SANDIA REPORT

SAND2014-1105
Unlimited Release
Printed February 2014

Statistically Significant Relational Data Mining: LDRD report

Tanya Berger-Wolf, Jonathan W. Berry, Sanjukta Bhowmick, Emily Casleton, Mark Kaiser, Vitus J. Leung, Daniel J. Nordman, Cynthia A. Phillips, Ali Pinar, David G. Robinson, Alyson G. Wilson

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2014-1105
Unlimited Release
Printed February 2014

Statistically Significant Relational Data Mining: LDRD report

Jonathan W. Berry, Analytics
Vitus J. Leung, Analytics
Cynthia A. Phillips, Analytics
Ali Pinar, Quantitative Modeling & Analysis
David G. Robinson, Analytics P.O. Box 5800
Albuquerque, NM 87185

Tanya Berger-Wolf
Department of Computer Science
University of Illinois
851 S. Morgan (M/C 152)
Room 1136 SEO
Chicago, IL 60607-7053

Sanjukta Bhowmick
Computer Science Department
University of Nebraska, Omaha
PKI 1110 South 67th Street
Omaha, NE 68182

Emily Casleton
Mark Kaiser
Daniel J. Nordman
Department of Statistics
Iowa State University
Ames, Iowa 50011-1210

Alyson G. Wilson
NCSU Statistics Department
2311 Stinson Drive
Campus Box 8203
Raleigh, NC 27695-8203

Abstract

This report summarizes the work performed under the project "Statitically significant relational data mining." The goal of the project was to add more statistical rigor to the fairly ad hoc area of data mining on graphs. Our goal was to develop better algorithms and better ways to evaluate algorithm quality. We concetrated on algorithms for community detection, approximate pattern matching, and graph similarity measures. Approximate pattern matching involves finding an instance of a relatively small pattern, expressed with tolerance, in a large graph of data observed with uncertainty.

This report gathers the abstracts and references for the eight refereed publications that have appeared as part of this work. We then archive three pieces of research that have not yet been published. The first is theoretical and experimental evidence that a popular statistical measure for comparison of community assignments favors over-resolved communities over approximations to a ground truth. The second are statistically motivated methods for measuring the quality of an approximate match of a small pattern in a large graph. The third is a new probabilistic random graph model. Statisticians favor these models for graph analysis. The new local structure graph model overcomes some of the issues with popular models such as exponential random graph models and latent variable models.

Acknowledgment

This work was funded by the Sandia National Laboratories' Laboratory Directed Research and Development (LDRD) program in the Computer and Information Science investment area. The coauthors on this report are the base research team and coauthors on material that is not published elsewhere, and therefore is in this report. We thank the coauthors who published work with us from this project. These coauthors are listed in Section 2. We thank Diane Suh and Randy Brost of Sandia National Laboratories for their detailed explanations of the high-school pattern matching problem and their comments on drafts of the material in Section 4.1. Randy is the person who first suggested "unicorn farms" as an example of a pattern that would not have a lot of readily observable instances.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Graphs are an abstract mathematical tool for representing relationships between pairs of objects. They can be drawns with circles representing the objects and lines representing relationship, thus allowing visual and intutive representations, at least on sufficiently small graphs. Frequently "big data" applications involve relationship information that can be represented as a graph. These networks are so large that at the highest level, they look like a "hairball." There are many applications that can be framed as finding structures in a large, complex graph. A few examples from the literature include the structural connectivity of brain networks [123], [118]; the behaviors of dolphins as a social network [82]; the international relations between countries based on attributes such as alliances, commerce, conflict, and international institution membership [54], [8]; the spread of a disease within a population [43]; the effect of disasters on fiber-optic networks [85]; the communications between a cell of a terrorist networks [115]; the reliability of sampled data on a protein-protein interaction network [98]; and the inter-organizational network of collaborations between rescue and relief organizations in response to the September 11, 2001 attack [116]. In bioinformatics, comparative network analysis can provide new insights into biological systems [135, 126]. Observing how these graphs change over time can assist in identifying poor quality web searches [92], detecting an intruder in a cyber warehouse [24], or monitoring network performance [25]. A more recent application in the medical area is the suggestion that graphs developed from magnetic resonance imaging might assist in learning assessment, the modeling disease progression, or to predict the vulnerability of soldiers to post traumatic stress disorder [42, 69]. Two fields which have contributed a considerable amount to the analysis of networks and creation of example datasets are computer science and sociology with interest in understanding the connections within the internet and the dynamics of social networks. The combination of recent advances in computational ability which makes analysis of large networks possible and the emergence of large, freely-available, and interesting networks, such as the Internet, Facebook, or the Wikipedia, which has brought networks into more mainstream analysis for statisticians as well.

In many data mining applications, especially the search for "unexpected" or "significant" structure in graph-based social information, it is difficult to specify exactly what one is looking for. For example, given a graph where nodes represent people and edges represent relationships between people, there is general agreement that a community is a set of nodes more connected internally than to the rest of the graph. But no formal graph theoretic definition of connectedness seems to capture what a human perceives to be the correct communities in all cases. Most community detection algorithms combine approximate optimization of a metric with ad hoc statistical methods. To date there are no rigorous ways to determine whether an algorithm has succeeded in finding the

"correct" answer in a real data set, or even in specially constructed benchmarks. Similar questions arise in other graph-data-mining problems that search for specific patterns, or attempt to explain (dynamic) relationships.

This report summarizes research from a Sandia National Laboratories Laboratory-Directed Research and Development project entitled "Statistically significant relational data mining." This report contains details only for research that has not yet been published in a peer-reviewed venue. Section 2 summarizes the published work from this project. Our goal was to develop statistically rigorous methods for finding, understanding and testing the significance of graph properties and structures. We focused primarily on two broad classes of problems in graphs: finding structure within a single graph, such as finding communities, and comparing across graphs, such as finding subgraphs and comparing pairs or sets of graphs.

Although we were open to any relevant technique, our goal was to apply Bayesian statistical methods to graph analysis. At the highest level, Bayesian methods start with a prior distribution that captures a prior domain knowledge and experience. For example, researchers such as Tanya Berger-Wolf at the University of Illinois, Chicago conduct extensive field research on animal populations such as zebras. Her research team has extensive knowledge about zebra behavior which could and should be incorporated into a prior distribution for any Bayesian method to detect communities in graphs of zebras. Researchers then must create plausible random graph models, which describe ways a pattern of interest, such as a community might evolve from underlying distributions. Then researchers must find methods to compute conditional probabilities that characterize the most likely solutions given the structure of the input instance. These allow computation of a posterior distribution, which describes the likely patterns in a particular data set. This distribution may not be in closed form, but generally there is a Monte-Carlo-based method to provably sample, say, communities from the posterior distribution. Although the general methods are well known to statisticians, they require customization to the structure of each problem.

Example: The Challenges of Community Detection Bayesian methods are particularly appealing in community detection. To motivate our work, we briefly describe Baysian methods, and the difficulties of addressing graph data mining questions in the context of community detection algorithms. Community detection algorithms accept a graph, which is a set of nodes (representing objects such as people) and links (representing relationships between the objects). The simplest community detection algorithm returns a partition of the nodes, where, as described above, each node is in some sense more closely tied to its group than to the rest of the graph. Real communities are hierarchical and overlapping. There are many recent papers that return communities with these more complex properties. However, we took a step back, going back to a basic partition problem to better address statistical foundations. Thus in our work for this project, our partitions associate each node with its primary community.

The literature for community detection is vast, appearing in computer science, physics, statistical, and social research areas. There are hundreds (or more) papers that treat community detection as an optimization problem, returning a graph that approximates an object objective such as conductance or modularity.

A problem with this optimization-based approach is, as mentioned above, there is no formal graph theoretic definition of connectedness seems to capture what a human perceives to be the correct communities in all cases. These objectives just represent plausible pressure. In a social network, even if there exists a ground truth to relationships, we must observe them somehow. In whatever interaction we use as a surrogate for the relationship, well likely miss some, from the inherent randomness in some interactions and from errors and incompleteness in the observation process.

Optimization-based algorithms that take topology and give communities implicitly assume topology is fundamentally influenced by communities. Bayesian methods make that assumption explicit. They start with a probability distribution that captures application-specific knowledge. The prior is a distribution on the communities. These distributions encode how community properties influence connections and what we might observe to infer relationships. In the zebra example, interactions depend upon gender and role in the herd. The Bayesian method is about transforming a generic communities of a zebra herd to a probability distribution of specific communities for a specific zebra herd. Given this distribution, we can make inferences, such as probability two nodes are in the same community. Giving a distribution of likely communities is far more informative and powerful than giving a point solution, which is what most community detection algorithms give.

This report does not give details for results that appear in refereed literature. We have published a paper [45] that gives a general Bayesian community detection methods. See Section 2 for an abstract of this paper.

Theres still no rigorous, universally accepted notion of community quality/correctness. In some cases, more notably in CS, people dont seem to care. Researchers attempting to parallelize the CNM community detection algorithm were concerned only with speed up, even though parallelization completely changed the communities. In our own attempts to parallelize our algorithm from [45], we attempted to compare communities from the parallel version to the communities from the serial version. The serial version had a provably correct method for generating communities from the posterier distribution. The parallel version only approximated that method. We determined that the standard way to compare communities, which relies on comparing vertex sets in the partition, has some important shortcomings. We briefly describe these issues in Section 3.

The other major graph data mining problem we consider is finding a pattern in a graph under uncertainty, or approximately comparing graphs. Graph similarity has numerous applications in multiple areas ranging from cyber security, understanding social networks, and predicting functions of protein networks. The computer science literature usually considers such pattern matching in the context of subgraph isomorphism. Given two (unlabeled) graphs, a small pattern graph P and a larger graph G, map the nodes of $P = (V_P, E_P)$ each to a unique node in $G = (V, E)$ using a function σ such that there is an $(u, v) \in E_P$ if and only if there is an edge $(\sigma(u), \sigma(v) \in E)$. That is, the edges of the induced graph in G exactly match the edges in the pattern P. Given uncertainty in the observations represented in graph G, it may be useful to relax the strict requirements of subgraph isomorphism. See Section 2 for paper that finds approximate matches according to a rigorous computer science definition (number of errors/mismatches).

The majority of our work in this area is developing candidate statistically-based methods for finding approximate matches and giving some measure of the matching quality. Section 4 describes two fundamentally different methods. The first is finds a beta distribution to measure the probability of a match between a candidate and a pattern. It is statistically justified, but requires difficult eliciation of information from experts. The second is more heuristic comparison of two graphs or a single graph into a library of potentially similar graphs. This method, based on the method of Macindoe and Richards [83] involves representing induced subgraphs as vectors of features, clustering the set of vectors from a given graph, and computing a statistically-justified distance between these this cluster-based representation. This method has worked well in practice on a Sandia application.

A final unpublished major contribution of this project is a new formal probabilistic random graph model which has some nice statistical properties not present in other popular probabilistic models such as expoential random graphs. We describe this in Section 5.

Definitions This section contains some basic definitions used later in the report.

A graph $G = (V,E)$ is composed of vertices and edges. V is the set of vertices (also called nodes) and $E \in V \times V$ is the set of edges of graph G. The order (or size) of a graph G is defined as the number of vertices of G and it is represented as $|V|$ and the number of edges as $|E|$.

If two vertices in G, say $u, v \in V$, are connected by an edge $e \in E$, this is denoted by $e = (u,v)$ and the two vertices are said to be adjacent or neighbors. Edges are said to be undirected when they have no direction, and a graph G containing only such types of graphs is considered *undirected*. When all edges have directions and therefore (u,v) and (v,u) can be distinguished, the graph is said to be *directed*. The term *arc* is commonly used when the graph is directed, and the term *edge* is used when it is undirected. In addition, a directed graph $G = (V,E)$ is considered *complete* when there is always an edge between any two vertices u, u' in the graph.

A subgraph H of a graph G is said to be *induced* if, for any pair of vertices u and v of H,(u,v) is an edge of H if and only if $(u,v)$is an edge of G. In other words, H is an *induced subgraph* of G if it has exactly the edges that appear in G over the same vertex set.

A *clique* in an undirected graph $G = (V,E)$ is a subset of the vertex set $C \subseteq V$, such that for every two vertices in C, there exists an edge connecting the two. If every two vertices in an induced subgraph are connected, the subgraph induced by C is complete.

If u and v are vertices, the distance from u to v, written $d(u,v)$, is the minimum length of any path from u to v. The eccentricity, $e(u)$, of the vertex u is the maximum value of $d(u,v)$, where v is allowed to range over all of the vertices of the graph. The radius of the graph G, $rad(G)$, is the minimum value of $e(u)$, for any vertex u, and the diameter, $diam(G)$, is the corresponding maximum value.

A *neighborhood* of a vertex $v \in V$ is an induced subgraph of G consisting of all vertices reachable from v by paths of length $\leq n$ and all edges connecting those vertices.

Chapter 2

Summary of Published Papers

This report is largely an archive of results supported by this LDRD that have not (yet) appeared. Thus this report does not contain details of material published in archival refereed conferences and journals. In this section, we include the abstract and references for each of these publications. In computer science, refereed conferences can have impact equal to or higher than some journals. Some of this work began under previous LDRD funding, but was finished and published under this project.

Community Detection Papers

Journal Papers

Jonathan Berry, Bruce Hendrickson, Randall LaViolette, and Cynthia Phillips published the paper "Tolerating the community detection resolution limit with edge weighting" in the journal *Physical Review E* in May 2011 [11].

Abstract: Communities of vertices within a giant network such as the World-Wide Web are likely to be vastly smaller than the network itself. However, Fortunato and Barthélemy have proved that modularity maximization algorithms for community detection may fail to resolve communities with fewer than $\sqrt{L/2}$ edges, where L is the number of edges in the entire network. This resolution limit leads modularity maximization algorithms to have notoriously poor accuracy on many real networks.

Fortunato and Barthélemy's argument can be extended to networks with weighted edges as well, and we derive this corollary argument. We conclude that weighted modularity algorithms may fail to resolve communities with fewer than $\sqrt{W\varepsilon/2}$ total edge weight, where W is the total edge weight in the network and ε is the maximum weight of an inter-community edge. If ε is small, then small communities can be resolved.

Given a weighted or unweighted network, we describe how to derive new edge weights in order to achieve a low ε, we modify the "CNM" community detection algorithm to maximize weighted modularity, and show that the resulting algorithm has greatly improved accuracy. In experiments with an emerging community standard

benchmark, we find that our simple CNM variant is competitive with the most accurate community detection methods yet proposed.

Jiqiang Guo, Daniel Nordman, and Alyson Wilson published "Bayesian Nonparametric methods for community detection," in the Journal *Technometrics* in November 2013 [45].

Abstract: We propose a series of Bayesian nonparametric statistical models for community detection in graphs. We model the probability of the presence or absence of edges within the graph. Using these models, we naturally incorporate uncertainty and variability and take advantage of nonparametric techniques such as the Chinese restaurant process and the Dirichlet process. Some of the contributions include: (1) the community structure is directly modeled without specifying the number of communities *a priori*; (2) the probabilities of edges within or between communities may be modeled as varying by community or pairs of communities; (3) some nodes can be classified as not belonging to any community; and (4) Bayesian model diagnostics are used to compare models and help with appropriate model selection. We start by fitting an initial model to a well-known network dataset, and we develop a series of increasingly complex models. We propose Markov chain Monte Carlo (MCMC) algorithms to carry out the estimation as well as an approach for community detection using the posterior distributions under a decision theoretic framework. Bayesian nonparametric techniques allow us to estimate the number and structure of communities from the data. To evaluate the proposed models for the example data set, we discuss model comparison using the deviance information criterion (DIC) and model checking using posterior predictive distributions. Supplementary materials are available online.

Matthew Rocklin and Ali Pinar published the paper "On Clustering on Graphs with Multiple Edge Types" in the Journal *Internet Mathematics* [107]. In this paper "clustering" means the same thing as community assignment, not a clustering of independent multi-dimensional vectors, typically addressed by methods such as k-means (see, for example, the scalable k-means++ algorithm in [6]).

Abstract: We study clustering on graphs with multiple edge types. Our main motivation is that similarities between objects can be measured in many different metrics. For instance similarity between two papers can be based on common authors, where they are published, keyword similarity, citations, etc. As such, graphs with multiple edges is a more accurate model to describe similarities between objects. Each edge/metric provides only partial information about the data; recovering full information requires aggregation of all the similarity metrics. Clustering becomes much more challenging in this context, since in addition to the difficulties of the traditional clustering problem, we have to deal with a space of clusterings. Reducing the multi-dimensional space into a single dimension poses significant challenges. At the same time, the multi-dimensional space can comprise latent structures, and searching this multi-dimensional space can reveal important information about the graph. We gen-

eralize the concept of clustering in single-edge graphs to multi-edged graphs and investigate problems such as: Can we find a clustering that remains good, even if we change the relative weights of metrics? How can we describe the space of clusterings efficiently? Can we find unexpected clusterings (a good clustering that is distant from all given clusterings)? If given the ground-truth clustering, can we recover how the weights for edge types were aggregated?

Refereed Conference Papers

Matthew Rocklin and Ali Pinar published a paper "Computing an Aggregate Edge-weight function for Clustering Graphs with Multiple Edge Types," in the Proceedings of 7th Workshop on Algorithms and Models for the Web Graph (WAW10) [108].

Abstract: We investigate the community detection problem on graphs in the existence of multiple edge types. Our main motivation is that similarity between objects can be defined by many different metrics and aggregation of these metrics into a single one poses several important challenges, such as recovering this aggregation function from ground-truth, investigating the space of different clusterings, etc. In this paper, we address how to find an aggregation function to generate a composite metric that best resonates with the ground-truth. We describe two approaches: solving an inverse problem where we try to find parameters that generate a graph whose clustering gives the ground-truth clustering, and choosing parameters to maximize the quality of the ground-truth clustering. We present experimental results on real and synthetic benchmarks.

They extended this work in a paper "Latent Clustering on Graphs with Multiple Edge Types," in the same conference the next year: Proceedings 8th Workshop on Algorithms and Models for the Web Graph (WAW11) [109].

Abstract: We study clustering on graphs with multiple edge types. Our main motivation is that similarities between objects can be measured in many different metrics, and so allowing graphs with multivariate edges significantly increases modeling power. In this context the clustering problem becomes more challenging. Each edge/metric provides only partial information about the data; recovering full information requires aggregation of all the similarity metrics. We generalize the concept of clustering in single-edge graphs to multi-edged graphs and discuss how this generates a space of clusterings. We describe a meta-clustering structure on this space and propose methods to compactly represent the meta-clustering structure. Experimental results on real and synthetic data are presented.

Jonathan Berry, Luke Fostvedt, Daniel Nordman, Cynthia Phillips, C. Seshadhri, and Alyson Wilson published the paper "Why Do Simple Algorithms for Triangle Enumeration Work in the

Real World?" at the Innovations in Theoretical Computer Science conference in 2014 (submitted in 2013) [12]. This is a paper on a general graph algorithm (triangle enumeration). Triangles are a key element of social networks and communities. The enumeration problem was motivated by the *Physical Review E* paper described above.

Abstract: Triangle enumeration is a fundamental graph operation. Despite the lack of provably efficient (linear, or slightly super-linear) worst-case algorithms for this problem, practitioners run simple, efficient heuristics to find all triangles in graphs with millions of vertices. How are these heuristics exploiting the structure of these special graphs to provide major speedups in running time?

We study one of the most prevalent algorithms used by practitioners. A trivial algorithm enumerates all paths of length 2, and checks if each such path is incident to a triangle. A good heuristic is to enumerate only those paths of length 2 where the middle vertex has the lowest degree. It is easily implemented and is empirically known to give remarkable speedups over the trivial algorithm.

We study the behavior of this algorithm over graphs with heavy-tailed degree distributions, a defining feature of real-world graphs. The erased configuration model (ECM) efficiently generates a graph with asymptotically (almost) any desired degree sequence. We show that the expected running time of this algorithm over the distribution of graphs created by the ECM is controlled by the $\ell_{4/3}$-norm of the degree sequence. As a corollary of our main theorem, we prove expected linear-time performance for degree sequences following a power law with exponent $\alpha \geq 7/3$, and non-trivial speedup whenever $\alpha \in (2,3)$.

Approximate Pattern Matching in Graphs Papers

Claire C. Ralph, Vitus J. Leung, and William McLendon III, published a paper "Brief announcement: subgraph isomorphism on a multithreaded shared memory architecture," in the 24th ACM Symposium on Parallelism in Algorithms and Architectures in 2012 [99]. These authors did considerable work on approximate subgraph isomorphism, where matches are allowed to deviate in an edge existence/non-existance in a small number of cases. This paper, however, considers parallel implemenation issues the classic exact subgraph isomorphism problem as a first step towards finding better approximate subgraph isomorphism implementations.

