	2867.53	2872.21

(Continued on next page)

TABLE C.3 – continued from previous page

	α			
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 600$		
3	1343.91	1345.36	1348.63	1353.25
5	1985.69	1987.14	1990.42	1995.05
7	2414.89	2416.34	2419.62	2424.26
9	2739.47	2740.92	2744.21	2748.86
11	3001.39	3002.84	3006.13	3010.79
13	3221.42	3222.87	3226.17	3230.83
15	3411.38	3412.84	3416.14	3420.81
		$n = 700$		
3	1564.24	1565.68	1568.95	1573.58
5	2309.09	2310.53	2313.81	2318.44
7	2806.47	2807.92	2811.2	2815.84
9	3182.21	3183.66	3186.94	3191.59
11	3485.15	3486.6	3489.89	3494.54
13	3739.47	3740.93	3744.22	3748.87
15	3958.94	3960.4	3963.69	3968.35

(Continued on next page)

TABLE C.3 – continued from previous page

		α		
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 800$		
3	1784.49	1785.93	1789.2	1793.82
5	2632.28	2633.73	2637	2641.63
7	3197.74	3199.19	3202.46	3207.1
9	3624.52	3625.96	3629.24	3633.89
11	3968.36	3969.81	3973.1	3977.74
13	4256.87	4258.32	4261.61	4266.26
15	4505.73	4507.18	4510.47	4515.12
		$n = 900$		
3	2004.68	2006.12	2009.39	2014.01
5	2955.33	2956.77	2960.04	2964.67
7	3588.77	3590.21	3593.49	3598.12
9	4066.49	4067.94	4071.22	4075.85
11	4451.16	4452.6	4455.88	4460.53
13	4773.75	4775.2	4778.49	4783.13
15	5051.91	5053.36	5056.65	5061.3

(Continued on next page)

TABLE C.3 – continued from previous page

		$\boldsymbol{\alpha}$		
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 1000$		
3	2224.82	2226.26	2229.52	2234.15
5	3278.25	3279.7	3282.97	3287.59
7	3979.6	3981.05	3984.32	3988.95
9	4508.2	4509.65	4512.93	4517.56
11	4933.61	4935.06	4938.34	4942.98
13	5290.23	5291.68	5294.96	5299.61
15	5597.62	5599.07	5602.36	5607

TABLE C.4

Table of critical values for test of significance of k clusters in an undirected network. $df = \frac{k(k+1)}{2} - 1$

		$\boldsymbol{\alpha}$		
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 100$		
2	141.74	143.18	146.44	151.06

(Continued on next page)

TABLE C.4 – continued from previous page

		α		
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 200$		
2	280.37	281.81	285.07	289.69
5	695.8	697.26	700.58	705.28
6	790.11	791.58	794.92	799.64
9	1029.34	1030.84	1034.24	1039.04
		$n = 300$		
2	419	420.44	423.7	428.32
5	1022.29	1023.75	1027.05	1031.72
6	1155.29	1156.75	1160.06	1164.75
9	1484.13	1485.61	1488.96	1493.71
10	1580.34	1581.83	1585.2	1589.97
13	1848.79	1850.3	1853.73	1858.57
14	1934.01	1935.53	1938.97	1943.84
		$n = 400$		
2	557.63	559.07	562.33	566.95
5	1347.49	1348.94	1352.23	1356.89
6	1518.55	1520.01	1523.31	1527.98
9	1934.69	1936.16	1939.5	1944.21
10	2054.49	2055.97	2059.31	2064.04
13	2384.14	2385.63	2389.02	2393.81
14	2487.47	2488.97	2492.37	2497.18

(Continued on next page)