Abstract: Graph algorithms tend to suffer poor performance due to the irregularity of access patterns within general graph data structures, arising from poor data locality, which translates to high memory latency. The result is that advances in high-performance solutions for graph algorithms are most likely to come through advances in both architectures and algorithms. Specialized MMT shared memory machines offer a potentially transformative environment in which to approach the problem. Here, we explore the challenges of implementing Subgraph Isomorphism (SI) algorithms

based on the Ullmann and VF2 algorithms in the Cray XMT environment, where issues of memory contention, scheduling, and compiler parallelizability must be optimized.

Random Graph Model Papers

Jaideep Ray, Ali Pinar, and C. Seshadhri published the paper "Are we there yet? When to stop a Markov chain while generating random graphs," in the 9th Workshop on Algorithms and Models for the Web Graph in 2012 [100].

Abstract: Markov chains are convenient means of generating realizations of networks with a given (joint or otherwise) degree distribution, since they simply require a procedure for rewiring edges. The major challenge is to find the right number of steps to run such a chain, so that we generate truly independent samples. Theoretical bounds for mixing times of these Markov chains are too large to be practically useful. Practitioners have no useful guide for choosing the length, and tend to pick numbers fairly arbitrarily. We give a principled mathematical argument showing that it suffices for the length to be proportional to the number of desired number of edges. We also prescribe a method for choosing this proportionality constant. We run a series of experiments showing that the distributions of common graph properties converge in this time, providing empirical evidence for our claims.

Chapter 3

Comparing Community Assignments

In this section, we explore scalability and resolution limits of information theoretic community assignment comparisons methods. Comparing a community assignment to a ground truth is one way to assess the quality of a community detection algorithm. However, the methods typically used in the literature have some significant issues.

We explore the scalability of normalized mutual information in the context of comparing community detection algorithms on social networks. Dunbar's social science theory bounds the size of meaningful communities of people, yet social networks grow without bound. We analyze the implications of these conditions and corroborate our results with experiments. The latter use the LFR benchmark generator and the Louvain community detection algorithm.

Our work is not complete, but we give algebraic arguments that the ubiquitous normalized mutual information (NMI) distance measure favors refinement (splitting the communities into smaller pieces) over solutions found by the most popular algorithm. We have initial arguments that NMI also favors refinement over small perturbations of community members. This problem also increases with scale. We defer that discussion to a future research paper as they are not yet in a state for public release.

3.1 Introduction

Community detection in networks is becoming a mature field in the sense that thousands of papers on the subject have been published in the past decade. However, some of the most fundamental questions still have not been answered in a compelling way. These range from the mere definition of "community" to the best algorithms to find communities in various contexts to the best ways to compare community assignments. In this section, we work with an intuitive definition of community (a tightly-connected subgraph), we generate community assignments using the well-cited Louvain algorithm [18], and we focus on the issue of comparing two different community assignments. We find that one of the most familiar and heavily used comparison methods has fundamental problems when applied to synthetic data generated using realistic assumptions.

Dunbar's classical thesis in social science is based on a careful study of primates' social groups. [26] He found evidence that social groups were defined by each member's knowledge

of the pairwise relationships of other members in the group. He used regression to relate the brain sizes of various species of primates to observed group sizes and argued that the relationship was causal and bounded by the quadratic knowledge requirement. Extrapolating his results to human brain sizes, he arrived at what is colloquially known as the *Dunbar number*: an informal limit of roughly 150 (with loose confidence bounds) for the sizes of meaningful human social groups.

We accept this number as reasonable and explore the consequences of such a limit on the comparison of two community assignments in social networks. In this paper, we restrict our attention to the fundamental context of non-overlapping communities. Since our community sizes are bounded by a constant, the number of communities grows without bound. We claim that methods for evaluating community detection algorithms should *scale*. For the moment, we informally define this to mean that they should provide similar qualitative information as the size of the network increases, but community sizes remain Dunbar-constrained.

We show that according to NMI, a pathologically-refined community assignment "out-performs" the most highly cited algorithm in the most important region of the study space: fuzzy communities that are hard to detect. It does this because NMI does not scale. We show below that as the graph size increases, holding community sizes small, NMI judges the difference between the pathological assignment and the ground truth to be smaller and smaller. This happens despite the fact that the relative sizes of the communities in the two assignments remains the same.

It is not our goal in this paper to solve the problems that we expose. Our contribution is to illustrate fundamental problems inherent in applying NMI to community comparison.

3.2 Background

The mutual information between two clusterings X and Y of a set of objects is defined as

$$I(X,Y) = \sum_{ij} p_{ij}(X,Y) \log \frac{p_{ij}(X,Y)}{p_i(X)p_j(Y)} \tag{3.1}$$

where p_i is the likelihood that a randomly selected object is in partition $x_i \in X$, p_j is defined similarly for Y, and p_{ij} is likelihood that a randomly selected object is in both partition $x_i \in X$ and $y_j \in Y$.

Mutual information is typically normalized to constrain its value to $[0,1]$. There are various similar ways to do this, but we follow [90] and others who normalize with:

$$\mathrm{NMI}(X,Y) = \frac{2I(X,Y)}{H(X)+H(Y)} \tag{3.2}$$

where $H(X)$ is the *entropy* of partitioning X

$$H(X) = -\sum_i p_i(X) \log p_i(X). \tag{3.3}$$

The *Modularity* of a partitioning of the vertex set of a graph is a measure of the extent to which the communities of the partitioning have more internal edges than would be expected under a null model. [87] Its mathematical definition is not relevant to this paper, but it or its variants serve as the objective function for many community detection algorithms. Community assignments of maximum modularity are shown by [30] to suffer from a resolution limit. The communities of such assignments have minimum size proportional to the square root of the total number of edges in the network. One implication of this limit is that communities that are Dunbar-constrained would not be resolved in a solution of maximum modularity.

The *Louvain* algorithm [18] attempts to deal with this problem by collapsing the graph into a hierarchy of nested communities, letting the user select from a variety of resolution options. Its kernel operation is to move a single vertex from its current community to join the community of one of its neighbors. Although each move is designed to maximize the change in modularity, the algorithm as a whole offers an informal way to address the resolution limit: simply look at the various levels of the hierarchy and pick one that makes sense rather than one that maximizes the global modularity. This algorithm is extremely popular in the literature and can be run efficiently on very large graph instances. We choose it to represent community detection algorithms in our study. However, note that our goal is not to praise or criticize it or any other algorithm.

We choose the LFR benchmark generator [72] since it is heavily used in the community detection literature. This generator accepts three fundamental parameters: α, the exponent for a power law distribution of vertex degrees, β, the exponent for a power law distribution of community sizes, and μ, a mixing parameter that decides how tightly connected the communities are. The latter generally ranges from 0.1 (on average, 9 out of 10 connections per vertex are internal) to 0.7 (only 3 out of 10 are internal). There are other LFR parameters that bound the distributions. For example, we specify a and a range of community sizes $[10, 15]$ for our experiments, keeping vertex degree constant at 8.

3.3 Misleading information from NMI

Figure 3.1 shows a common idiom among community detection papers. The x-axis is the LFR μ parameter. Moving from left to right, communities become fuzzier and harder to identify. The y-axis is the NMI between the LFR ground truth community and the that found by an algorithm. The four lines are meant to represent the solution qualities of various algorithms as communities become fuzzier. Intuitively, higher on the y-axis is better. It has become common to judge one algorithm as "better" than another if its performance line stays high farther to the right of the plot. For example, in the figure, we would be tempted to declare the "CRP" algorithm the winner of this comparison. After all, its NMI with the ground truth is roughly 0.8 at the LFR μ value of 0.6, which indicates quite fuzzy communities. The other algorithms can't manage better than NMI values of roughly 0.5 at that point.

However, this information can be quite misleading. Consider a trivial "algorithm" that simply places each vertex alone in its own community. We will call that the *singleton* solution. Figure 3.2

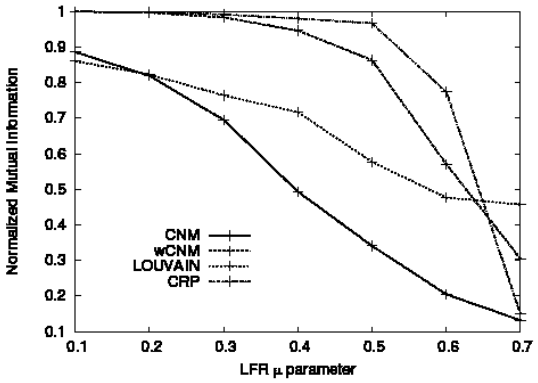


Figure 3.1. A typical plot comparing community detection algorithms

includes the performance of this solution.

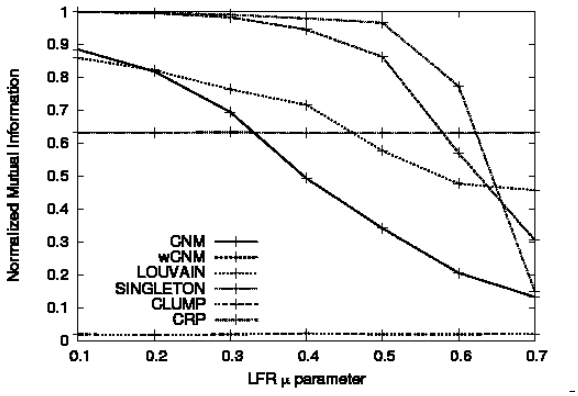


Figure 3.2. The singleton solution as an algorithm

Perhaps surprisingly, the singleton solution is right in the mix with respect to quality as measured by NMI. At the fuzziest community levels, it even seems to dominate the competition. Furthermore, we will show that this phenomenon is even worse as the graph size grows, holding community size relatively constant (in keeping with Dunbar's theory).

3.4 The scalability of NMI

Let $X = \langle X_1, X_2, \ldots, X_K \rangle$ be our LFR ground truth community assignment, i.e. a partitioning of the vertex set of a graph. Each community will have constant size n_x. Let $Y = \langle Y_1, Y_2, \ldots, Y_n \rangle$ be the singleton assignment. The probability that a randomly chosen vertes is in X_i and Y_j is:

$$p(X_i, Y_j) = \frac{|X_i \cap Y_j|}{n} = \frac{n_{xy}}{n} = \frac{1}{n}.$$

The probability that it is in X_i is:

$$p(X_i) = \frac{n_x}{n},$$

and we note the useful expression:

$$\frac{p(X_i, Y_j)}{p(X_i)p(Y_j)} = \frac{n|X_i \cap Y_j|}{n_x n_y}.$$

Since we are considering Y to be the singleton assignment, we have:

$$\frac{n|X_i \cap Y_j|}{n_x n_y} = \frac{n(1)}{n_x(1)} = \frac{n}{n_x}.$$

Plugging into the equation for NMI, we have:

$$\begin{aligned} \mathrm{NMI}(X,Y) &= -\frac{2\sum_{x\in X}\sum_{y\in Y} n_{xy}\log(\frac{n\ n_{xy}}{n_x n_y})}{\sum_{x\in X} n_x \log(\frac{n_x}{n}) + \sum_{y\in Y} n_y \log(\frac{n_y}{n})} \\ &= 1 - \frac{\log n_x}{2\log n - \log n_x}. \end{aligned}$$

This expression clearly approaches 1.0 as $n \to \infty$. With one million vertices and communities of size 10, $\mathrm{NMI}(X,Y) = 0.909$.

3.5 Discussion

Community detection algoritms should produce good quality solutions in social networks with Dunbar-bounded communities and ever-increasing numbers of vertices. We have shown that NMI is not a suitable way to judge this quality, at least as typically used in practice. The graph community assignment is transformed into a clustering problem of balls and bins, losing the edge information. We have other work showing other problems as well as the scaling issue highlighted in the previous section, and we plan to publish these results.

Chapter 4

Approximately Finding Patterns in Graphs

This section considers two different ways to do approximate matching of graphs. In each case, we can assume we are given a target graph and would like to find instances that are "close to" it in a library or collection of uncertain graphs. This collection could be a set of candidate subgraph matches taken from a large graph. This is typical of finding patterns in geospatial semantic graphs, typical of work for the PANTHER grand challenge, for instance. The collection could also be individual graphs from some other application where we wish to compare a new graph from the "wild" to a set of known graphs.

In Section 4.1, we wish to answer the question "What is the probability that a given candidate is an instance of the target?" We were motivated by the approximate subgraph matching problem in geospatical semantic graphs. Generally the pattern is a small graph with attributes (properties). We explicitly assume that the target is rare, and therefore there are not enough (good) data for training. The method combines expert elicitation with regression to match a beta distribution.

In Section 4.2, we describe a way to compute a distance measure between graphs and to cluster sets of similar graphs. This method is largely topological and intended for graphs with perhaps 10,000 nodes.

4.1 Learning a Beta Distribution

This section describes one possible method for finding a pattern of interest in a larger graph under uncertainty. It has the advantage of statistical foundations and can give a confidence interval on match quality. However, it depends upon care eliciation of expert knowledge. This eliciation must estimate a series of conditional probabilities. It will likely be difficult for the expert to give high-quality answers. The expert must also estimate his/her uncertainty at each step, which is even more challenging. Also the sequential nature of the eliciation, considering a set of attributes in a given order, makes the probability calculations also order dependent. This may not be intuitive for someone trying to use this information to make a decision.

Our goal is to find a small pattern in a large graph. Computer scientists can think of this as approximate subgraph isomorphism. The graph has *attributes* on both the nodes and the edges. By this we mean certain properties or features with values. They can be real numbers, categorical

values (from a fixed finite set), binary values, etc. There is some tolerance or uncertainty in the pattern at multiple levels, both at the fundamental properties or attributes of the nodes and in the existence and combination of the pieces into a whole. There is also uncertainty in the attributes of the large graph we search to find approximate instances of the pattern.

Throughout this section, we use a canonical example where the large graph is a geospatial semantic graph representing an aerial image. Nodes represent features on the ground such as buildings, roads, patches of grass, trees, lakes, etc. The pattern is a general high school. It has a classroom building, a parking lot, and a football field. The classroom building should be of ground type "building" and should be in a certain size range (say, a range of acceptable areas with decreasing value down to some minimum and up to some maximum size). The parking lot should be of ground type "pavement" and also fit in a (different) size range. The football field should be of ground cover "grass," at least for areas with sufficient rainfall, though "dirt" or a mixture of "dirt" and "grass" may also be acceptable. The football field should have a certain aspect ratio and be a certain size (within tolerance). The football field should be sufficiently near the building and the parking lot should be sufficiently near the building.

We describe one possible procedure for answering the following question: Given a subset of the large graph which is a candidate match for the pattern, what is the quality of the match? In this section, we describe how to give a probabilistic answer: "The probability this pattern is a high school is 65%. This is a mean probability. The actual probability is between 52% and 78% with 95% confidence." Both the mean and the confidence interval are based on "best available information" as described below. It is possible to generalize to give a probability distribution over a set of competing possibilities ("What is the probability this is a high school vs a middle school vs a church vs something else?"). However, that description is out of the scope of this discussion.

A prior distribution incorporates knowledge from experience or other forms of expertise to express general uncertainty in whether a set of objects represents a high school or not based on the attributes of the objects. One can think of it as a distribution representing normal variation in high schools (not to be confused with the normal distribution). Usually, one updates a prior distribution based on some data (observation) to get a posterior distribution for that data instance. In this first-pass method, we are creating a distribution that depends on the observed data using a reasonably simple calculation.

There are many examples of high schools, even if you focus on, say, urban high schools in the US northeast. One can use a set of examples to create a prior distribution. However, we assume for this section that the pattern we seek is rare. It is something we want to recognize if we see it, but we may never see it. For the sake if discussions we refer to these hard-to-observe patterns as unicorn farms. In this setting, we are forced to use expert knowledge. Someone who has been thinking about what a unicorn farm might look like can help us build the prior. There are careful formal elicitation processes that require extensive study and training to master. We will describe one way elicit data from an expert for this particular problem later in the section.

We will now describe how to produce a predictive distribution for the probability a single candidate "entity" is of the type being searched for. If there are multiple candidates, one may repeat this procedure for each candidate independently. We assume the candidate has been determined by

a preliminary screening search. For example, in the preliminary search, the area of the building is at least the minimum and at most the maximum value specified by an expert.

We consider information accumulating piece by piece indexed as $i = 1, \ldots, n_a$, where n_a is the number of attributes. That is, we consider each attribute in a consistent order. The order is set by the order of elicitation, described later. It can be determined by an expert's opinion of importance, or could be arbitrary. But once set by the elicitation, we will consider each attribute in that given order. For example, building size might correspond to $i = 1$, presence of grassy area might correspond to $i = 2$, aspect ratio of grassy area might correspond to $i = 3$, and so forth.

We denote the probability that a candidate C is "of interest" after considering the first i attributes by θ_i.

We wish to produce a distribution (at attribute i) that reflects our belief about the possible values of $\theta_i \in (0, 1)$. Here is one possbile way to create a predictive distribution.

Prior Distribution for θ_i.
The simplest predictive distribution for θ_i is to use a modeled prior. Since $\theta_i \in (0, 1)$ we need the support of this prior to be the unit interval, and the natural (and flexible) choice is a beta distribution. The density of a beta distribution is usually written as

$$g(\theta_i|\alpha_i, \beta_i) = \frac{\Gamma(\alpha_i + \beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} \theta_i^{\alpha_i - 1}(1 - \theta_i)^{\beta_i - 1}; \quad 0 < \theta_i < 1. \tag{4.1}$$

where $\alpha_i > 0$ and $\beta_i > 0$ are parameters and $\Gamma(h)$ is the Gamma function:

$$\Gamma(h) = \int_0^\infty x^{h-1} e^{-x} dx$$

for $h > 0$. It's probably easier to think of the Gamma function based on its properties. It's an extension of the factorial function, extended to real values except negative integers. There is a nice discussion in the course notes for Scott Hyde [61], which we summarize now. If h is a positive integer, then $\Gamma(h) = (h-1)!$. The function satisfies a recursive property similar to factorial:

$$\Gamma(h) = (h-1)\Gamma(h-1). \tag{4.2}$$

This relationship, along with tables of values for $\Gamma(h)$ for $1 < h < 2$ make it relatively straightforward to compute $\Gamma(h)$ for any h (except negative integers). See Hyde's web page [61] for some examples. The Gamma function should be implemented in any standard statistics package. For example, it's in R. The ratio of Gamma functions that is the normalization constant in the beta distribution with density given above (4.1) is also $1/B(\alpha_i, \beta_i)$, where the denominator is a beta function.

Note that the parameters of the beta prior in (4.1) are indexed by attribute number. The prior reflects expert belief about the possible values of the probability that candidate C is of interest at attribute i – that is, his/her belief about the possible values of θ_i. The greatest belief will be given

by the mode of this distribution. The expected value and variance are

$$\begin{aligned} E(\theta_i) &= \frac{\alpha_i}{\alpha_i+\beta_i} \\ var(\theta_i) &= \frac{\alpha_i\beta_i}{(\alpha_i+\beta_i)^2(\alpha_i+\beta_i+1)} \end{aligned} \tag{4.3}$$

An interval that contains 90% or 95% of our belief about the value of θ_i is given by the central 90% or 95% of the density (4.1).

There are many pairs of α_i and β_i that give the same mean value. For example, whenever $\alpha_i = \beta_i$, the beta distribution has a mean of $1/2$. But larger values of $\alpha_i = \beta_i$ have smaller variance, reflecting greater confidence. See Robinson [106] for a nice discussion of the beta distribution applied to prediction of batting averages in baseball. Even if the reader knows only a little about baseball, this is a nice discussion. Robinson's example case illustrates that the higher confidence comes from more real trials blended with prior beliefs.

We would like to update values of α_i and β_i as information becomes available or is considered in sequence. To do this, first reparameterize (4.1) as

$$\begin{aligned} \mu_i &= \frac{\alpha_i}{\alpha_i+\beta_i} \\ \phi_i &= \frac{1}{\alpha_i+\beta_i+1}. \end{aligned} \tag{4.4}$$

Notice that both α_i and β_i are allowed to vary over i, but only in a constrained manner so that ϕ_i remains constant over i. μ_i is the mean of the beta distribution we are using to model θ_i. If we must give a single best guess at θ_i, it will be this mean. The parameter ϕ controls confidence intervals and comes from elicitation of confidence from experts.

One might consider a supervised learning approach, learning based on labeled examples, for setting the prior distribution. For a "unicorn farm" there probably will not be enough data for such an approach. Even for high schools, where there is a fair number of training instances, supervised learning may not be a good idea for this particular approximate matching problem. The reason is the training data sets would need to be even larger than under "regular" supervised learning. We need a lot of covariates because of interactions among the pieces. For example, in the high school case, there are six attributes; building size, pavement area, grass area, grass aspect ratio, distance between grass and pavement and distance between building and pavement. If we allow for all possible interactions among these attributes, we would require 62 parameters (6 main effects, 15 two-way interactions, 20 three-way, 15 four-way and 6 five-way). Certainly this number could be reduced, but it is quite possible we could end up with a model that has 10 to 15 parameters.

Without sufficient training data, we must choose parameter values based on expert opinion. Here's an outline of a potential procedure: sequential updating based on expert opinion. This outline is specific for the high school problem, but it generalizes to other settings.

We use regression to find the expected values for the probability a candidate is a high school depending upon the values of the relevant attributes/features. The key element in this potential procedure is to determine parameter values β_i sequentially, indexed over attributes. Regression parameters are normally denoted by β. But that conflicts with using it as a parameter of the beta distribution. We make the substitution to γ for clarity for non-statisticians.

Recall that mean μ_i is a function of beta distribution parameters α_i and β_i. We find these parameters by regression, where $\gamma_{i,0}$ and $\gamma_{i,1}$ are the linear regression variables, the intercept and slope values respectively, for attribute i.

$$\log\left(\frac{\mu_i}{1-\mu_i}\right) = \sum_{j=1}^{i}(\gamma_{j,0} + \gamma_{j,1} z_j) \tag{4.5}$$

where z_j is defined in the following list.

Define the following quantities:

- C is the candidate entity, a subgraph (generally a small set of connected set of nodes and edges) selected from the larger graph. It might pass some minimum selection criteria, such as containing a building in the size range $6,000$ to $30,000$ square meters, having a large $> 10,000m^2$ paved region and a grassy area in the vicinity.
- θ_i are probabilities that candidate C is of interest at attribute i, $i = 1,\ldots,n_a$ (or a possibly earlier point of determination)
- Let x_1 be building size for C, x_2 the area of the nearby paved area, x_3 the area of the grassy area, x_4 the aspect ratio of the grassy area, x_5 the distance between paved area and building, and x_6 the distance between paved and grassy areas.
- Let $T_1,\ldots,T_6$ denote "target" values for the six attributes as determined by expert opinion. For example, we might choose $T_1 = 10,000$, $T_2 = 15,000$, $T_3 = 10,000$, $T_4 = 2.5$, $T_5 = 75$ and $T_6 = 275$.
- Variables z_i represent deviation of the candidate's value for attribute i from the target value for attribute i.
- Variables x_i represent the candidate's attribute values.
- p_i and Δ_i are determined from expert opinion. p_i is the additional probability the candidate is of interest if it matches the target for the ith attribute beyond the probability of interest for matches on the first $i-1$ attributes. Note that $\sum_i p_i \leq 1$. Δ_i gives the decrease in probability of a candidate being of interest per unit deviation away from the target value of attribute i. See the description below.

  Note: There appears to be an alternative interpretation of of p_i that may be easier to elicit. The $\sum_i p_i$ represents the probability the candidate is of interest if all the attributes precisely match their target value. That is, the conditions on the probability must be perfectly met. The

reason this sum is not generally equal to 1 is that the experts should allow for other objects that match the template but are not high schools. Since the p_i partition the total probability, you can think of them as weighting the importance of the individual attributes. That piece p_i can be earned (or not earned) by an attribute. The computation/procedure below doesn't quite work that way, since otherwise it would not be as sensitive to order as it is. However, this kind of elicitation may lead to a less order-dependent method.

Step 1
Construct the covariate

$$z_1 = |x_1 - T_1| \text{ or } z_1 = \frac{|x_1 - T_1|}{\sigma_1}, \tag{4.6}$$

where, in the later alternative, σ_1 would need to be determined somehow.

Expert opinion provides two values

- If $x_1 = T_1$ (or $z_1 = 0$) then the probability a candidate entity is of interest is p_1.
- For every unit x_1 differs from T_1 (or every unit of z_1) the probability a candidate entity is of interest decreases by Δ_1.

From this information we can determine values of $\gamma_{1,0}$ and $\gamma_{1,1}$ in (4.5). The model at attribute $i = 1$ is

$$\log\left(\frac{\mu_1}{1-\mu_1}\right) = \gamma_{1,0} + \gamma_{1,1} z_1 \tag{4.7}$$

For $x_1 = T_1$, or equivalently $z_1 = 0$, where the candidate value on attribute 1 exactly matches the target, expert opinion has provided that $\mu_1 = p_1$ so that

$$\gamma_{1,0} = \log\left(\frac{p_1}{1-p_1}\right) \tag{4.8}$$

For $x_1 \neq T_1$, or equivalently $z_1 \neq 0$, expert opinion gives $\mu_1 = p_1 - \Delta_1 z_1$. Using (4.7) we have

$$\gamma_{1,1} = \frac{1}{z_1}\left(\log\left[\frac{p_1 - \Delta_1 z_1}{1-(p_1 - \Delta_1 z_1)}\right] - \gamma_{1,0}\right) \tag{4.9}$$

Step 2
Construct the covariate

$$z_2 = |x_2 - T_2| \text{ or } z_2 = \frac{|x_2 - T_2|}{\sigma_2}, \tag{4.10}$$

where, in the latter alternative, σ_2 would need to be determined somehow.

Expert opinion now provides two values.

- Given that $\mu_1 = p_1$ (i.e., x_1 was the target T_1), if $x_2 = T_2$ (or $z_2 = 0$) then the probability a candidate entity is of interest is $p_1 + p_2$. Note that this means the expert provides information on the increase in probability caused by x_2 being its target value *given that x_1 was its target value*.

- For every unit x_2 differs from T_2 (or every unit of z_2), the probability a candidate entity is of interest p_2 decreases by Δ_2.

From this information we can determine values of $\gamma_{2,0}$ and $\gamma_{2,1}$ in (4.5). The model at attribute $i = 2$ is

$$\log\left(\frac{\mu_2}{1-\mu_2}\right) = \gamma_{1,0} + \gamma_{1,1}z_1 + \gamma_{2,0} + \gamma_{2,1}z_2 \tag{4.11}$$

Now, for $x_1 = T_1$ (or $z_1 = 0$) **and** $x_2 = T_2$ (or $z_2 = 0$) expert opinion gives $\mu_2 = p_1 + p_2$ so that

$$\gamma_{2,0} = \log\left(\frac{p_1+p_2}{1-(p_1+p_2)}\right) - \gamma_{1,0} \tag{4.12}$$

For $x_2 \neq T_2$ or $z_2 \neq 0$ expert opinion gives $\mu_2 = (p_1 - \Delta_1 z_1) + (p_2 - \Delta_2 z_2)$ so that from the model,

$$\log\left(\frac{\mu_2}{1-\mu_2}\right) = \gamma_{1,0} + \gamma_{1,1}z_1 + \gamma_{2,0} + \gamma_{2,1}z_2$$

This implies that

$$\gamma_{2,1} = \frac{1}{z_2}\left(\log\left[\frac{p_1 - \Delta_1 z_1 + p_2 - \Delta_2 z_2}{1-(p_1 - \Delta_1 z_1 + p_2 - \Delta_2 z_2)}\right] - \gamma_{1,0} - \gamma_{1,1}z_1 - \gamma_{2,0}\right) \tag{4.13}$$

Step i

The progression is now established. In general, for $i = 1, \ldots, n_a$, the model is given by (4.5) where

$$z_i = |x_i - T_i| \text{ or } z_i = \frac{|x_i - T_i|}{\sigma_i}, \tag{4.14}$$

In the later case, σ_i would need to be determined somehow.

Expert opinion provides the incremental probabilities

$p_i =$ increase in probability given all previous attributes at targets and attribute i on target

and

$\Delta_i =$ decrease in p_i for each unit current attribute x_i differs from target T_i

Then $\gamma_{i,0}$ and $\gamma_{i,1}$ may be determined as

$$\gamma_{i,0} = \log\left(\frac{\sum_{j=1}^{i} p_j}{1-\sum_{j=1}^{i} p_j}\right) - \sum_{j=1}^{i-1}\gamma_{j,0} \tag{4.15}$$

and

$$\gamma_{i,1} = \frac{1}{z_i}\left[\log\left(\frac{\sum_{j=1}^{i}(p_j - \Delta_j z_j)}{1-\sum_{j=1}^{i}(p_j - \Delta_j z_j)}\right) - \sum_{j=1}^{i}\gamma_{j,0} - \sum_{j=1}^{i-1}\gamma_{j,1}z_j\right] \tag{4.16}$$

where, as before, the righthand sides of (4.15) and (4.16) use values we have already calculated.

Once we have computed all the μ_i, and selected a ϕ_i, we can use (4.4) to compute the α_i and β_i. Either way, we can use a tool like R to find the 95% region, specifically the value of $\theta_{\min}$ with 2.5% of the distribution weight below it, and the value of $\theta_{\max}$ with 2.5% of the distribution weight above it. The mean (θ_{n_a})is our best guess, and the range $[\theta_{\min}, \theta_{\max}]$ gives a 95% confidence interval. The choice of ϕ, derived from elicitation, determines on the variance, hence the size of the confidence interval, and the implied level of confidence in that answer.

Comments

1. The number of attributes included $i = 1, \ldots, n_a$ could be determined by when experts believe that an entity with all target values is an entity of interest with probability equal to or close to 1. That is, for some small δ, choose n_a such that

$$\sum_{j=1}^{n_a} p_j = 1 - \delta$$

2. Clearly, the order in which attributes are considered may have an effect. A check on this whole approach could be to obtain expert opinions in several orders and compare results. An expert's p_j values will differ depending on the order of attribute consideration.

3. This approach does not deal with interactions directly, rather it has only hidden the issue under a blanket of expert opinion.

One could use a more complex approach when the beta distribution doesn't adequately express uncertainty in θ_i. That approach involves also using priors for the α_i and β_i parameters. It is more complicated than the simple model above and probably not reasonable for a first pass.

4.1.1 An example

Here is an example using the method described above. This is slightly more complicated because it allows the Δ deviation values to differ depending upon whether the value is above or below the target. The values used in this example are artificial and are intended only to illustrate the manner in which expert opinion could be used. Any number of modifications to the procedure presented are possible so this should not be taken to be a presentation of a finished procedure.

Suppose we have identified a candidate as a potential high school. There are six attributes to be assessed in the high school template, 1: Building Size, 2: Size of Paved Area, 3: Size of Grassy Area, 4: Aspect Ratio of Grassy Area, 5: Distance from Pavement to Building, and 6: Distance from Pavement to Grassy Area. Target values and units for these attributes are

- Building Size. Observed as m^2, units are $1000m^2$, target is $T_1 = 10$ (i.e., $10,000m^2$).
- Paved Area. Observed as m^2, units are $500m^2$, target is $T_2 = 24$ (i.e., $12,000m^2$).

- Grassy Area. Observed as m^2, units are $100m^2$, target is $T_3 = 80$ (i.e., $8,000m^2$.
- Grassy Area Aspect Ratio. Unitless, target is $T_4 = 2.0$.
- Distance Pavement to Buildling. Observed as m, units are m, target is $T_5 = 50$.
- Distance Pavement to Grassy Area. Observed as m, units are m, target is $T_6 = 200$.

The steps in the proposed procedure can be summarized as follows.

1. Determine regression coefficient values $\{(\gamma_{i,0}, \gamma_{i,1} : i = 1, \ldots, 6\}$ according to the procedure outlined above. We require the values $\{p_j : j = 1, \ldots, 6\}$ and corresponding Δ_j values. Here, we make one elaboration in that the decrease in probability for covariate values that differ from target is allowed to be different for covariate values that are less than the target and those that are greater than the target. Thus, we now need values $\{\Delta_{j,\ell}, \Delta_{j,h} : j = 1, \ldots, 6\}$. We use $p_1 = 0.3$ (building size), $p_2 = 0.2$ (paved area size), $p_3 = 0.1$ (grassy area size), $p_4 = 0.2$ (grassy area aspect ratio), $p_5 = 0.1$ (distance from paved area to building) and $p_6 = 0.05$ (distance from paved area to grassy area). A candidate with target values for each of the six attributes then would have probability 0.95 of being an entity of interest. We describe how to determine the values of $\Delta_{j,\ell}$ and $\Delta_{j,h}$ below.
2. With regression coefficients determined in the previous step, determine values of the parameters of the associated beta distribution. Note that we need one additional parameter: the ϕ described previously. This should be determined from confidence elicitation. For these examples, we will treat it like a "tuning parameter."
3. With parameters α_i and β_i (at attribute i), a 95% interval for the probability that the candidate is an entity of interest is determined as $(q_{0.025}, q_{0.975})$ where

$$\begin{aligned} 0.025 &= \frac{\Gamma(\alpha_i+\beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} \int_0^{q_{0.025}} y^{\alpha_i-1}(1-y)^{\beta_i-1}\,dy \\ 0.975 &= \frac{\Gamma(\alpha_i+\beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} \int_0^{q_{0.975}} y^{\alpha_i-1}(1-y)^{\beta_i-1}\,dy \end{aligned}$$

We illustrate the way to determine $\Delta_{j,\ell}$ and $\Delta_{j,h}$ using building size. Using units of $1000m^2$ for this attribute, the target was $T_1 = 10$. Somewhat arbitrarily, but guided by our experience, we decided that buildings less than $4,000m^2$ should have probability 0 of being a high school, and similarly for buildings greater than $30,000m^2$ should have probability zero. Computing simple linear decreases for buildings less than or greater than the target then produced $\Delta_{1,\ell} = 0.05$ and $\Delta_{1,h} = 0.015$. Figure 1 contains an illustration. We followed the same procedure for the other attributes. Values are summarized in Table 1.

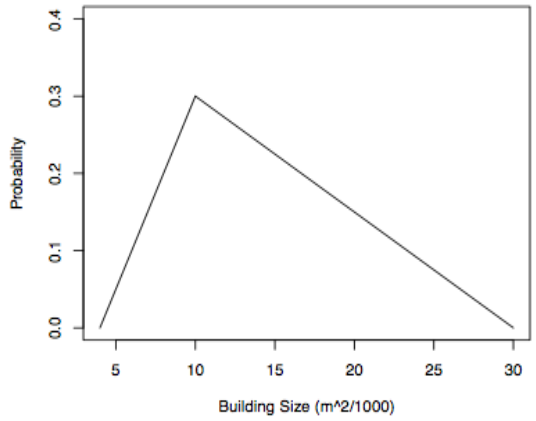


Figure 4.1. Illustration of function to choose $\Delta_{1,\ell}$ and $\Delta_{1,h}$.

Attribute	Units	Target	p_i	Low Cut	High Cut	$\Delta_{i,\ell}$	$\Delta_{i,h}$
Bldg Size	1000	10	0.30	4	30	0.0500	0.0150
Paved Size	500	24	0.20	10	38	0.0143	0.0143
Grass Size	100	80	0.10	50	120	0.0033	0.0025
Aspect Ratio	NA	2.0	0.20	1.5	3.5	0.4000	0.4000
Distance 1	1	50	0.10	0	100	0.0020	0.0020
Distance 2	1	200	0.05	0	400	0.00025	0.00025

Table 4.1. Summary of setup values for six attributes in the high school template.

Example 1

The first example has all attributes at their target values and takes $\phi = 0.005$ (which implies quite large values for α_i and β_i and thus small variances for the beta prior distribution). Results are:

```
  attribute pred     lower     upper
1      bldg 0.30 0.2384827 0.3653249
2      pave 0.50 0.4307789 0.5692211
3     grass 0.60 0.5312303 0.6668655
4    aspect 0.80 0.7418233 0.8524684
5     dist1 0.90 0.8547968 0.9376059
6     dist2 0.95 0.9157908 0.9756950
```

Repeating this example (all attributes at target) but with $\phi = 0.05$ produces substantially wider intervals:

```
  attribute pred     lower     upper
1      bldg 0.30 0.1221411 0.5176262
2      pave 0.50 0.2835712 0.7164288
3     grass 0.60 0.3781208 0.8019413
4    aspect 0.80 0.5987966 0.9419732
5     dist1 0.90 0.7350336 0.9879965
6     dist2 0.95 0.8197441 0.9988709
```

Example 2

In this example, we have a building that is somewhat smaller ($9000m^2$) than the target, but all other attributes remain at target values. The tuning parameter was $\phi = 0.005$.

```
  attribute      pred     lower     upper
1      bldg 0.2500000 0.1924669 0.3122916
2      pave 0.4375000 0.3694188 0.5067714
3     grass 0.5384615 0.4690800 0.6071106
4    aspect 0.7567568 0.6949531 0.8136735
5     dist1 0.8750000 0.8257531 0.9171193
6     dist2 0.9366197 0.8989383 0.9660265
```

Example 3

In this example, we keep the somewhat-too-small building from example 2, increase the paved area size to $15,000m^2$, increase the grassy area to $12,000m^2$, with a somewhat-too-small aspect ratio of 1.8. The results are

```
  attribute      pred     lower     upper
1      bldg 0.2500000 0.1924669 0.3122916
2      pave 0.2573458 0.1991648 0.3201457
3     grass 0.2573458 0.1991648 0.3201457
4    aspect 0.3621950 0.2969764 0.4300375
5     dist1 0.5609652 0.4916819 0.6290876
6     dist2 0.7295409 0.6658892 0.7888231
```

Example 4

The previous examples have all had relatively minor departures from target values. In this example we take a quite large building with a quite large paved area, but there is a grassy area of about the right dimensions nearby, although slightly farther away than target for a high school. We used these values: Building Size $22,000m^2$, Paved Area $17,000m^2$, Grassy Area $8,000m^2$ (target), Aspect Ratio 2.0 (target), Distance 1 $50m$ (target) and Distance 2 $400m$. Results give much smaller probabilities of a high school.

```
  attribute      pred      lower     upper
1      bldg 0.1200000 0.07872944 0.1684924
2      pave 0.1770000 0.12729916 0.2328454
3     grass 0.2439136 0.18693486 0.3057666
4    aspect 0.4624428 0.39377559 0.5318253
5     dist1 0.6593543 0.59223878 0.7234357
6     dist2 0.5770000 0.50787440 0.6446593
```

4.2 Bertillonage: A Statistical Measure for Graph Similarity

This section describes an approach for constructing a formal metric that characterizes the similarity between two graphs. Our goal is to approximate the complexity of a graph with a small(er) set of features and employ those features to identify other similar graphs. We have coined the phrase *graph Bertillonage* to describe this process.

Bertillonage is a simple forensic analysis technique based on bio-metrics that was developed in 19th century France before the advent of fingerprints. Alphonse Bertillon was a French criminologist and anthropologist who created the first system of physical measurements, photography, and record-keeping that police could use to identify recidivist criminals. Bertillon developed an anthropometric method based on measurements from head and body, shape of facial features, and individual marks (tattoos, scars, etc.). These characteristics were filed with photographs of the suspects and cross-indexed to permit quick, systematic access. The result was that the identification of a suspect was systematically reduced from the search of a massive database of suspects to the examination of small set of suspects with similar features.

The method, referred to as Bertillonage, worked well under ideal conditions, but was eventually abandoned in favor of fingerprints.

The objective of forensic Bertillonage shares many of the same characteristics with the current problem: using a set of graph features that capture the fundamental complexities of a graph, and comparing the similarity of these features with the features of another graph. This similarity metric allows quick searching of a library of graphs to find graphs with similar structural (and possibly functional) characteristics. The resulting set of 'suspects' can be examined manually for a detailed comparison.

There are two major groups of approximate graph matching/similarity methods [37]. The first of these requires a correspondence between nodes, i.e. nodes are labeled and the labels have meaning. For example, the graph similarity metrics proposed in [92] or the network performance measures [25]. The second group does not require node correspondence and typically rely on feature extraction and pattern matching or (sub) graph matching coupled with a type of node correspondence. This latter group includes such methods as spectral analysis such as that proposed by Wilson [133] for undirected graphs. (As noted by Wilson, the spectrum of a graph (i.e. the set of eigenvalues) is considered to be too weak to be a useful tool for representing graphs).

Various other methods in this latter group include the state vector machine approach suggested by Li, et al [78]. This approach employs the use of feature vectors which shares some general concepts the approach we describe in Section 4.2, Also similar is the use of global algebraic feature vectors as suggested by Fiedler [27]. Finally, there are node affinity algorithms such as PageRank [91].

In this section, we consider graphs where there is no obvious correspondence between nodes. They may have been created to represent different phenomena, or different individuals/settings. Thus our data is unlabeled, in the sense that the nodes have no identity. We consider undirected and/or directed edges. Finally, we allow incomplete data such as missing nodes or edges. Our goal is not to determine if two graphs are isomorphic, but to determine the *similarity* between two graphs. Ideally this similarity metric will have a statistical basis and it will be possible to perform statistical hypothesis tests on the graph similarity.