TABLE C.4 – continued from previous page

	α			
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 500$		
2	696.26	697.7	700.96	705.58
5	1671.96	1673.41	1676.7	1681.34
6	1880.75	1882.2	1885.5	1890.15
9	2382.88	2384.34	2387.66	2392.36
10	2525.73	2527.2	2530.53	2535.24
13	2914.8	2916.28	2919.65	2924.4
14	3035.58	3037.07	3040.44	3045.21
		$n = 600$		
2	834.89	836.33	839.59	844.2
5	1995.97	1997.42	2000.7	2005.34
6	2242.26	2243.72	2247	2251.65
9	2829.53	2831	2834.3	2838.99
10	2995.11	2996.57	2999.89	3004.59
13	3442.42	3443.9	3447.24	3451.98
14	3580.21	3581.69	3585.05	3589.8

(Continued on next page)

TABLE C.4 – continued from previous page

	α			
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 700$		
2	973.52	974.96	978.22	982.83
5	2319.66	2321.11	2324.39	2329.03
6	2603.3	2604.75	2608.03	2612.68
9	3275.12	3276.58	3279.88	3284.55
10	3463.17	3464.63	3467.94	3472.63
13	3967.91	3969.38	3972.71	3977.43
14	4122.4	4123.87	4127.21	4131.95
		$n = 800$		
2	1112.15	1113.59	1116.85	1121.46
5	2643.12	2644.56	2647.84	2652.47
6	2963.99	2965.43	2968.71	2973.36
9	3719.91	3721.36	3724.66	3729.33
10	3930.26	3931.72	3935.03	3939.7
13	4491.8	4493.27	4496.6	4501.3
14	4662.77	4664.24	4667.57	4672.29

(Continued on next page)

TABLE C.4 – continued from previous page

		α		
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 900$		
2	1250.78	1252.22	1255.48	1260.09
5	2966.39	2967.83	2971.11	2975.74
6	3324.4	3325.85	3329.13	3333.77
9	4164.09	4165.54	4168.84	4173.5
10	4396.61	4398.07	4401.36	4406.03
13	5014.48	5015.94	5019.26	5023.96
14	5201.73	5203.2	5206.52	5211.23
		$n = 1000$		
2	1389.41	1390.85	1394.11	1398.72
5	3289.51	3290.96	3294.23	3298.86
6	3684.61	3686.05	3689.33	3693.96
9	4607.78	4609.24	4612.53	4617.18
10	4862.36	4863.81	4867.11	4871.77
13	5536.17	5537.64	5540.95	5545.64
14	5739.57	5741.04	5744.36	5749.05

TABLE C.5

Table of critical values for test of significance of k clusters in a directed network. $df = k^2 - 1$

		$\boldsymbol{\alpha}$		
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 100$		
3	245.98	247.45	250.79	255.52
		$n = 200$		
3	469.56	471.01	474.32	478.99
5	733.66	735.15	738.51	743.26
7	946.02	947.53	950.96	955.81
9	1139.6	1141.14	1144.64	1149.59
		$n = 300$		
3	691.59	693.05	696.33	700.99
5	1063.71	1065.18	1068.5	1073.21
7	1351.58	1353.07	1356.45	1361.22
9	1605.75	1607.26	1610.69	1615.53
11	1844.89	1846.42	1849.91	1854.83
13	2077.77	2079.33	2082.87	2087.89

(Continued on next page)

TABLE C.5 – continued from previous page

	α			
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 400$		
3	912.98	914.43	917.71	922.35
5	1391.49	1392.96	1396.27	1400.96
7	1752.7	1754.18	1757.53	1762.27
9	2064.78	2066.28	2069.67	2074.46
11	2353.15	2354.67	2358.1	2362.95
13	2629.83	2631.37	2634.85	2639.77
15	2901.35	2902.91	2906.44	2911.43
		$n = 500$		
3	1134	1135.45	1138.72	1143.36
5	1718.01	1719.47	1722.77	1727.45
7	2151.3	2152.77	2156.1	2160.82
9	2519.74	2521.23	2524.59	2529.36
11	2855.54	2857.05	2860.45	2865.26
13	3174.02	3175.54	3178.98	3183.85
15	3483.53	3485.07	3488.55	3493.47

(Continued on next page)