Typical approaches involve some variation of iterative heuristics, edit distance, or feature/pattern mapping. Our approach to measuring graph similarity is based on a Macindoe and Richards' graph comparison method [83]. They compare graphs by measuring the energy required to transform one graph-feature-based representation into another. They represent a graph as a multi-dimensional histogram. The histogram counts/represents points, each of which is a feature vector of a subgraph. Specifically each point represents an induced neighborhood subgraph for each vertex in the graph. The set of induced subgraphs, as a whole, represents local structure. The features in the vector are more classic global graph statistics on the induced subgraph. Macindoe and Richards then use Earth Movers Distance (EMD) to measure the effort required to transform the histogram (weighted multi-dimensional bins) of one graph into the histogram of another graph. This quantitative difference serves as their distance measure. Given the distance measure, Macindoe and Richards use these pairwise graph distances to compare sets of graphs. They use heat maps, which display all pairwise distances, and dendrograms, which is a tree structure. At every branch in a dendrogram, elements on one side are (according to the clustering method) heuristically more closely related to each other than to the elements on the other side. It is a way of displaying the data set according to a particular agglomerative method, and quite sensitive to the details of the method. We describe these various pieces in this section.

Robner [111] introduced *signatures* as a variable-sized substitute for classic histograms, especially for applications such as image comparison. In a histogram, multi-dimensional vectors are placed into evenly-spaced fixed-sized bins, one for each dimension, based on the values in that dimension. In a signature, the points are clustered, for example, using a standard methods such as (one-dimensional) k-means [5, 80], with the clusters taking the place of the histogram bins.

Rubin [111] describes some of the advantages of a signature. His brief description was that histograms can be wasteful if many values are not represented. In his example, in a desert scene, many of the pixels will be yellow-brown for the ground and plants, and blue for the sky, with little in the green range. However, some elements of an image may need a fine scale to represent important details. Any method that just picks a fixed histogram size (such as Macindoe and Richards, who use a 0.2 granularity for all values), "cannot achieve balance between expressiveness and efficiency." Signatures can adjust level of detail, much as adaptive grids for scientific computing can give more cells to areas where the geometry or underlying phenomena need it, without doing wasteful computations in simpler areas.

As an example, Figure 4.2 shows equal-sized bins for a two-dimensional data set. Many of the bins in the upper right are empty. Figure 4.3 shows bins with different sizes, which has less wasteful empty space. Signatures are more general than this simple rectilinear example. Points are not assigned to a cluster based only on threshold comparisons to their element values.

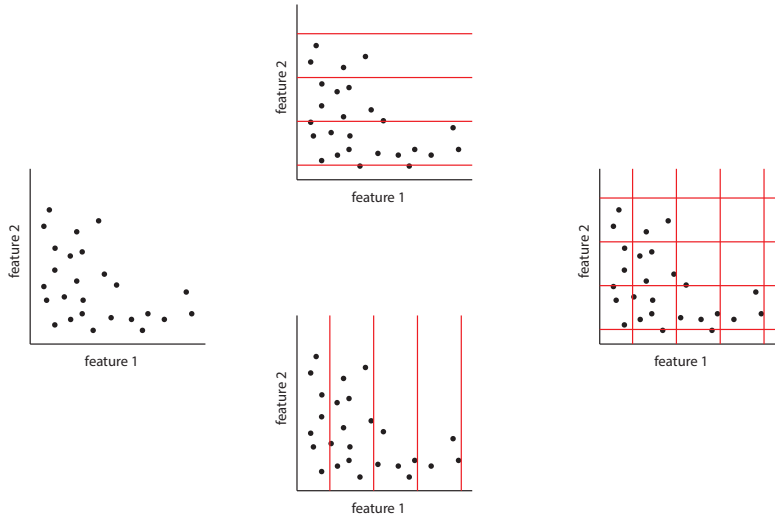


Figure 4.2. Equal Size Histogram Bins (2D). The upper right corner has empty bins, which are computationally wasteful.

Rubner also used introduced Earth Mover's distance (EMD) in the same paper [111]. Rubner listed some of the advantages of EMD, paired with a signature, for image comparisons: adaptive detail, partial matches (important when, for example, something obscures part of one image), and allowing images of different sizes. These are all properties that can be helpful for graph comparison.

The primary novelty of our work is the details of the graph signature and our applications. We use simpler metrics than Macindoe and Richards, which were sufficient to give good results on some Sandia applications. We will not describe the applications in detail in this report, though we briefly describe then in Section 4.2.4. Macindoe and Richards' experiments in [83] are on fifteen real data sets. All but one have no more than 450 vertices, and the largest has 1490 nodes. In a first

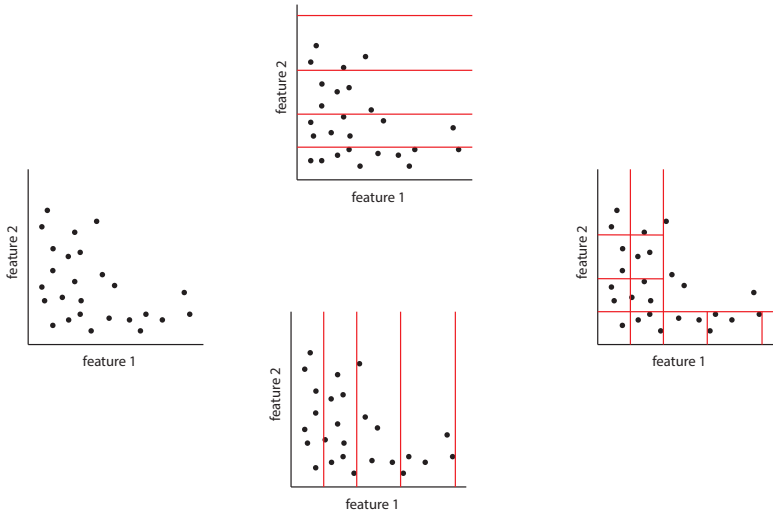


Figure 4.3. A simple example of variable Size Signature Bins (2D).

step to an investigation over random graphs, we describe a simple example involving a library of Erdős-Rényi random graphs. The approach can distinguish between graphs generated with slightly different size and density parameters. We have also applied it to an application with graphs with 10,000 or more nodes. We give a list of technical differences after we describe our graph signatures and how to compute graph similarity.

4.2.1 Graph Signatures

Using the Macindoe-Richards method, We can partially describe some aspects of any given graph G with a vector of features. To create our signature for graph G, we generate a set of subgraphs G_i from graph G, as described below. We sketch the subgraphs with a small feature of topological (that is, structural) statistics. We then cluster the resulting set of vectors in each dimension independently using standard methods. The concatenated representation of the clusters is the signature for graph G.

Some researchers currently use attributes as well as structural features for these vectors. However, as a first attempt to do this kind of graph comparison, we chose to use features that apply to every graph, including those with no attribute data.

Graph Structural Metrics We now list some previously-defined graph structure metrics that could be candidates as features in our vectors. Macindoe and Richards [83] consider leadership, clustering coefficient, and diversity in their signature. They claim Richards and Wormald [101] first introduced these measures. They believe this triple is particularly relevant for social networks. We use closeness centrality, betweenness centrality, and leadership. For our applications, we wanted metrics that were not necessarily tailored to social networks.

Centrality Metrics Centrality metrics capture node importance.

Closeness Centrality Closeness centrality (normalized) measures the (average) distance from a particular node to all others. In and unweighted graph, if information were to start at node v and cross all adjacent edges simultaneously in a step, and all nodes that receive that information and relay to all their neighbors, etc, the closeness centrality measures the average time required for a node other than v to receive the message. Closeness centrality is the inverse of the average distance, so that nodes with small average distance get a large closeness score. Let $\ell(v,i)$ represent the length of the shortest path from node v to node i. For closeness centrality score for node v in Graph G is:

$$C_v(G) = \frac{|V|-1}{\sum_{i \neq v} \ell(v,i)} \tag{4.17}$$

Betweenness Centrality Betweenness centrality (normalized) measures the potential for a node to control, or disrupt, the communication within the network. It is the average fraction of shortest paths between i, j that go through a particular node, taken over all pairs i, j.

$$C_v(G)|_{i,j} = \sum_{i \neq v \neq j} \left[\frac{K g_{ij(v)}}{g_{ij}} \right] \tag{4.18}$$

Where g_{ij} is the number of shortest paths between i and j, and $g_{ij(v)}$ is the number of shortest paths between i and j that contain v. For directed graphs the normalization is $K = 1/(n^2 -$

$3n+2$) and for undirected graphs $K = 2/(n^2-3n+2)$, where n is the number of nodes in the graph. This represents the number of pairs of nodes in the graph that do not involve node v [$(n^2-3n+2) = (n-1)(n-2)$].

Leadership Metric [32] Measures the degree to which a particular node dominates the connections between nodes.

$$C_v(G) = \frac{\sum_{i=1}^{n}(d_{max} - d_v)}{(n-1)} \tag{4.19}$$

If all nodes have the same degree, then the leadership metric is low (zero). Alternatively, if one node is connected to all the other nodes, which themselves are not connected (star graph), the centrality is maximal. This measure is for the graph as a whole, not for an individual vertex in the graph.

Here is a metric that Macindoe and Richards use for their signatures [83]:

Clustering Metric/Clustering Coefficient Triadic closure expressed as a clustering metric:

$$T(G) = \frac{3 \times (\text{\# triangles})}{(\text{\# length-two-paths})} \tag{4.20}$$

If two nodes are connected through a third node, T captures the probability that two nodes are directly connected. Graphs with few triangles, such as bipartite graphs in the worst case, indicate less clustering of nodes. T is maximal for a complete graph. Macindoe and Richards use 6 instead of 3 in the numerator, though 3 (the number of length-2 paths in a triangle) is the more standard value.

The following are two of potentially many alternative/additional graph metrics that might be useful in the future as measures to capture local structure:

***S* metric** The S metric captures degree diversity. It is not the same as the diversity measure Macindoe and Richards used. Diversity metrics capture the topological survivability [110] and variable connectivity [3] of a graph. Is the graph dominated by a small number of large, highly connected subgraphs, or is it composed of a large number of very loosely connected nodes (or small subgraphs)?

For a graph with n vertices, let d_i be the degree of of vertex $i, 1 \leq i \leq n$ and let $D = \{d_1, d_2, \ldots, d_n\}$ be the degree sequence of the graph, where, without loss of generality, $d_1 \geq d_2 \geq \cdots d_n$. Define the S metric for graph G with degree sequence D as:

$$S(G) = \sum_{(i,j)\in E} d_i d_j = \sum_{i\in V}\sum_{j\in V} d_i a_{ij} d_j \tag{4.21}$$

where $A = \left[a_{ij}\right]$ is the vertex adjacency matrix. The S metric is also a summary measure for an entire graph.

Node Entropy The entropy of a node, sometimes referred to as diversity, is defined as the scaled Shannon entropy of the weights of the incident edges.

$$D(G) = H(i)/\log(d_i) \tag{4.22}$$
$$H(i) = -\sum_{j=1}^{d_i} p_{ij}\log(p_{ij}) \tag{4.23}$$
$$p_{ij} = w_{ij}/\sum_{k=1}^{d_i} w_{ik} \tag{4.24}$$

where d_i is the degree for node i, w_{ij} is the weight of the edge between nodes i and j.

Signature Construction We now describe our graph signatures. Our choices are heuristic, based on what worked for a particular application. We compared the graph distances we compute via the procedure described below to ground truth distances from application experts. However, any particular application will likely require changes to the details below. We discuss at least one alternative after we describe the method.

Our procedure for constructing a signature of a graph is as follows:

1. For each node v in the graph, induce the 2-hop neighborhood. This is a subgraph consisting of all nodes reachable from v by paths of length at most 2, and all edges connecting those nodes. This neighborhood defines the local graph structure at the node. Macindoe and Richards [83] chose the 2 neighborhood. We also found this distance is sufficient to capture the local structure, but is not a computational burden for the graph sizes we have tested. For large graphs with a heavy-tailed degree distribution, typical of social networks, the 1.5 neighborhood may be a more computationally reasonable subgraph. This is node v's one-hop neighborhood (neighbors and all edges joining them) plus all the neighbors' neighbors, without edges between node that are not directly connected to node v.

2. Given the induced subgraph for node i, calculate the local characteristics for this subgraph. We used three values:closeness, betweenness, and leadership. Leadership is a property of the subgraph. Closeness and betweenness are properties of a single node within a graph. For the subgraph induced by vertex i's neighborhood, we calculate closeness and betweenness for vertex i only. Thus for the graph G_i, the 2-hop neighborhood of node i, we calculate the triple (c_i, b_i, l_i). All elements of this triple are numbers between 0 and 1.

3. For a graph with n nodes, after computing the n triples as described in the step above, create three sets of values, one for each dimension: $C = \{c_i : 1 = 1 \ldots n\}$, $B = \{b_i : 1 = 1 \ldots n\}$, and $L = \{l_i : 1 = 1 \ldots n\}$.

4. Do the above steps for all m graphs, to produce m vectors $C_1, \ldots, C_m$, and similarly m vectors $B_1, \ldots, B_m$ and m vectors $L_1, \ldots, L_m$. Using k-means (or principal component analysis) identify k one-dimensional clusters for $\cup_{j=1\ldots m} C_j$. Similarly identify k one-dimensional clusters on the union of the B vectors, and k one-dimensional clusters on the union of the L vectors.

There is no need to use the same number of clusters k for each topological element, but we found, for our application, that using $k = 10$ for each set of values gives a signature that allows us to compare graphs.

5. For each graph G, count the number of points in each cluster: n_{c1}, n_{c2}, ..., n_{ck}, n_{b1}, n_{b2}, ..., n_{bk}, n_{l1}, n_{l2}, ..., n_{lk}. This set of counts, taken in this specific order, is the unique piece of the signature for graph G.

6. The signature is cluster number paired with the counts. Assuming each clustering has the same number of clusters k, then the cluster numbers for closeness are $1,\ldots,k$, the cluster numbers for betweenness are $k+1,\ldots,2k$, and the cluster numbers for leadership are $2k+1,\ldots,3k$. Thus the signature is $(1,n_{c1})$, $(2,n_{c2})$, ..., (k,n_{ck}), $(k+1,n_{b1})$, $(k+2,n_{b2})$, ..., $(2k,n_{bk})$, $(2k+1,n_{l1})$, $(2k+2,n_{l2})$, ..., $(3k,n_{lk})$.

More generally, authors represent signatures as $P = \{(x_1,p_1),\ldots,(x_m,p_m)\}$. The x_i are represent a cluster. In our signature above, this is a cluster number. But it could be the numerical value of the centroid from the k-means computation (the "average" value of points in the cluster). The p_i are the number of points in each cluster, sometimes called the *weight* of the cluster.

One alternative way to create a signature is to follow the first two steps above. Then, for each graph, use a k-means clustering of the 3-dimensional (c,b,l) vectors. More generally, use higher-dimensional k-means if the vectors of attributes/values has higher dimension. The signature is then the centroid of each cluster and the number of points in each cluster.

4.2.2 Earth Mover's Distance

Rubner, Tomasi, and Guibas [112] introducded Earth Movers Distance (EMD) as a measure of similarity between images, thereby reducing the effort to find similar images within a library of images. So far, EMD shows promise as the foundation for a graph similarity metric. The following synopsis of Ruber's formal description of EMD is from Robert Fisher's web page [29], with some corrections and clarifications.

EMD is defined for signatures of the form $P = \{(x_1,p_1),\ldots,(x_m,p_m)\}$ and $Q = \{(y_1,q_1),\ldots,(y_m,q_n)\}$ where x_i is the center of cluster i and represents the feature of interest, e.g. 'color' for an image, and p_i is the weight of cluster i, e.g. number of points of that feature type in the cluster.

EMD is conceptually a transportation problem. The "earth mover" part of the problem is to take piles of dirt, represented by P, and move them to fill in holes, represented by Q. Let $F = [f_{ij}]$ represent the flow of material between P_i (supply) to Q_j (demand). Two signatures P and Q can be compared by finding the flow F that minimizes the transportation problem:

$$Work(P,Q;F) = \left(\min_{f_{ij}} \sum_{i,j} f_{ij}\ell_{ij}\right) \qquad s.t.: \tag{4.25}$$

$$
\begin{aligned}
f_{ij} &\geq 0 \quad \text{earth can only be moved from P to Q} && (4.26)\\
\sum_j f_{ij} &\leq p_i \quad \text{the earth to be moved must no more than what is in P} && (4.27)\\
\sum_i f_{ij} &\leq q_j \quad \text{the earth to be moved must be no more than what Q can receive} && (4.28)\\
\sum_i f_{ij} &= \min\left(p_i, q_j\right) \quad \text{move the minimum amount of earth} && (4.29)
\end{aligned}
$$

Solving the transportation problem yields the optimum flow F^* which can then be used to find the Earth Movers Distance:

$$EMD(P,Q;F) = \left(\min_{f_{ij}^*} \sum_{i,j} f_{ij}^* \ell_{ij}\right) / \sum_{i,j} f_{ij}^* \tag{4.30}$$

where ℓ_{ij} is the *ground* distance between cluster i in signature P and cluster j in signature Q. The ground distance is the distance between two vectors of features, which in turn requires a distance between individual features. This distance represents the cost of turning a unit mass of one feature into a unit mass of the feature in another signature. For EMD, the ground distance between clusters is usually the ground distance between their centroids. Figure 4.4 depicts a basic transportation problem where the number of clusters in P,$m = 4$, and the number of clusters in Q, $n = 3$.

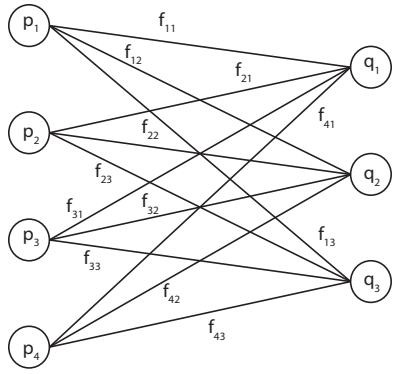


Figure 4.4. EMD as Transportation Problem

Unlike the χ^2 or Kolmogorov-Smirnov(KS) distance, which are fundamentally one-dimensional, EMD is not a bin-to-bin distance comparison between two signatures. For example, consider Figure 4.5). The K-S distance involves aligning two cumulative distribution functions over the same domain and finding the value where they differ the most. There is an inherent order, so this is like comparing items in order, illustrated on the right side of Figure 4.5. EMD has no inherent order, so any red cluster and map to any combination of blue clusters.

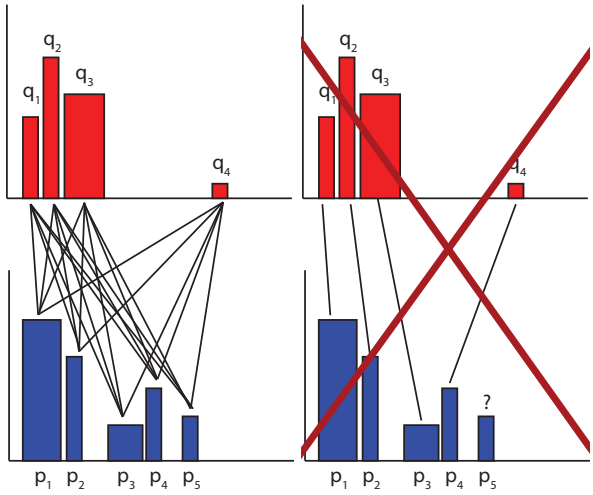


Figure 4.5. Some distance measures/statistical distance measures have an inherent order, illustrated on the right side. EMD, illustrated on the left, allows clusters (bins) to be split and to map to any other set of clusters (bins). In optimal solution, most pieces will split into only a few pieces at most.

EMD is also a true metric if the ground distance is metric and if the total weights of the two signatures are equal. This provides a metric structure for graphs and images. Rubner et. al. [112] discuss the computational advantages of using a true metric. Computationally efficient C implementations are available [79].

Ground distance

Ground distance is the distance between two cluster representatives, which must be defined for each application. When the representatives are simply vectors of numbers, variants of vector norms are popular. Since our cluster representatives are (one-dimensional) integers, our ground distance is simply the absolute value of the difference between the pair of numbers. For example, the distance between $2k+5$ and 12 is $|2k-7|$. We could define distances that more harshly penalize between clusters of, say, closeness, and clusters of betweenness. It was not necessary for our initial driving application. For values that are categorical, a subject-matter expert must generally create a table of values that appropriately represents how close two possible values are. As a simple example, biologists have quantified how closely related each pair of amino acids is based on how easily one can evolve into another in a protein sequence.

Rubner et. al. [111, 112] originally suggested Euclidean distance in R^s (the L_2-norm) as an appropriate ground distance ℓ_{ij} between image color signatures. Since color is a number, one can use the normal notion of arithmetic. Texture is more challenging. Kauchak's lecture [64]

summarizes papers from authors including Rubner and Tomasi. He says the Gabor Filters can be appropriate distance metrics for texture.

There is no universally accepted ground distance metrics for all settings, but the dominant metrics used are L_d-norms:

$$L_d(p,q) = \left[\sum_{i=1}^{n} |p_i - q_i|^d\right]^{1/d}$$

where $p = \{p_1, \ldots, p_n\}$ and $q = \{q_1, \ldots, q_n\}$. For "dimension" $0 < d < 1$ the norm is a not a metric since it violates the triangle inequality. However, for $d \geq 1$ the L_d-norm is referred to as the Minkowski metric and is the basis for many ground distance measures.

Statistical Similarity

Levina and Bickel [77] demonstrated that when the ground distance is a valid metric and the two signatures have equal total weight (and can therefore be normalized) then EMD is equivalent to Mallows distance. Mallows distance is a statistical measure used to characterize the similarity between distributions, in much the same way as the χ^2 test.

Assume for now that the signatures P and Q are normalized and therefore depict probability distribution functions. Define the random variables: $X \sim P$, $Y \sim Q$ and the joint distribution between X and Y is defined: $F \sim (X,Y)$, where $X \sim P$ means that random variable X is distributed according to distribution P . Mallows distance between the distributions P and Q is then defined as the minimum expected difference between X and Y, taken over all valid joint distributions F:

$$M_d(P,Q) = \min_F\{(E_F||X-Y||^d)^{1/d} : (X,Y) \sim F, X \sim P, Y \sim Q\} \tag{4.31}$$

Table 4.2. EMD - Normalized Signatures

	0	1	2	3	Y
0	0.25	0.25	0	0	0.5
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0.25	0.25	0.5
X	0.25	0.25	0.25	0.25	

Consider Table 4.2. The far right column represents the marginal distribution of Y and the last row represents the marginal distribution of X. The interior of the table represents the joint distribution F that we are trying to find. Our goal is to find the joint distribution, subject to the constraints that the joint and marginal distributions must be valid, i.e. sum to one. The expectation function

in Equations 4.31 is identical to the transportation problem described in Equation 4.25, while the row and column constraints also have equivalent counterparts in the transportation problem.

As noted previously, for signatures with the *same total mass*, the EMD is a true metric on distributions, and it is identical to the Mallows distance. Normalizing signatures with the same mass does not affect their EMD. However, EMD on signatures is not invariant to weight scaling unless both signatures are scaled by the same factor. Levina and Bickel [77] describe this issue in more detail.

Mallows distance between empirical distributions is given by:

$$M_d(P,Q) = \left(\frac{1}{n} \min_{(j_1,j_2,\ldots,j_n)} \sum_{i=1}^{n} |p_i - q_j|^d \right)^{1/d}$$

where the minimum is taken over all permutation of $\{1,\ldots,n\}$. That is, we consider all possible ways to match one cluster of P to one cluster of Q.

For one dimensional signatures, the Hungarian assignment algorithm, as a special case of the transportation problem, solves the problem. let $p_{(1)} \leq \ldots p_{(n)}$ and $q_{(1)} \leq \ldots q_{(n)}$ be the ordered signature values. The Mallows distance is the L_d distance between the ordered vectors:

$$M_d(P,Q) = \left(\frac{1}{n} \sum_{i=1}^{n} |p_{(i)} - q_{(i)}j|^d \right)^{1/d}$$

Issues with EMD

All is not rosy, however, in the case where the signatures are not of equal mass. If one signature is a partial match for another signature, then a degenerate situation develops.

The following example is from Levina and Bickel[77]. Consider the two signatures in Figure 4.6. Q is a partial match for P because $q_1 = p_1$ and $q_4 = p_4$. First consider the case where both signatures are normalized. Then $Q = \{(1,0.5),(4,0.5)\}$ and $P = \{(1,0.25),(2,0.25),(3,0.25),(4,0.25)\}$. Then, using L_1 as the ground distance, we find that $M_1(P,Q) = EMD_1(P,Q) = 0.5$. Alternatively, if the signatures are not normalized and again use L_1 as the ground distance, we find that $EMD(P,Q) = 0$. In fact, the EMD remains zero even if we add an arbitrary number of new clusters (bins) to P, because Q will still be an exact partial match.

Alternative EMD Formulations

As noted previously, EMD is a true metric for normalized signatures. True metric can lead to more efficient data structures and search algorithms. When the two signatures to be compared are similar in size then EMD behaves as an approximate metric. Pele and Werman[97] suggested a variation

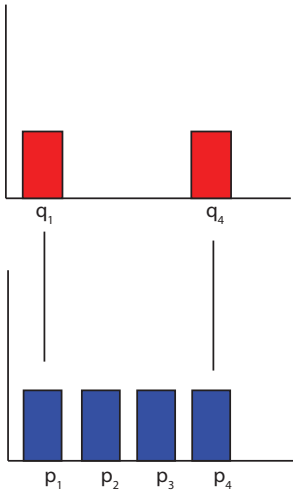


Figure 4.6. EMD Degenerate Case

of EMD that penalizes for unequal sizes. Assuming, without loss of generality, that $\sum p_i \geq \sum q_j$ introduce an additional infinite "demand" that absorbs the excess "supply" from P.

As before, solving the transportation problem (Equation 4.25) yields the optimum flow F^*. EMD* is a variant of EMD that heavily penalizes the difference in sizes. EMD* is not normalized. Thus multiplying EMD by the normalization in the first term of Expression 4.33 retrieves the unnormalized EMD, which is then augmented with the penalty.

$$EMD^*(P,Q;F) = \left(\min_{f^*_{ij}} \sum_{i,j} f^*_{ij}\ell_{ij}\right) + |\sum_i P_i - \sum_j Q_j| \times \alpha \max_{i,j}\{\ell_{ij}\} \quad (4.32)$$

$$= \left(EMD * \sum_{i,j} f_{ij}\right) + |\sum_i P_i - \sum_j Q_j| \times \alpha \max_{i,j}\{\ell_{ij}\} \quad (4.33)$$

where, if $\alpha \geq 0.5$ and the ground distance is a metric, then EMD* is a metric. This alternative provides relief in two important situations: first is when the total mass of the signatures is important and second, when the mass difference between the signatures is important. One could define a lesser penalty for unequal weights. However, for our application, the maximum distance gives acceptable behavior.

Here are the primary differences between the method described in this section and the work of McIndoe and Richards [83]. In this list M-R is short for Macindoe-Richards.

- MR use histograms with fixed spacing, which is inefficient as described above. We use variable-width bins to capture structure information, using signatures instead of classic histograms.

- M-R require normalized histograms. We do not. Normalization is not necessary and using it can lose critical information. In reality, the only requirement is that either the signatures are of roughly equal volume or we use the alternative EMD metric to penalize size differences. This permits comparison of graphs of vastly different sizes.
- M-R require that the structural metrics be statistically independent. In our method signatures can be multi-dimensional and statistically correlated.

EMD as used in this section is a formal metric that measures the information difference between two graphs. The similarity metric is equivalent to the Akaike Information Criteria. This metric explicitly accounts for error introduced when comparing small graphs. (This is still being tested in practice). One limitiation of the EMD metric is that it does not permit formal hypothesis testing of similarity.

EMD Example In this section, we give a simple one-dimensional example of EMD and EMD*. Consider the two one-dimensional signatures in Figure 4.7 and assume the L_1-norm (Manhattan) is the ground distance. The solution to the associated transportation problem outlined in Equation 4.25 is summarized in Table 4.3. The bold-type numbers in the last column are the weights for Signature 1, shown in green in Figure 4.7. The bold-type numbers in the last row are the weights for Signature 2, shown in red in Figure 4.7. Entry (i, j) of the table holds flow value f_{ij} from Signature 1 to Signature 2.

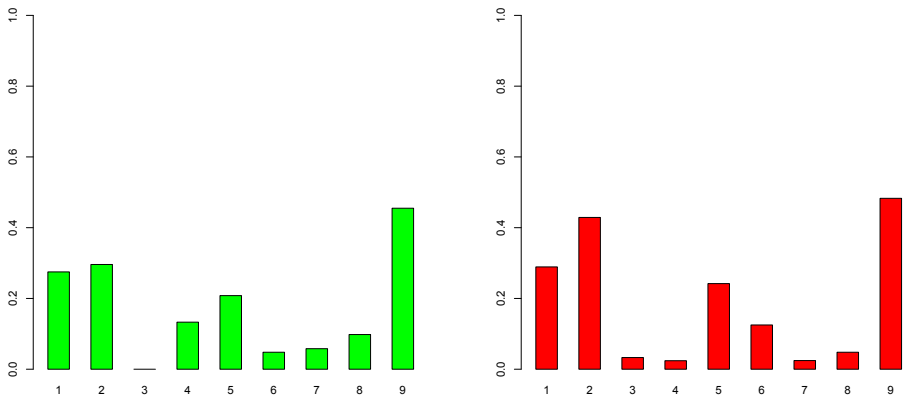


Figure 4.7. Signatures for EMD Comparison

Table 4.3. EMD Calculation for Signatures in Figure 4.7

	1	2	3	4	5	6	7	8	9	**Signature 1**
1	0.275									**0.275**
2		0.296								**0.296**
3			0.002							**0.002**
4		0.052	0.026	0.021	0.032					**0.131**
5					0.208					**0.208**
6						0.048				**0.048**
7						0.035	0.023			**0.058**
8						0.03		0.054	0.014	**0.098**
9									0.455	**0.455**
Signature 2	**0.285**	**0.421**	**0.028**	**0.021**	**0.240**	**0.166**	**0.023**	**0.054**	**0.469**	

The total flow from Signature 1 to Signature 2 is $\sum_{i,j=1,\ldots 9} f_{ij} = 1.571$, which is the total weight of Signature 1. To compute EMD, we consider only the f_{ij} values where $i \neq j$. Flow $f_{42} = 0.052$ contributes $2 * 0.052$ to EMD because the L_1 distance between 4 and 2 is $4 - 2 = 2$. Summing over all off-diagonal elements, we have $EMD = 0.173$. Since the total weight of Signature 2 is 0.136

larger than the total weight of Signature 1, the excess 'supply' is absorbed in the $EMD*$ distance. Since the maximum distance is 8 and $\alpha = 1$, we have:

$$EMD^*(P,Q;F) = \left(EMD * \sum_{i,j} f_{ij} \right) + |\sum_i P_i - \sum_j Q_j| \times \alpha \max_{i,j}\{d_{ij}\} = 1.36 \tag{4.34}$$

4.2.3 Hierarchical Cluster Analysis

Given a measure of pairwise distances, we can use hierarchical cluster analysis to heuristically compute similarities among larger groups of graphs. The algorithm begins with each object (graph in this case) by itself and a matrix of pairwise distances. At each step, the algorithm finds the two objects that are closest according to the distance measure (hence, most similar in some way), and merges them into a new larger cluster. It then updatess the distance matrix by computing distances between the new merged object and the other objects. A dendrogram, such as the one shown in Figure 4.10 can represent this process, where each merge combines two branches. Usually the merges are drawn "bottom up," so the primitive graphs are leaves of the tree, and internal nodes represent the merging of the nodes on the left and right sides. Looking "top down," removing the top horizontal bar breaks the set into two pieces. These pieces were the last to merge. We will describe below how to interpret rooted dendrograms such as the one in Figure 4.10. Unrooted dendrograms, such as the one in Figure 4.9, are easier to interpret. Edge distances are proportional to distances between clusters. We focus on the dendrograms based on a precomputed distance or similarity matrix. Alternative methods, which we do not consider here, include k-means, or fuzzy clustering. We emphasize that dendrograms are a heuristic way to view the set of graphs, approximately grouping graphs in each branch that are more similar to each other than to the rest of the graphs.

Given an initial set of pairwise distances, there are a number of higher-level ways to compare distances among sets of objects. That choice determines how to update the distance matrix after a merge. The user must choose the best method based upon a particular application (that is, what the graphs and feature vectors represent). Sometimes this just involves trial and error, though an experienced user may gain insight over the course of many different applications. A non-exhaustive list of higher-level combination methods include: Ward's, single-link, complete-link, average, McQuitty, median, or centroid. These are the set of combination methods availabe in the R software package. We use the *hclust* method from the R statistical package to produce the dendrograms in this report.

In single-link hierarchical agglomerative clustering, the distance between two clusters is based completely on the similarity of the two most similar members (shortest distance). That is, given clusters C_1 and C_2, the distance between them is the minimum distance between a primitive member $c_1 \in C_1$ and a primitive member $c_2 \in C_2$. This is a local criteria. It tends to produce elongated clusters that are sensitive to noise and outliers. In contrast, complete-link clustering bases the group distance on the maximum distance between a member of one cluster and a member of the other. The details of the other methods are more complex. We have directly copied Müllner's [84]

excellent descriptive table into Figure 4.8.

To appreciate how dramatic the difference between clustering methods can be, Figure 4.9 depicts unrooted dendrograms created from the same distance matrix. The dendrogram on the left is based on Ward's distance update method and the dendrogram on the right is based on the complete (maximum distance) method.

Example

We now give a simple intuitive example. Table 4.4 presents the Euclidean distance between ten major U.S. cities. This matrix is analogous to our similarity matrix constructed using the Earth Movers Distance metric.

	Atlanta	Chicago	Denver	Houston	LosAngeles	Miami	NewYork	SanFrancisco	Seattle	WashingtonDC
Atlanta	0	587	1212	701	1936	604	748	2139	2182	543
Chicago	587	0	920	940	1745	1188	713	1858	1737	597
Denver	1212	920	0	879	831	1726	1631	949	1021	1494
Houston	701	940	879	0	1374	968	1420	1645	1891	1220
LosAngeles	1936	1745	831	1374	0	2339	2451	347	959	2300
Miami	604	1188	1726	968	2339	0	1092	2594	2734	923
NewYork	748	713	1631	1420	2451	1092	0	2571	2408	205
SanFrancisco	2139	1858	949	1645	347	2594	2571	0	678	2442
Seattle	2182	1737	1021	1891	959	2734	2408	678	0	2329
WashingtonDC	543	597	1494	1220	2300	923	205	2442	2329	0

Table 4.4. City Distances

The first step in the hierarchical clustering process is to find the pair of samples (cities) that are the most similar, that is are the closest in Euclidean distance. This pair is Washington D.C. and New York with a distance of 205. We join them with a node (shown as a line) a height 205 in Figure 4.10. We must replace the two elements in the distance matrix with a merged element, and compute its distance to all other elements. For this example, we use the *complete* or maximum-distance agglomeration method. For example, to compute the distance from Chicago to the NY-DC cluster, we consider the distance between Chicago and each city in the cluster. The distance between Chicago and New York is 713, while the distance between Chicago and Washington D.C. is 597. The maximum is 713, so this becomes the distance between Chicago and the NY-DC cluster. Table 4.5 gives the new distance matrix.

After agglomeration, we search the new distance matrix for the two closest objects (groups of one or more cities). The closest pair is Los Angeles and San Francisco with a distance of 347. We put a connection node between these two cities at height 347. The results of this agglomeration are presented in Table 4.6. This procedure of finding the closest pair, agglomerating, and then updating with maximum over all pairs across a cluster, continues until all leaves are joined into a single connected dendrogram. The final series of agglomerations is summarized in Tables 4.7 through 4.12.

By choosing the maximum (complete) combination method, cities outside a cluster consider all nodes inside to be as far away as the maximum. Consider the line on the dendrogram in Figure 4.10 that joins Miami on the left to a cluster of New York, Washington DC, Atlanta, and Chicago on the right. This line represents the seventh agglomeration. That is, Table 4.10 represents the distance

Name	Distance update formula for $d(I \cup J, K)$	Cluster dissimilarity between clusters A and B
Single	$\min(d(I,K), d(J,K))$	$\min_{a \in A, b \in B} d[a,b]$
Complete	$\max(d(I,K), d(J,K))$	$\max_{a \in A, b \in B} d[a,b]$
Average	$\frac{n_I d(I,K) + n_J d(J,K)}{n_I + n_J}$	$\frac{1}{\lvert A \rvert \lvert B \rvert} \sum_{a \in A} \sum_{b \in B} d[a,b]$
Weighted/McQuitty	$\frac{d(I,K) + d(J,K)}{2}$	
Ward	$\sqrt{\frac{(n_I + n_K) d(I,K)^2 + (n_J + n_K) d(J,K)^2 - n_K d(I,J)^2}{n_I + n_J + n_K}}$	$\sqrt{\frac{2\lvert A \rvert \lvert B \rvert}{\lvert A \rvert + \lvert B \rvert}} \cdot \lVert \vec{c}_A - \vec{c}_B \rVert_2$
Centroid	$\sqrt{\frac{n_I d(I,K)^2 + n_J d(J,K)^2}{n_I + n_J} - \frac{n_I n_J d(I,J)^2}{(n_I + n_J)^2}}$	$\lVert \vec{c}_A - \vec{c}_B \rVert_2$
Median	$\sqrt{\frac{d(I,K)^2}{2} + \frac{d(J,K)^2}{2} - \frac{d(I,J)^2}{4}}$	$\lVert \vec{w}_A - \vec{w}_B \rVert_2$

Table 1: Agglomerative clustering schemes. Let I, J be two clusters joined into a new cluster, and let K be any other cluster. Denote by n_I, n_J and n_K the sizes of (i.e., number of elements in) clusters I, J, K, respectively.
The update formulas for the "Ward", "Centroid" and "Median" methods assume that the input points are given as vectors in Euclidean space with the Euclidean distance as dissimilarity measure. The expression $\vec{c}_X$ denotes the centroid of a cluster X. The point $\vec{w}_X$ is defined iteratively and depends on the order of clustering steps: If the cluster L is formed by joining I and J, we define $\vec{w}_L$ as the midpoint $\frac{1}{2}(\vec{w}_I + \vec{w}_J)$.

Figure 4.8. A table explaining the agglomerative combination methods in R taken directly from Müllner [84]. $d[a,b]$ is the distance between points/elements/clusters a and b.

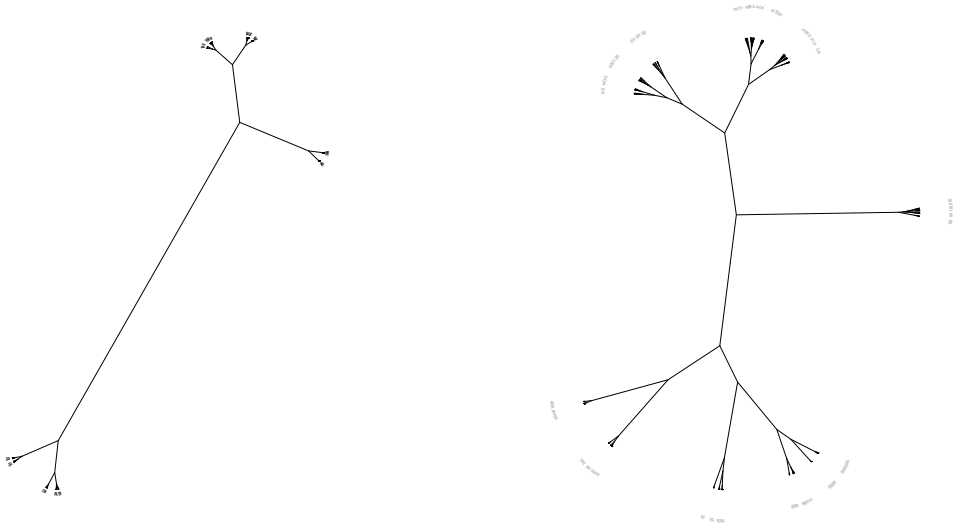


Figure 4.9. Clustering: Ward's Method (left) vs Complete Method (right) from the same distance matrix.

matrix before this agglomeration. The smallest entry in this matrix is 1188, the distance between Miami and the Chicago-Atlanta-NY-Washington DC cluster. Because we used the maximum distance, we consider NY to be as close to Miami as Chicago (the farthest away in the 4-city cluster). The line joining Miami to the 4-city cluster is at height 1188. This means that all four cities are no more than 1188 miles from Miami.

To illustrate the effect of combination rule, we also show the dendrograms based on single or minimum (Figure 4.11) , average (Figure 4.12) , median (Figure 4.14) , Ward (Figure 4.15) , and McQuitty (Figure 4.16) agglomeration methods. We used the hierarchical cluster method in the R statistical package (the hclust) method, which accepts a combination rule as a parameter. The minimum decision rule is seldom used since it generally results in clusters of leaves that are more heterogeneous than is desired when performing clustering. The average method is a common alternative in which the similarities are, as the name implies, averaged at each step.