TABLE C.5 – continued from previous page

	α			
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 600$		
3	1354.79	1356.23	1359.51	1364.14
5	2043.71	2045.17	2048.47	2053.13
7	2548.27	2549.73	2553.05	2557.75
9	2972.06	2973.54	2976.88	2981.62
11	3354.1	3355.59	3358.97	3363.75
13	3713.03	3714.54	3717.95	3722.78
15	4059.06	4060.59	4064.04	4068.92
		$n = 700$		
3	1575.41	1576.86	1580.13	1584.76
5	2368.85	2370.3	2373.59	2378.25
7	2944.09	2945.55	2948.86	2953.55
9	3422.5	3423.98	3427.31	3432.03
11	3849.94	3851.42	3854.79	3859.55
13	4248.37	4249.87	4253.26	4258.06
15	4629.87	4631.38	4634.81	4639.65

(Continued on next page)

TABLE C.5 – continued from previous page

	α			
k	**0.1**	**0.05**	**0.01**	**0.001**
		$n = 800$		
3	1795.92	1797.37	1800.64	1805.27
5	2693.56	2695.01	2698.3	2702.95
7	3339.05	3340.51	3343.82	3348.5
9	3871.56	3873.03	3876.35	3881.06
11	4343.75	4345.23	4348.58	4353.32
13	4780.95	4782.45	4785.82	4790.6
15	5197.13	5198.63	5202.04	5206.86
		$n = 900$		
3	2016.34	2017.78	2021.05	2025.68
5	3017.95	3019.4	3022.68	3027.33
7	3733.37	3734.83	3738.13	3742.8
9	4319.54	4321.01	4324.33	4329.02
11	4835.99	4837.47	4840.81	4845.54
13	5311.4	5312.89	5316.25	5321.02
15	5761.62	5763.12	5766.51	5771.31

(Continued on next page)

TABLE C.5 – continued from previous page

	α			
k	**0.1**	**0.05**	**0.01**	**0.001**
	$n = 1000$			
3	2236.68	2238.12	2241.39	2246.02
5	3342.08	3343.53	3346.81	3351.45
7	4127.16	4128.62	4131.91	4136.58
9	4766.66	4768.13	4771.44	4776.13
11	5326.98	5328.45	5331.78	5336.5
13	5840.14	5841.62	5844.98	5849.73
15	6323.89	6325.39	6328.77	6333.55

BIBLIOGRAPHY

Abramowitz, M. and I. A. Stegun (Eds.) (1972). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Chapter 24.1.4, pp. 824–825. Dover.

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

Albert, R. and A.-L. Barabasi (2002, January). Statistical mechanics of complex networks. *Reviews of Modern Physics 74*, 47–97.

Albert, R., H. Jeong, and A.-L. Barabasi (1999, September). Internet: Diameter of the world-wide web. *Nature 401*, 130–131.

Arenas, A., A. Fernández, and S. Gómez (2008, May). Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics 10*, 053039.

Barabasi, A.-L. and R. Albert (1999). Emergence of scaling in random networks. *Science 286*, 509–512.

Barabasi, A.-L., R. Albert, and H. Jeong (1999). Mean-field theory for scale-free random networks. *Physica A 272*, 173–187.

Bollobas, B. (2001). *Random Graphs* (2nd ed.). Cambridge studies in advanced mathematics. Cambridge University Press.

Clauset, A. (2005). Finding local community structure in networks. *Physical Review E 72*(026132), 1–6.

Clauset, A., C. Moore, and M. E. J. Newman (2006). Structural inference on hierarchies in networks.

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

Costa, L. d. F. (2004). Hub-based community finding.

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

Doreian, P., V. Batagelj, and A. Ferligoj (2004). Generalized blockmodeling of two-mode network data. *Social Networks 26*, 29–53.

Doreian, P., V. Batagelj, and A. Ferligoj (2005). *Generalized Blockmodeling*. Structural Analysis in the Social Sciences. Cambridge University Press.

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

Eubank, S. (2005). Network based models of infectious disease spread. *Japanese Journal of Infectious Disease 58*(S9-S13).

Faloutsos, M., P. Faloutsos, and C. Faloutsos (1999). *Computational Communications 29*.