Table 4.5. First Agglomeration

	Atlanta	**Chicago**	**Denver**	**Houston**	**LosAngeles**	**Miami**	**SanFrancisco**	**Seattle**	**NY-WDC**
Atlanta	0	587	1212	701	1936	604	2139	2182	748
Chicago	587	0	920	940	1745	1188	1858	1737	713
Denver	1212	920	0	879	831	1726	949	1021	1631
Houston	701	940	879	0	1374	968	1645	1891	1420
LosAngeles	1936	1745	831	1374	0	2339	347	959	2451
Miami	604	1188	1726	968	2339	0	2594	2734	1092
SanFrancisco	2139	1858	949	1645	347	2594	0	678	2571
Seattle	2182	1737	1021	1891	959	2734	678	0	2408
NY-WDC	748	713	1631	1420	2451	1092	2571	2408	0

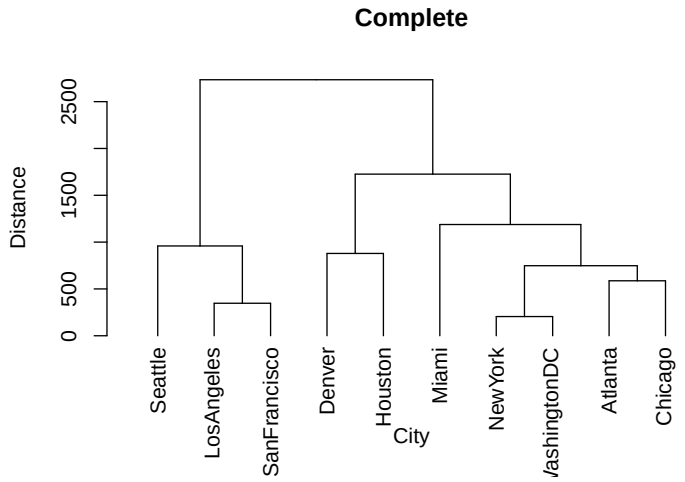


Figure 4.10. Dendrogram (Complete, or maximum)

Table 4.6. Second Agglomeration

	Atlanta	**Chicago**	**Denver**	**Houston**	**San-LA**	**Miami**	**Seattle**	**NY-WDC**
Atlanta	0	587	1212	701	2139	604	2182	748
Chicago	587	0	920	940	1858	1188	1737	713
Denver	1212	920	0	879	949	1726	1021	1631
Houston	701	940	879	0	1645	968	1891	1420
San-LA	2139	1858	949	1645	0	2594	959	2571
Miami	604	1188	1726	968	2594	0	2734	1092
Seattle	2182	1737	1021	1891	959	2734	0	2408
NY-WDC	748	713	1631	1420	2571	1092	2408	0

Table 4.7. Third Agglomeration

	Ch-At	**Denver**	**Houston**	**San-LA**	**Miami**	**Seattle**	**NY-WDC**
Ch-At	0	1212	940	2139	1188	2182	748
Denver	1212	0	879	949	1726	1021	1631
Houston	940	879	0	1645	968	1891	1420
San-LA	2139	949	1645	0	2594	959	2571
Miami	1188	1726	968	2594	0	2734	1092
Seattle	2182	1021	1891	959	2734	0	2408
NY-WDC	748	1631	1420	2571	1092	2408	0

Table 4.8. Fourth Agglomeration

	Ch-At-NY-WDC	Denver	Houston	San-LA	Miami	Seattle
Ch-At-NY-WDC	0	1631	1420	2571	1188	2408
Denver	1631	0	879	949	1726	1021
Houston	1420	879	0	1645	968	1891
San-LA	2571	949	1645	0	2594	959
Miami	1188	1726	968	2594	0	2734
Seattle	2408	1021	1891	959	2734	0

Table 4.9. Fifth Agglomeration

	Ch-At-NY-WDC	Den-Ho	San-LA	Miami	Seattle
Ch-At-NY-WDC	0	1631	2571	1188	2408
Den-Ho	1631	0	1645	1726	1891
San-LA	2571	1645	0	2594	959
Miami	1188	1726	2594	0	2734
Seattle	2408	1891	959	2734	0

Table 4.10. Sixth Agglomeration

	Ch-At-NY-WDC	Den-Ho	Miami	Sea-San-LA
Ch-At-NY-WDC	0	1631	1188	2571
Den-Ho	1631	0	1726	1891
Miami	1188	1726	0	2734
Sea-San-LA	2571	1891	2734	0

Table 4.11. Seventh Agglomeration

	Mi-Ch-At-NY-WDC	Den-Ho	Sea-San-LA
Mi-Ch-At-NY-WDC	0	1726	2734
Den-Ho	1726	0	1891
Sea-San-LA	2734	1891	0

Table 4.12. Final Agglomeration

	Den-Ho-Mi-Ch-At-NY-WDC	**Sea-San-LA**
Den-Ho-Mi-Ch-At-NY-WDC	0	2734
Sea-San-LA	2734	0

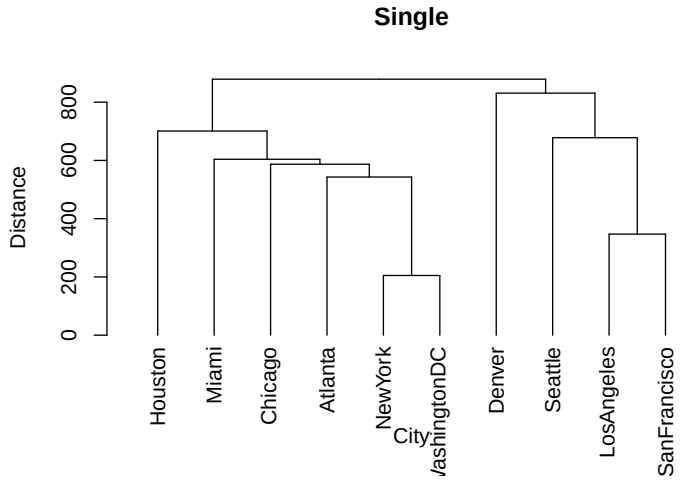


Figure 4.11. Dendrogram (Single or Minimum)

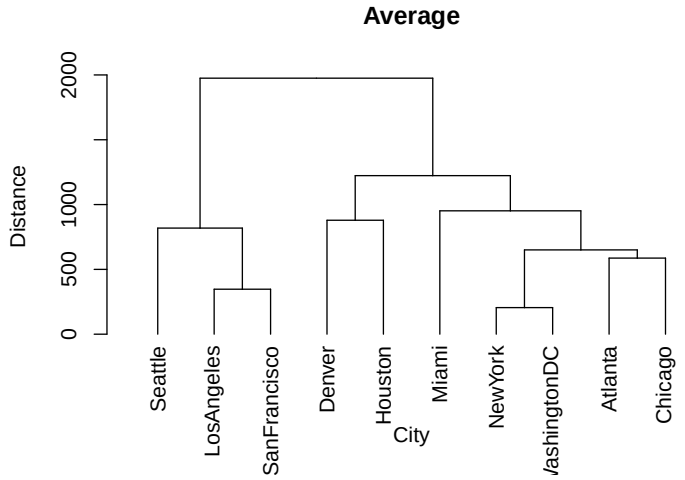


Figure 4.12. Dendrogram (Average)

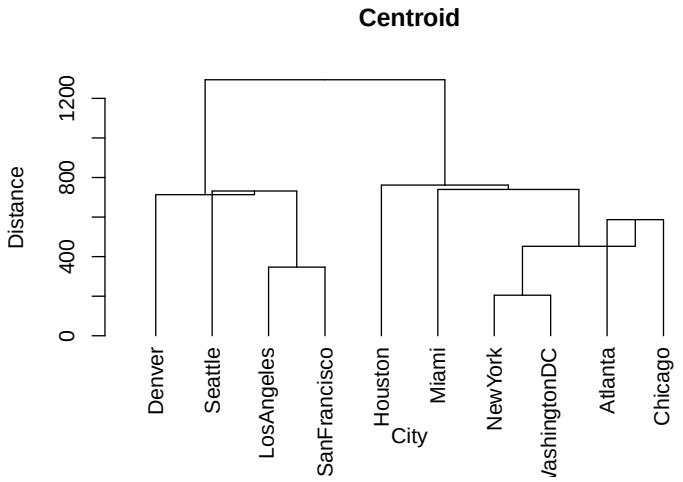


Figure 4.13. Dendrogram (Central)

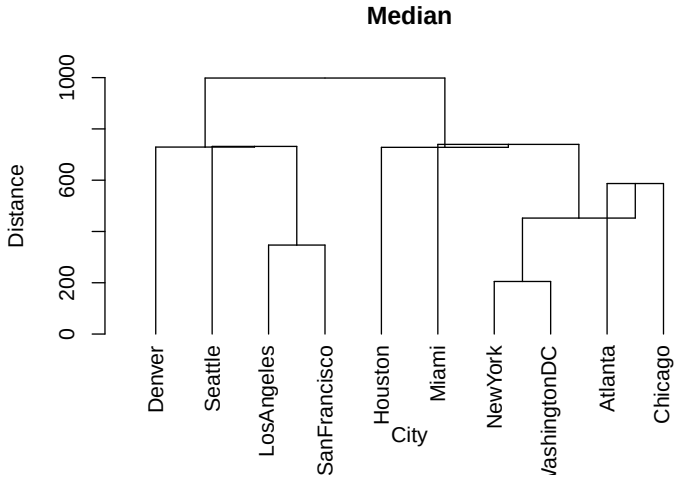


Figure 4.14. Dendrogram (Median)

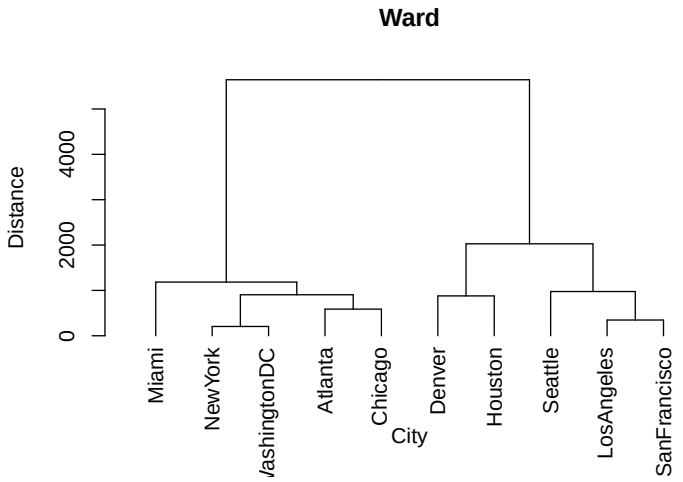


Figure 4.15. Dendrogram (Ward)

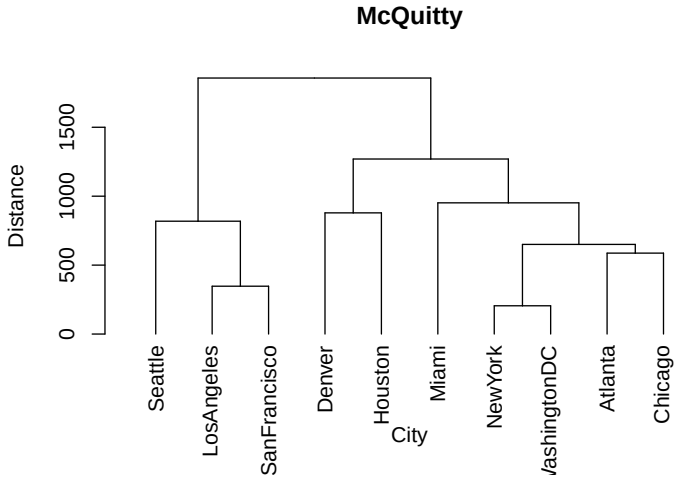


Figure 4.16. Dendrogram (McQuitty)

Cautions

Phylogentic trees are a type of dendrogram that highlights the physical ancestry of the leaves. Our dendrograms represent the similarity between graphs by way of the EMD metrics. The dendrograms that result depict, depending on the application, the similarity between the structure and attributes of the graphs.

Agglomeration of leaves in no way implies that one graph is either an exact or approximate subgraph of a merged set of leaves as would be the case for a phylogenetic tree.

4.2.4 Experiments

In this section we describe experiments with artificial data we can control, and we summarize our application of this method to real data.

Tests with Eros-Renyi Graphs

To demonstrate the use of EMD as a similarity metric, we generated a truth set of graphs and used the procedure described above to cluster the graphs into communities with similar structural characteristics. The truth set consisted of 200 randomly sampled, undirected Erdős-Rényi graphs (see Section 5.1). We selected the number of vertices randomly from the set $N \in \{50, 75,100, 300, 500, 750, 1000, 1500, 2000, 3000\}$ and selected the mean degree randomly from the set $\bar{d} \in \{1,2,\ldots,10\}$. The probability of a connection between any two nodes in a graph is therefore approximately: $p = \bar{d}/N$.

We constructed six sets of 200 graphs using different initial random seeds and then randomly selected 200 graphs from the 1200 graphs we generated. We generally had multiple instances of graphs generated with the same pair of parameters to represent variation in the graph structure even with identical combination of $\{N,d\}$. We label each sampled graph as $gi-N-\bar{d}$, where the subscripts i, N, and d refer to, respectively, a serial number for graphs with the same parameters, number of vertices, and the average degree used to generate the graph.

We generated the pair-wise similarity metric (EMD) for all pairs of the 200 graphs, creating a 200×200 distance matrix characterizing the similarity between the graphs. Figure 4.19 shows the dendrogram for this distance matrix based on Ward's method for combining clusters. Another way to visualize the dendrogram distances is the heat map, the colored matrix in Figure 4.19. There is a matrix square for each pair of graphs (G_1, G_2) with a coloring representing the dendrogram distance between the two graphs. This is the height of the node created when those two graphs first merged (that is, when a cluster holding graph G_1 first merged with a cluster containing graph G_2). Thus in the city-based example in Section 4.2.3, the distance from Miami to each of the four cities (Chicago, New York, Atlanta, and Washington DC) is 1188. All four squares have the same value. The heat map is symmetric across the diagonal. Darker red indicates higher similarity

(lower distance) and lighter colors indicate higher dissimilarity.

Figure 4.17 depicts a simple dendrogram of the clusters of our 200 samples of Erdős-Rényi graphs using Ward's method. Figure 4.18 depicts an enlarged perspective of the lower right of this figure. Generally graphs of similar size, N, and similar sparseness, e.g. similar average degree d tend to be close in the tree. In addition, graphs with the same characteristics from different simulation runs are also generally clustered. There is no consensus among researchers what the "correct" dendrogram should be. So methods for validating this kind of experiment are an interesting open question.

Figures 4.20 and 4.21 show a dendrogram and its heat map for the same Erdős-Rényi graphs, but using the complete metric instead of Ward's.

A Real Application

While the Erdös-Rényi-based clustering provides some measure of the success of the approximate matching algorithm, a stronger demonstration of the potential of the algorithm is its use as a library search tool. Consider the situation where we have a new graph, observed in the "wild". We wish to find similar graph(s) from within a known library of graphs. The library might be associated with patterns of network behavior or cellular function. A graph from the wild might represent an unknown, potentially toxic cellular function. We can search the library for graphs with similar structural characteristics with the hope of gaining insight into the nature of the observed graph. Figure 4.22 depicts the results of a sample retrieval where the library is queried for the three closest matches to the "wild graph." Again, the size and mean degree of the retrieved graphs are similar to the characteristics of the unknown graph. The search also gives the user estimates of the distance (dis-similarity) between the observed graph and those in the library.

Other Applications

Figure 4.23 depicts another application where there is benefit characterizing the similarity between graphs. Graphs based on network traffic are noisy: servers drop off, isolating segments of the network, websites are down for maintenance, users are absent due to travel, etc. The result is that tracking or monitoring network traffic to identify anomalous behavior can be difficult if this noise is not considered. The availability of a network similarity measure provides a statistical metric similar to that used in production control charts to detect significant deviations of the network from 'normal operation' without having to explicitly characterize what is meant by 'normal'.

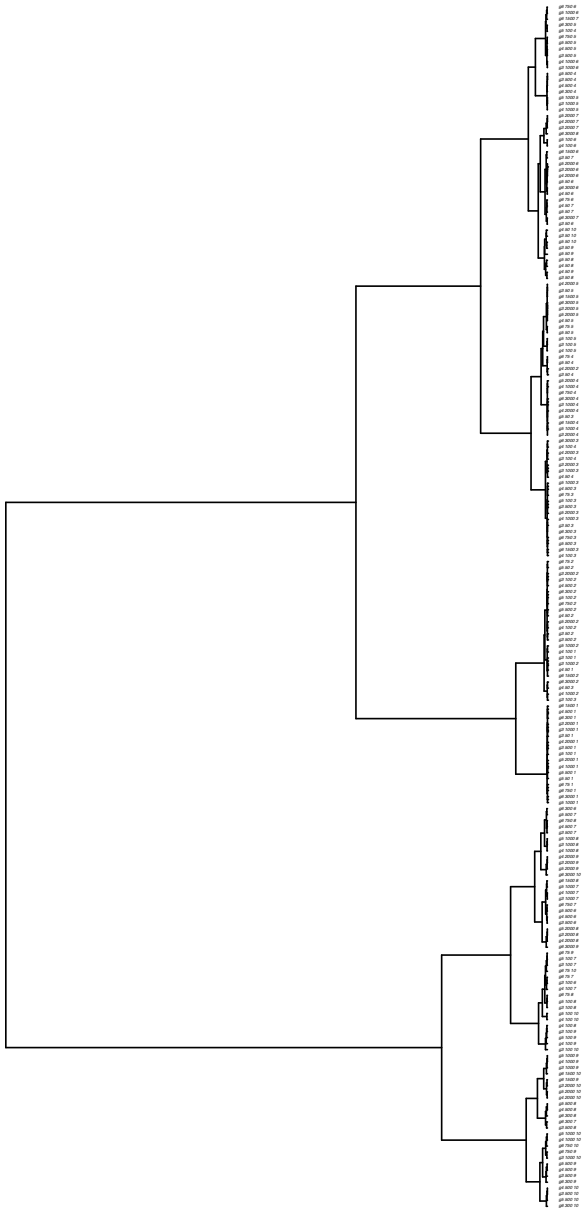


Figure 4.17. Dendrogram Clustering of Erdős-Rényi Graphs

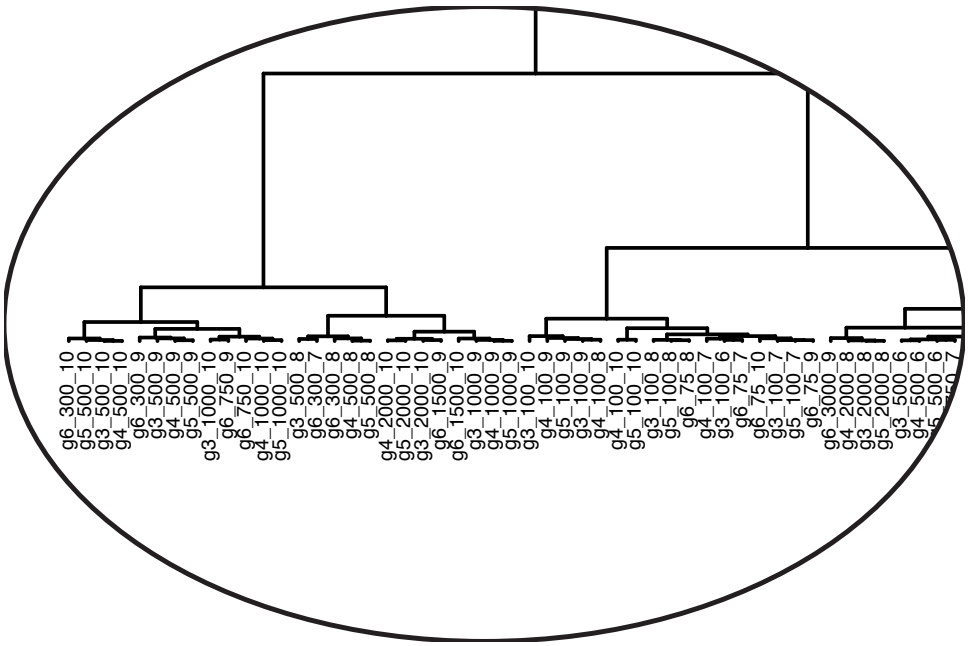


Figure 4.18. Dendrogram Clustering of Erdős-Rényi Graphs

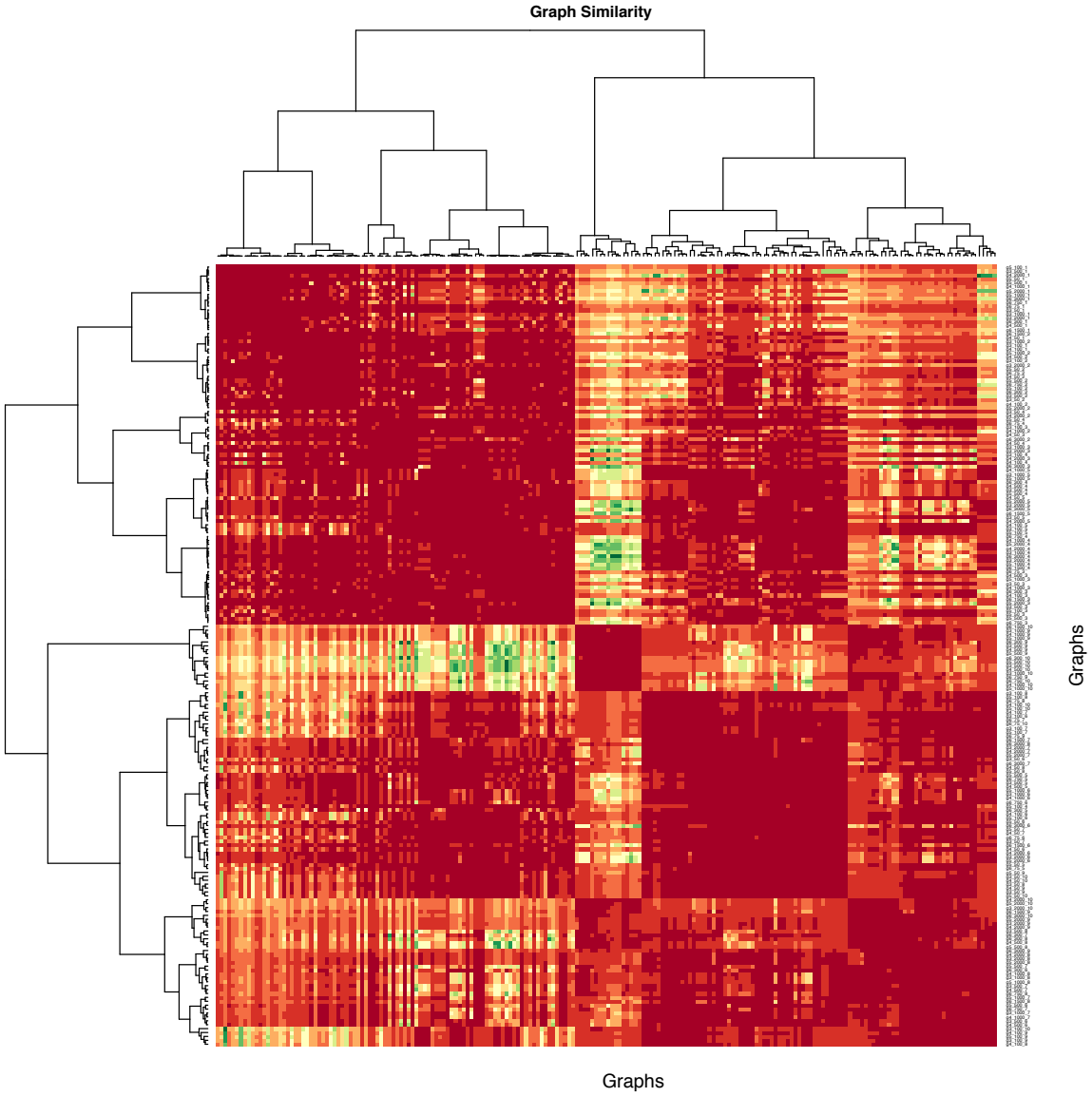


Figure 4.19. Heatmap of Similarity Clustering (Ward's method)

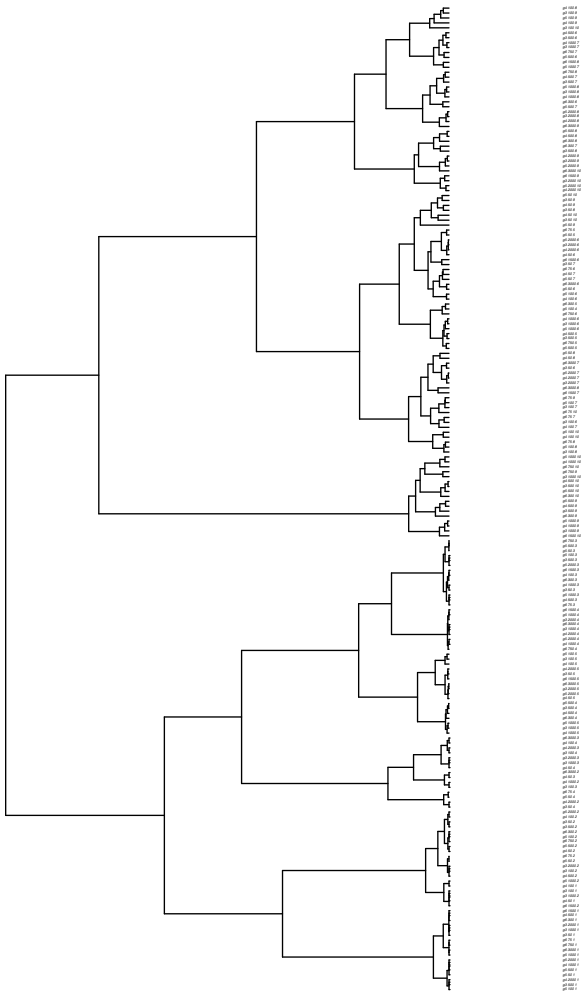


Figure 4.20. Dendrogram Clustering of Erdős-Rényi Graphs - Complete

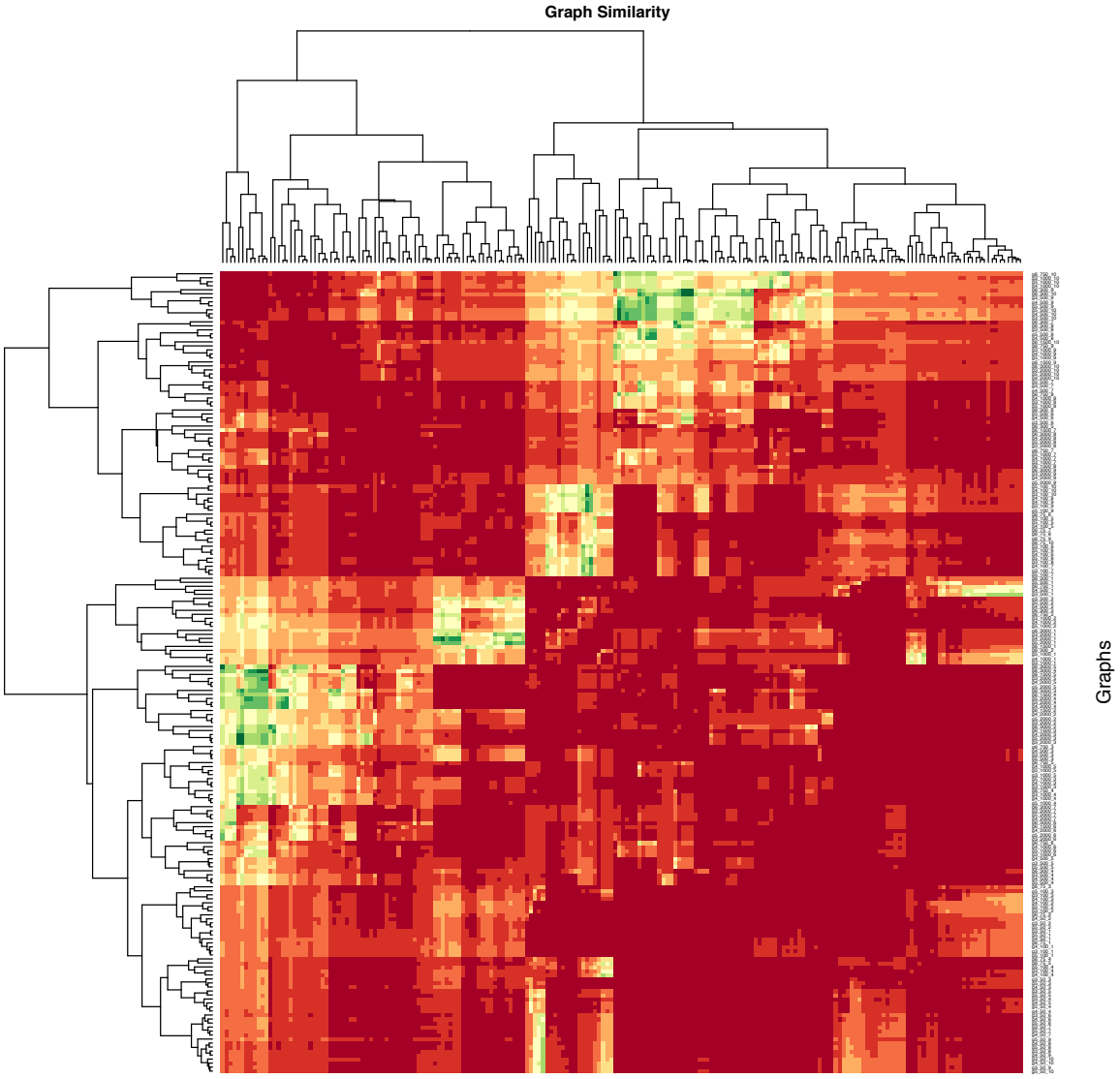


Figure 4.21. Heatmap of Similarity Clustering- Complete

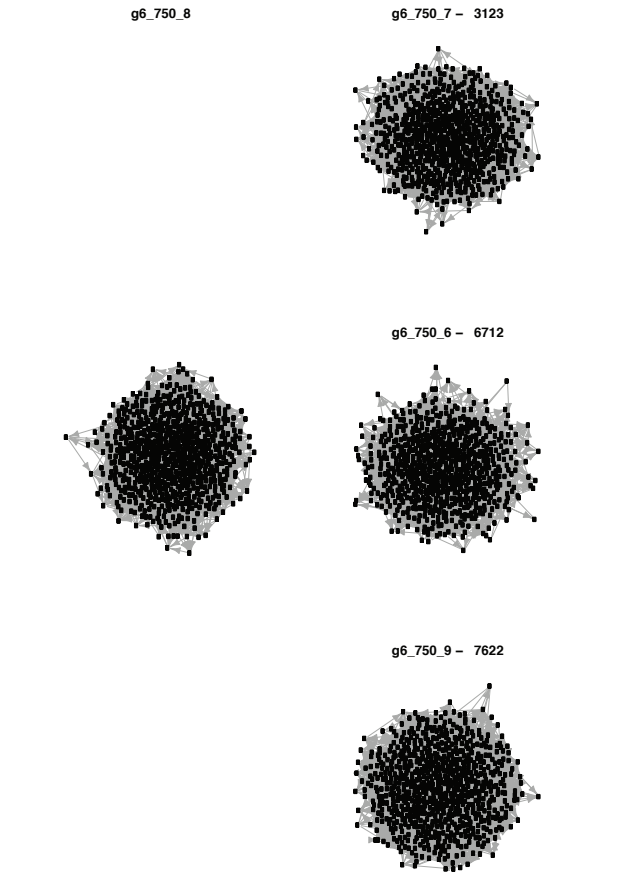


Figure 4.22. Library Search

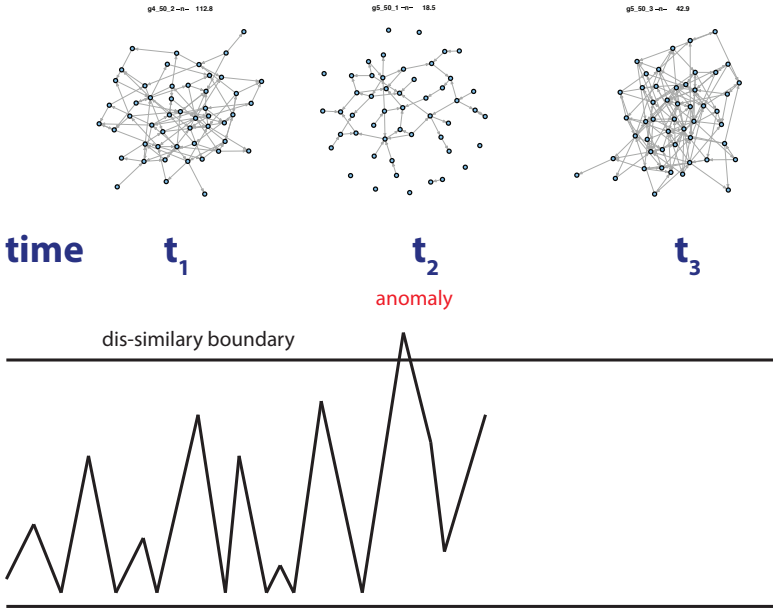


Figure 4.23. Anomaly Detection Via Similarity Metric Monitoring

Chapter 5

The Local Structure Graph Model

This section describes a new probabilistic graph model, the local structure graph model (LSGM). This material will form the background material and the first of three or more technical chapters in Emily Casleton's PhD thesis at Iowa State University. This LDRD project supported Emily's research for two years.

The wide variety of applications from various disciplines for which networks have been employed demonstrates the flexibility of networks as a modeling tool. However, this diverse array of problems has led to a diverse array of solutions from a diverse array of disciplines. Many of the models or methods developed for the analysis of networks have been specific to the application or worse, observed network, and thus most are ad hoc and not appropriate for other types of observed networks [128]. In this section, we present and explore some attributes of a newly-constructed model for the statistical analysis of observed graphs, the *local structure graph model (LSGM)*. We did not develop the LSGM with a specific application in mind and thus it does not suffer from the over-fitting that is sometimes seen in current graph analysis methods. However, the model is extremely flexible and should be applicable to many applications.

Snijders et. al. [120] described the ERGM and a subclass of the latent variable models, the latent space models, as the two competing models for probabilistic modeling and statistical analysis of social networks. Although specific applications may be well-suited for one approach or the other, there are many instances where both would be applicable. Each method presents advantages and challenges with a short list presented in Table 5.1. Snijders et. al. [120] call models that integrate the advantages of both approaches while minimizing the difficulties "the next generation" of models. The LSGM is a member of this new wave of graph analysis models.

Then Section 5.2 provides an overview of the newly-developed model for graph analysis, the LSGM. The LSGM is a probabilistic graph analysis technique, with similarities and differences between both of the existing statistical modeling techniques: the ERGM and the latent variable approaches. In Section 5.1, provides relevant background about other random graph analysis methods and models including ERGM and latent variable approaches.

The two distinguishing features of the LSGM are a conditional specification and a neighborhood definition. Similar to the latent variable approach, the model is specified through full conditional binary distributions; however, in the LSGM the random variables of interest, i.e., the edge variables, are not conditionally independent. Rather they exhibit a Markov dependence where edges are conditionally dependent on edges that belong to the same neighborhood. Thus, the defi-

	Advantage	Disadvantage
ERGM	⋆ Scientific justification of statistics ⋆ Incorporate many network features	⋆ Degeneracy ⋆ Likelihood intractable ⋆ Cannot account for unobserved structure
LVM	⋆ Computationally tractable ⋆ Not degenerate ⋆ Can handle data that is missing at random	⋆ Cannot model some dependencies, such as transitivity

Table 5.1. Comparison of the strengths and weaknesses of the two probabilistic modeling approaches to network analysis: the Exponential Random Graph Model (ERGM) and Latent Variable Models (LVM).

nition of neighborhoods defines the dependence structure. The neighborhood structure explored in this section is incidence, or two edges are neighbors if they share a node; however, the model can incorporate a more general neighborhood definition. To prevent neighborhoods from becoming too large, a problem that can lead to model degeneracy [115], we analyze nodes in a geographical setting, either observed or imposed. From this space, a "circle of influence" around the nodes restricts the number of potential edges.

Explicit identification of neighborhoods differentiates the LSGM from the other two methods. Due to the conditional independence of edge variables, there is no neighborhood structure for the latent variable models. In an ERGM, the choice of statistics in the negpotential function (5.5) implies a dependence structure and thus neighborhood specification. However, the induced neighborhood structure is rarely of interest and often meaningless. One advantage of the neighborhood definition is its ability to explain and control local dependencies. This leads to a more straightforward interpretation of parameters which, in turn, leads to a better understanding of a common issue with the ERGM: model degeneracy. Another benefit of conditional specification is that it is possible to compute a probability for all edges, not just observed edges.

Section 5.2 contains a more detailed argument for the LSGM with focus on the combination of conditional specification and neighborhood definition and the advantages of these features. We explore various ways this method can generate random graphs and examine the features of the resulting graphs. In its simplest form, the LSGM has two parameters. We demonstrate the effects of these parameters through simulation studies. Finally, we estimate the parameters from a realized network with a detailed interpretation of the fit.

5.1 Background: Random Graph Models

Most of the modern methods of network modeling and construction can be traced back to the often misinterpreted Erdös-Rényi graph. Two equivalent specifications of this model were proposed at approximately the same time in 1959. In a series of papers, Erdös and Rényi [65] specified a random graph model where the number of nodes, n, and the number of edges, m, are fixed and a uniform distribution is placed on all N possible graphs, where

$$N = \binom{\binom{n}{2}}{m}$$

In the same year, Gilbert [35] proposed an equivalent specification of the model. Gilbert's specification considers all graphs with a fixed number of n nodes and edge formation according to a constant, independent probability p assigned to each of the $\binom{n}{2}$ pairs of nodes. From this specification, the likelihood of a particular graph can be represented as a binomial distribution. In an unjust twist of history, this second specification proposed by Gilbert is is often referred to as the Erdös-Rényi graph. Some researchers do acknowledge Gilbert's contribution, referring to this model as the Erdös-Rényi-Gilbert model. It is also called the Bernoulli graph [47], Poisson model [21], or classical random graph model [65].

Kolaczyk [65] defines a (random) network model as the collection

$$\{\mathbb{P}_\theta(G), G \in \mathscr{G}; \theta \in \Theta\}$$

where $\mathscr{G}$ is the set of all possible graphs and $\mathbb{P}_\theta$ is a probability distribution over $\mathscr{G}$ with parameter vector θ. We discuss three common approaches to specifying a random-graph model. The first method is to restrict the set of graphs under consideration, $\mathscr{G}$, by identifying a desired set of features, such as the number of nodes and edges. As in the specification of Erdös and Rényi, $\mathbb{P}_\theta$ is then specified as a uniform distribution over the resulting set of possible graphs. The next approach to specifying the model is to induce $\mathbb{P}_\theta$ through an algorithmic generating mechanism which simulates a graph. Random variables are often assigned to certain components of the generative process and probability distributions specified for the random variables. A limitation of this method from a statistical viewpoint is the frequent lack of a likelihood function for the entire graph. Although the generative algorithm may induce $\mathbb{P}_\theta$, in most instances it is prohibitively difficult to formulate and is rarely, if ever, attempted. The last approach is to explicitly specify $\mathbb{P}_\theta$ by associating subgraph configurations and covariate information with graph topology of interest. This is the approach taken by most statisticians in the field of network analysis.

The three approaches to model specification are not mutually exclusive nor do they encompass all possibilities. For instance, the Erdös-Rényi-Gilbert model can be cast as a model which fits into all three categories. Some generative methods have only partial probability structures. Thus, it is not clear how to take a graph generated through some of the methods and perform a probability analysis.

The next two subsections introduce some of the contributions to the field of network science. The previous work is divided into that with a focus on "Algorithmic Construction" and that with

a focus "Probabilistic Modeling". Models under "Algorithmic Construction" mostly fall under the first two approaches of network model specification. For most of the models discussed in this section there is a lack of interest in a likelihood function. In contrast, models discussed in the "Probabilistic Modeling" section are defined by a likelihood and thus follow the third approach to network model specification. Neither section represents an exhaustive list of the proposed methods and models in the literature, as the field of network science is vast and quickly evolving.

Most of the discussion concerns simple graphs, or graphs with unweighted edges and no self-loops, although we extend this where necessary. Edges of the graph can be directed or undirected. We assume that the graph is observed at a single point in time, thus ignoring recent work on dynamic aspects of networks. Let V represent the set of vertices, or nodes and E the set of edges between pairs of nodes. Assume there are $|V| = n$ nodes and $|E| = m$ edges. In this section, we represent a graph G by edge values collected into $\mathbf{Y}$, the $n \times n$ adjacency matrix. Each entry Y_{ij} a binary random variable designating the presence, $Y_{ij} = 1$, or absence, $Y_{ij} = 0$, of an edge between nodes i and j. We represent a realization of the graph as $\mathbf{y}$.

Lastly, although the term graph is often referred to as the mathematical representation of a network, the two terms will be used interchangeably throughout this section. We attempt to keep the notation consistent, rather than retaining what individual authors use. See Table 5.2 for a list of notation for this section.

5.1.1 Graph Analysis: Algorithmic Construction

The defining feature of the graph analysis techniques that will be discussed in this section is the absence of a likelihood function of the entire graph, either from a lack of consideration or from an inability to discern the functional form. This work has been published largely in the computer science and statistical physics literature where the focus is on the ability to generate "realistic" graphs. Researchers in this area have condensed networks of interest into common, seemingly important features. The goal of the proposed graph-generation algorithm is to quickly generate a graph with as many of the important features as possible. These algorithms may require parameter specificagtion and there is often probability involved in the formation and deletion of edges. However, given an observed network, these models cannot usually estimate values for the parameters, quantify uncertainty, or account for measurement error.

The argument for algorithmic graph generation is to gain an understanding of the processes that lead to the formation of an observed graph [75]. Often the network to be analyzed is observed at a single point in time. Intuitively, if the algorithm results in a graph comparable to the observed network, it is plausible that the observed graph arose as a result of operations similar to those taken by the algorithm. Understanding the graph formation procedure can lead to an ability to detect abnormalities in another observed network, allow one to compress a large network into a smaller one with the same features [21], or to extrapolate and test out scenarios on graphs which cannot be observed [75], e.g., the Internet in five years. It can also enable more rigorous testing of algorithms designed for networks that "look like" the single example.