Felsenstein, J. (1985). Confidence limits on phylogenies: an approach using the bootstrap. *Evolution 39*, 783–791.

Flake, G., S. Lawrence, and C. Giles (2000). Efficient identification of web communities. *ACM Conference on Knowledge and Data Discovery (KDD 2000)*, 150–160.

Fortunato, S. (2009, 06). Community detection in graphs.

Fortunato, S. and M. Barthélemy (2007, January). Resolution limit in community detection. *Proceedings of the National Academy of Sciences of the United States of America 106*(1), 36–41.

Ganesh, A., L. Massoulie, and D. Towsley (2005). The effect of network topology on the spread of epidemics. *Proceedings - IEEE INFOCOM 2*, 1455–1466.

Gfeller, D., J.-C. Chappelier, and P. D. L. Rios (2005). Finding instabilities in the community structure of complex networks. *Physical Review E 72*(056135).

Girvan, M. and M. E. J. Newman (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.

Greene, D. and P. Cunningham (2006). Efficient prediction-based validation for document clustering. In *Proc. 17th European Conference on Machine Learning*.

Guimerà, R. and L. A. N. Amaral (2005, February). Cartography of complex networks: modules and universal roles. *Journal of Statistical Mechanics: Theory and Experiment* (P02001).

Guimerà, R., M. Sales-Pardo, and L. A. N. Amaral (2004). Modularity from fluctuations in random graphs and complex networks. *Physical Review E 70*(025101).

Handcock, M. S. and J. H. Jones (2004). Likelihood-based inference for stochastic models of sexual network formation. *Theoretical Population Biology 65*, 413–422.

Handcock, M. S., A. E. Raftery, and J. M. Tantrum (2007). Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A 170*, 301–354.

Hartigan, J. and M. Wong (1979). A k-means clustering algorithm. *Applied Statistics 28*, 100–108.

Hoff, P. D., A. E. Raftery, and M. S. Handcock (2002, December). Latent space approaches to social network analysis. *Journal of the American Statistical Association 97*(460), 1090–1098.

Hogg, R. V., J. W. McKean, and A. T. Craig (2005). *Introduction to Mathematical Statistics* (6th ed.). Upper Saddle River, New Jersey: Pearson Prentice Hall.

Hwang, W., T. Kim, M. Ramanathan, and A. Zhang. Bridging centrality: graph mining from element level to group level. In *Proceedings of the 14th ACM SIGKDD internation conference on Knowledge discovery and data mining*, pp. 336–344.

Ino, H., M. Kudo, and A. Nakamura (2005). Partitioning of web graphs by community topology. In *Proceedings of the 14th international conference on World Wide Web*, pp. 661–669.

Jonsson, P. F., T. Cavanna, D. Zicha, and P. A. Bates (2006). Cluster analysis of networks generated through homology: automatic identification of important protein communities involved in cancer metastasis. *BMC Bioinformatics 7*(2).

Karrer, B., E. Levina, and M. E. J. Newman (2008). Robustness of community structure in networks. *Physical Review E 77*(046119).

Kaufman, L. and P. Rousseeuw (2005). *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: Wiley.

Kernigan, B. W. and S. Lin (1970, February). An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*.

Kerr, M. and G. Churchill (2001). Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments. *Proceedings of the National Academy of Sciences of the United States of America 98*, 8961–8965.

Krivitsky, P. N., M. S. Handcock, A. E. Raftery, and P. D. Hoff (2009). Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Social Networks 31*, 204–213.

Kü cükpetek, S., F. Polat, and H. Ŏguztüzün (2005). Multilevel graph partitioning: An evolutionary approach. *The Journal of the Operational Research Society 56*(5), 549–562.

Leary, C., M. Schwehm, M. Eichner, and H. Duerr (2007). Tuning degree distributions: Departing from scale-free networks. *Physica A 382*, 731–738.

Levine, E. and E. Domany (2001). Resampling method for unsupervised estimation of cluster validity. *Neural Computation 13*(11), 1299–1323.

Ling, R. F. (1972). On the theory and construction of k-clusters. *The Computer Journal 15*(4), 326–332.

Ling, R. F. (1973). A probability theory of cluster analysis. *Journal of the American Statistical Association 68*(341), 159–164.

Liu, Y., D. N. Hayes, A. Nobel, and J. Marron (2008, September). Statistical significance of clustering for high-dimension, low–sample size data. *Journal of the American Statistical Association 103*(483), 1281–1293.

Lusseau, D., K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson (2003). *Behavioral Ecology and Sociobiology 54*, 396–405.

MapleSoft (2009). Maple.

Massen, C. P. and J. P. K. Doye (2008). Thermodynamics of community structure.

McShane, L. M., M. D. Radmacher, B. Freidlin, R. Yu, M.-C. Li, and R. Simon (2002). Methods for assessing reproducibility of clustering patterns observed in analyses of microarray data. *Bioinformatics 18*(11), 1462–1469.

Milgram, S. (1967). The small world problem. *Psychology Today 2*, 60–67.

Newman, M. E. J. (2001). The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences of the United States of America 98*(2), 404–409.

Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review 45*(2), 167–256.

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

Newman, M. E. J. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America 103*(23), 8577–8582.

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

Newman, M. E. J. and E. A. Leicht (2007). Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences of the United States of America 104*(23), 9564–9569.

Newman, M. E. J., S. H. Strogatz, and D. J. Watts (2001). Random graphs with arbitrary degree distributions and their applications. *Physical Review E 64*(026118).

Palshikar, G. K. and M. M. Apte (2008). Collusion set detection using graph clustering. *Data Mining and Knowledge Discovery 16*, 135–164.

Papadopoulos, S., A. Skusa, A. Vakali, Y. Kompatsiaris, and N. Wagner (2009, 02). Bridge bounding: A local approach for efficient community discovery in complex networks.

Pons, P. and M. Latapy (2005). Computing communities in large networks using random walks. *Lecture Notes in Computer Science 3733*, 284–293.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery (2007). *Numerical Recipes: The Art of Scientific Computing* (3rd Edition ed.). Cambridge University Press.

R Development Core Team (2009). *R: A Language and Environment for Statistical*

Computing. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Radicchi, F., C. Castellano, F. Cecconi, V. Loreto, and D. Parisi (2004, March). Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America 101*, 2658–2663.

Reichardt (2004). Detecting fuzzy community structures in complex networks with a potts model. *Physical Review Letters 93*(21), 218701.

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

Reichardt, J. and S. Bornholdt (2007). Partitioning and modularity of graphs with arbitrary degree distribution. *Physical Review E 76*(015102).

Rosvall, M. and C. T. Bergstrom (2008, 12). Mapping change in large networks.

Ruan, X.-G., J.-L. Wang, and J.-G. Li (2006). A network partition algorithm for mining gene functional modules of colon cancer from dna microarray data. *Genomics, Proteomics & Bioinformatics 4*(4), 245–252.

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

Suzuki, R. and H. Shimodaira (2004). An application of multiscale bootstrap resampling to hierarchical clustering of microarray data: how accurate are these clusters? pp. P034. Fifteenth International Conference on Genome Informatics (GIW 2004).

Suzuki, R. and H. Shimodaira (2006). Pvclust: an r package for assessing the uncertainty in hierarchical clustering. *Bioinformatics 22*(12), 1540–1542.

Tibshirani, R., G. Walther, D. Botstein, and P. Brown (2001). Cluster validation by prediction strength. Technical report, Department of Statistics, Stanford University.

Žiberna, A. (2007a). *Generalized blockmodeling of valued networks*. Ph. D. thesis, University of Ljubljani.

Žiberna, A. (2007b). Generalized blockmodeling of valued networks. *Social Networks 29*, 105–126.

Watts, D. J. and S. H. Strogatz (1998). Collective dynamics of 'small-world' networks. *Nature 393*(6684), 440–442.

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

Zhang, Z., S. Zhou, T. Zou, and G. Chen (2008). Fractal scale-free networks resistant to disease spread. *Journal of Statistical Mechanics: Theory and Experiment 2008*(09), P09008 (11pp).