Three features are commonly observed in "real" networks [73]: a skewed or heavy-tailed degree distribution, a small diameter, and clustering. Generators aim to emulate these three features exactly as they appear in a network of interest, in addition to as many other features as possible. For example, a recently proposed model boasts the ability to simulated graphs which match real networks on 11 network characteristics [38]. We describe only the three generally agreed upon, most important features.

The degree of a node is the number of edges incident with the node [113]. In an undirected graph, the degree of node i, denoted $d(i)$, is found by summing over either the ith row or column ofthe adjacency matrix, $d(i) = Y_{+j} = \sum_{j=1}^{n} Y_{ij}$. The degree distribution is the collection of degrees for all nodes in the graph, $\{d(1), d(2), \ldots, d(n)\}$. Many networks of interest contain a few nodes with a large degree while a majority of the nodes have a small degree. There are generally nodes at all scales in between, with lower-degree vertices more frequent. In a social network such as Facebook, this implies there are a few popular people, e.g., celebrities, while most people have far fewer connections. In a graph of the Internet, there are a few websites to which many other sites link, e.g., Wikipedia, Google, while a vast majority have substantially fewer. This phenomena suggests the degree distribution is heavily right skewed. Historically, one common example or, is a power law with probability density function of the form

$$p(d(x)) = A \times d(x)^{-\gamma} \tag{5.1}$$

where $A > 0$ and $\gamma > 1$ is the power law exponent. Networks with this property are called scale-free graphs [10]. The value of γ is often used as a metric to determine how well the algorithm was able to replicate the degree distribution of the observed network [7], although estimation of the exponent is not straightforward nor is the method to compute it consistent. For example, Chakrabarti and Faloutsos [21] list seven of the more commonly used methods.

The second important feature is a measure of graph connectedness. Distance between nodes in graph theory is defined as the number of edges on the shortest path between them. If no such path exists, the distance is defined to be infinity. A graph is connected if all distances are finite and unconnected otherwise. The diameter of the graph is the maximum distance between all pairs of nodes [44]. If the graph is disconnected, often is suffices to consider the largest connected subgraph [113]. Alternatively, one can compute the effective diameter, which is the minimum number of edges between some, often large, percentage of the pairs of nodes [21]. The length of the diameter in empirical networks is quite small, especially compared to the size of the network, resulting in the "small-world" effect. For example, Watts and Strogatz [132] examined a graph with 225,226 movie actors as nodes with an edge between actors in the same film. The diameter of the largest connected component was 3.65In the same work, Watts and Strogatz found the US power grid, represented as 4,941, nodes had a diameter of 18.7.

Clustering is the final important feature for graph generators to replicate. This phenomena refers to the large number of triangles in an empirical network. Transitivity is a related concept. In a social network setting, transitivity implies that two individuals are more likely to be friends if they share a common friend over two individuals chosen from the population at random. Newman, Watts, and Strogatz [86] claim the probability is several orders of magnitude greater for nodes with a distance of two between them. The clustering coefficient C quantifies the amount of clustering.

This statistic is the proportion of the length-2 paths, or "wedges," that are closed and thus form a triangle as is shown in the formula below.

$$C = \frac{3 \times \text{Number of triangles}}{\text{Number of length-2 paths}} \tag{5.2}$$

Empirical networks have a larger value of a clustering coefficient than if edges were formed at random. The movie actor example of [132] has a clustering coefficient of 0.79, so 79% of possible triangles were closed. The authors contrast this with a single graph of the same number of nodes and edges generated by placing the edges at random which resulted in a clustering coefficient of 0.00027.

Researchers have identified som limitations of the graph generation algorithmic approach to network analysis. First, although the algorithms attempt to recreate as many features of the realized network of interest as possible, there is no consideration as to if these features are sufficient or necessary descriptions of network structure [28]. The important features are chosen because they appear frequently in observed graphs, although recent analysis suggests some features are not as ubiquitous as previously believed [38]. In fact, Bar, Gonen and Wool [7] suggest that the power law degree distribution demonstrated in the Autonomous Systems (AS) network, a crucial component of Internet connectivity, may be a consequence of the manner in which the data were collected. Descriptions used to summarize the algorithms do not explore the full parameter space [38] and are given as point estimates without a measure of uncertainty, and thus the summary quantities of the "real" networks can be highly misleading [28].

Statistical methods for estimating the model parameters of observed data are also lacking [28]. When statistical methods are used, they are sometimes applied incorrectly or without regard for assumptions. As an example, one way to estimate the power law exponent (γ in Equation 5.1) is to plot the degree distribution on a log-log scale and estimate the slope either through ordinary least squares or visual inspection. This approach has been used even in the presence of strong non-linearity [38]. There is no ability to account for or quantify measurement error or biased data. Again, Bar, Gonen, and Wool[7] suggest that up to 50% of the edges in the AS network are not observed; however, even with this acknowledgement, the authors claim they "cannot model data that are unknown".

Despite the statistical limitations of graph generating algorithms, a lot of attention has been given to the approaches in the statistical physics and computer science literature. In 2010, Kolaczyk [66] speculated that at least 2/3 of the published work on network analysis focused on descriptive methods. We discuss a few historically important and illustrative examples of graph generating algorithms. Many of the algorithms not included in the discussion are slight variations of those considered. Those discussed are a very small fraction of the algorithms available as [38] stated, "Alternative graph generation mechanisms appear every day".

5.1.2 Random Graph Models

Random Graph Models are those for which the set of possible graph $\mathscr{G}$ has been defined and equal probability is place on each graph, $G \in \mathscr{G}$. Models of this type are those specified according to the first approach discussed by Kolaczyk [65]. A random graph model is then completely determined by the set of plausible graphs of interest. This can be accomplished in one of two ways: by explicitly stating the set or by determining the possible graphs that could arise from a graph generation algorithm.

As was mentioned in Section 5.1, the Erdös-Rényi-Gilbert model can be cast as all three types of model specification. To partially justify this claim, we discuss this model as a random graph model. In this context, the Erdös-Rényi-Gilbert model is often referred to as the classical random graph model or just *the* random graph. With the specification of Erdös and Rényi, the set $\mathscr{G}$ is defined as all graphs with a n nodes and m edges for fixed numbers n and m. Gilbert's specification of this model can be considered a graph generation scheme. Goldenberg et. al. [38] refer to models that were originally proposed to describe a single, static network, but can be cast as a process to generate a graph as "pseudo-dynamic". For the Erdös-Rényi-Gilbert model, the process begins with n disconnected nodes. Each pair of nodes is considered in turn and an edge is added between them with probability $p = m/\binom{n}{2}$, independent of all previous edge decisions. The set of nodes remain fixed and once an edge is added, there is no mechanism to remove it.

One advantage of the Erdös-Rényi-Gilbert model is that many properties can be calculated exactly [86]. Much attention has been given to "phase changes" [28] or "phase transitions" [21]. One such transition occurs occurs at $\lambda = pn = 1$. For $\lambda < 1$, the graph will contain small, disconnected groupings of edges. The phase associated with $\lambda > 1$ is characterized by one giant component [28]. A giant component is defined as a strongly connected set of nodes containing a majority of the nodes of the graph [65] which scales up with the size of the network. Newman, Watts, and Strogatz [86] add the existence of a giant component to the list of features desired in a graph that represents reality. Because of this giant component, graphs simulated for $\lambda > 1$ do possess the small-world property. However, these graphs fail to reproduce the other two important features observed in empirical graphs from the "real world" [21]. The degree distribution of the Erdös-Rényi-Gilbert model is Binomial and approaches a Poisson as $n \to \infty$ and therefore, the graphs resulting from the generative method are not scale-free. Because the edges forming independently, there is no pressure for triangle formation and thus the graphs lack the desired clustering. Often, the Erdös-Rényi-Gilbert model is used as a "straw-man" model, as newly proposed algorithms demonstrate their worth by comparison to the Erdös-Rényi-Gilbert model.

In order to address some of the shortcomings of the classical random graph model, one proposed solution is to restrict $\mathscr{G}$ to only graphs with desired properties and make each of these graphs equally likely. Graph generation algorithms of this type are called generalized random graph models. Modifications to the set $\mathscr{G}$ are intended to produce graphs with feature such as small clusters of highly connected nodes, more triangle closures [28], or most commonly a specified degree distribution [2]. For this last type of restriction, the number of nodes and the degree distribution of the graph are fixed. Thus, the set of possible graphs in this generalized random graph model is a subset of those allowed under the Erdös-Rényi-Gilbert model, given Erdös and Rényi's specification. We

focus on restrictions to $\mathscr{G}$ which fix the degree distribution.

The method to simulate a graph with a specified number of nodes and a marginal degree distribution is as follows. Begin with a graph of n unconnected nodes. Each node is randomly assigned a degree or number of edges to which the node is an end point. Nodes are then joined until none of the nodes have any extra degrees. The standard algorithms developed to perform this last step are the matching algorithm and switching algorithm, discussed in [65]. Alternatively, the Chung-Lu model [2] begins with a list of node degrees, not necessarily drawn from any distribution. The final graph matches the desired distribution asymptotically, but individual nodes can have degrees other than the desired degree.

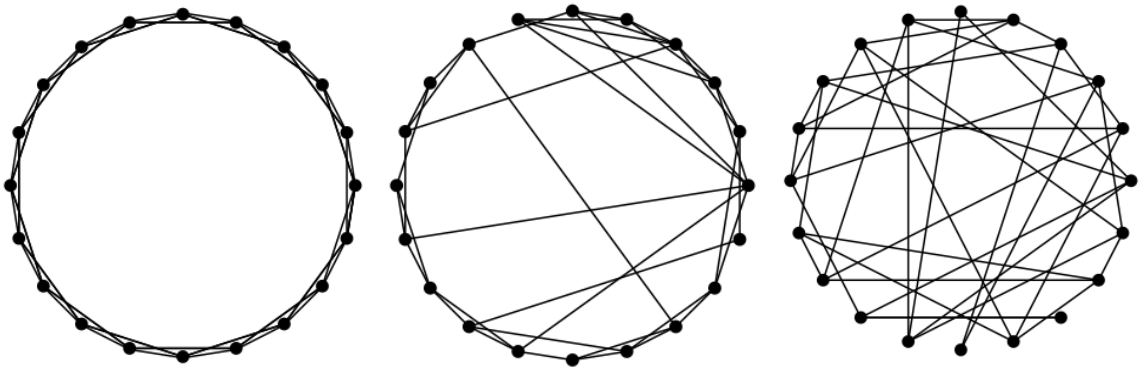
Like the classical random graph model, mathematical properties for the generalized random graph model where the degree distribution is fixed can be solved in the limit of large n. Specifically, if the degree distribution is fixed as a power law of form shown in Equation 5.1, the existence of a giant component, as well as its size, can be determined as a function of A and the power law exponent, γ. Under this approach, the diameter and average path length of the graphs in this set can also be determined [2]. More generally, if the specified degree distribution is not necessarily a power law, the emergence of the giant component can be computed based on the first two moments of the degree distribution and its size as a function of the number of nodes n.

The criticism of the generalized random graph model is that it often only matches the degree distribution, and if the giant component exists the small world property, but often fails to account for the high level of transitivity, the third important feature for graph algorithms to consider. In addition, Krivitsky et. al. [70] note that this model cannot distinguish between two graphs which have the same degree distribution but with structure that differs according to other metrics.

5.1.3 Small World Models

Network analysis began with the classical random graph model and a goal of understanding its properties. As the number and availability of observed networks increased, the limitations of the classical random graph model as a representation of reality became more clear. This realization prompted what Kolaczyk [65] refers to as a significant historical shift in the approach to network analysis. The move was away from a theoretical understanding of the random graph model and to the creation of models designed to explicitly generate a graph with features of interest. Clearly, the generalized random graph model could be considered of this type with its ability to recreate a specified degree distribution exactly. A seminal work that helped to spur this change to graph generation is Watts and Strogatz [132] which introduced the small-world model.

A small-world model is highly connected and transitive [38], resulting in a small diameter but large value of the clustering coefficient of Equation 5.2. The combination of these two features is not possible with the random graph model because as the number of nodes increases the diameter also increases, while the clustering coefficient is inversely related to the number of nodes [65]. Watts and Strogatz [132] envisioned a spectrum of randomness for networks. At one extreme is the regular graph which exhibits no randomness. In a regular graph, all nodes have the same degree.



(a) Regular, $p = 0$ **(b)** Small World, $p = 0.2$ **(c)** Random, $p = 1$

Figure 5.1. Demonstration of the Small-World model of [132].
The graphs increase in randomness from right to left.

A particular regular graph of interest to Watts and Strogatz is has n nodes equally spaced on the circumference of a circle and each node is connected to its closest k neighboring nodes. There will be an edge between node i and the closest $k/2$ nodes clockwise on the circle and $k/2$ in the counter-clockwise direction from node i. An example with $n = 20$ and $k = 4$ is shown in the far left plot of Figure 5.1. This kind of regular graphs is valuable as they result in a high value of the clustering coefficient, C. At the other end of the randomness spectrum is what Watts and Strogatz [132] refer to as a random graph. A random graph contains the same number of edges, $m = kn$, where each edge connects two nodes chosen independently and at random, excluding the possibility of self-loops and multi-edges. An example of a random graph with $n = 20$ and $m = 4 \times 20 = 80$ is shown on the far right of Figure 5.1. The advantage of a random graph is the small diameter, or small-world behavior. The Small-World model, also known as the Watts-Strogatz Model, is a compromise between the regular and random graphs on the randomness spectrum, which is able to retain both advantageous features, the high clustering of the Watts-and-Strogatz regular graph and the small-world property of the random graph.

The small-world model can also be cast as a generation mechanism and thus is a "pseudo-dynamic" model as defined by [38]. The process of generating a small-world model begins with a regular graph of n nodes each with k connections. Each edge is considered in turn and has a fixed, independent probability p of being rewired. If an edge is to be rewired, one node of the edge is relocated to a different node, chosen uniformly from the remaining $n-2$ nodes. Again, care is taken to avoid self-loops and multi-edges, thus the number of edges remains constant at $m = kn$. The middle plot of Figure 5.1 shows a small-world network with probability to rewire each edge, $p = 0.2$. The parameter p controls the amount of randomness. In the extremes, $p = 0$ implies that no edges are rewired, resulting in the regular graph. A $p = 1$ implies that all edges are rewired and thus a random graph is simulated.

The disadvantage of the small-world model is that it does not produce graphs with a power

law degree distribution. The regular graph will have a degenerate degree distribution as all nodes have degree k and the random graph has a Binomial degree distribution. The small-world model will have a degree distribution somewhere between the two extremes; however, the value of p is often chosen to be small and thus the degree distribution more often resembles the degenerate distribution of the regular graph. In addition, there do not exist formal statistical methods to assess the fit of this model to empirical networks nor to examine the evolution of the Small-World model [38]. The properties of the Small-World model are not as easily accessible as the random graph models of Section 5.1.2. Kolacyzk [65] describes this as an "open problem", although there have been some attempts.

5.1.4 Preferential Attachment

The preferential attachment models are network growth models [65] since they are designed to model the evolution of a network over time. The basis of the modern varieties of preferential attachment models was developed by Barabási and Albert specifically to model the expansion of the World Wide Web (WWW). The authors were motivated by the observation that new webpages tended to form links to the more popular, currently existing pages. The rationale behind the approach has been referred to as the rich get richer or cumulative advantage [21] and is related to Zipf's Law and the Chinese Restaurant Process [128].

In contrast to the network approaches previously discussed, the preferential attachment models allow for network growth through the addition of nodes. The process of generating a network begins with n_0 nodes and m_0 edges. At each iteration one new node is added to the network with $q < n_0$ edges to existing nodes. The probability a new node connects to an existing node v is proportional to v's degree. Thus, if $d(v)$ represents the degree of node v, the probability node v will connect to the newly added node is

$$p_v = \frac{d(v)}{\sum_i d(i)} \tag{5.3}$$

After t iterations the graph will contain $n_0 + t$ nodes and $m_0 + qt$ edges. An important property of the preferential attachment model is that it produces graphs which are scale free. In fact, as the number of iterations grows, the power law exponent γ approaches 3 [65] regardless of the number of nodes added at each iteration, q [21]. The resulting graph also exhibits a small-world behavior. Asymptotic bounds have been determined for the diameter of the preferential attachment graphs which related to the number of nodes logarithmically [76].

There are numerous limitations of the networks produced by the original inception of the preferential attachment method. First, the diameter growth as the number of nodes increases may not match realistic graphs as some recent evidence suggests the diameter may actually shrink as the network grows [75]. The small-world behavior is present but the model does not include a knob to control it [128]. Another feature that is not controlled within the model is the power law exponent which always approaches $\gamma = 3$. Although the shape of the degree distribution, particularly the tails, may change as new nodes are added, the average degree remains constant at q. Empirical evidence suggests that as a network grows the average degree should increase rather than remain

constant [21]. Next, the model is unable to produce networks with a dense core and leaves because new edges are added in sets of size q [7]. It does not produce graphs with several connected components or isolated nodes [21]. The model was developed for simulating undirected graphs. Modifications have been attempted to produce a directed network, but were unable to recreate reciprocity [17].

When a deficiency of the original preferential attachment model is presented, it is often followed with a modified version that addresses the stated inadequacy. The simplicity of the approach has also led to its many extensions. Chakrabarti and Faloutsos [21] detail ten of these extension and which inadequacy of the original preferential attachment model it addresses. For example, the initial attractiveness model allows for a more general power law by adding a parameter A to the edge connectivity probability in Equation 5.3 so that it becomes

$$p_v = \frac{d(v)+A}{\sum_i [d(i)+A]}$$

with the resulting power law exponent $\gamma = 2 + \frac{A}{q}$. A more elaborate variation is the forest fire model presented in [76]. This model results in networks that are scale free, have a decreasing diameter, an increasing average degree, are directed, and allows for community structure. Two additional parameters are used in this model: a forward burning probability, p_{fb}, and a backward burning ratio, r_{bb}. At each iteration, a new node is added to the graph and edges are added according to the following steps:

1. Choose an "ambassador node", w uniformly at random from the existing nodes of the graph. Form a link from the new node to w.
2. Draw a random number, n_1 from the binomial distribution with mean $(1-p_{fb})^{-1}$.
3. Choose n_1 of the currently existing edges of node w. Select edges that are directed to node w with probability r_{bb} times less than edges that are directed away from node w. Let $w_1, w_2, \ldots, w_{n_1}$ represent the nodes at the other ends of the edges selected.
4. Connect the newly added node with a directed edge to $w_1, w_2, \ldots, w_{n_1}$.
5. Repeat steps 2 and 3 recursively for each of the $w_1, w_2, \ldots, w_n$

Extensions to the forest fire method allow isolated nodes, or orphans, to choose multiple ambassador nodes.

The preferential attachment models are an example of how many of the algorithmic graph generators are applied to create networks and the forest fire model an example of how complicated the algorithms can become. Simulated networks resulting from these models are mostly used as a basis of comparison to certain characteristics of realized networks. The goal is for the simulated networks to match the realized graphs of interest on the characteristics studied. Metrics and statistics have also been developed to test the resemblance of the simulated graph to the network of interest. Often these values relate back to the three main features of a network. Little effort has

been made to statistically fit the models to network data or attempt to estimate the parameters of the model for a given observation. However, code in the newly-released FEASTPACK [67] does contain methods for estimating parameters for the BTER model [117].

5.1.5 Graph Analysis: Probabilistic Modeling

In contrast to the algorithmic graph generators of the previous section, the models described in this section can be described with a likelihood function. Therefore, it is possible to formulate a statistical model for an entire graph with a joint distribution and to conduct statistical inference. Models described in this section are able to estimate controlling parameters and determine probability for a realized graph, and assess the fit of the model. Although it is possible to generate graphs from this model, these models go beyond generation.

We discuss in detail two broad classes of probabilistic modeling of random graphs that are common in the literature. First, is the Exponential Random Graph Model (ERGM). This model specifies a joint distribution for the collection of variables that represent the edges with a goal of describing global network features through interactions of local edge configurations. Conversely, the models categorized as Latent Variable Models focus on the properties of the individual nodes [60] of the graph. This category encompasses a wider range of hierarchical graph models. The common thread between these models are that the edges variables are specified as conditional distributions considered to be independent given some latent variable, such as block membership or position within a social space.

5.1.6 Exponential Random Graph Models

Exponential Random Graph Models (ERGM) are possibly the most widely used and extensively-studied model that falls under the category of probabilistic modeling. The models arose out of collaborations between sociology, psychology, and statistics and were originally developed to model social networks. Unlike many of the other analysis techniques discussed, the ERGM has been applied to more than just that area in which it was originally intended, such as biology/medicine [43, 118]. Its popularity can be attributed to the ERGM's ability to represent graph topology while allowing for complex dependencies.

Introduction An ERGM is specified as a joint distribution for $\mathbf{Y}$ in the exponential family form [65]. The use of the exponential family is attractive because the sufficient statistics are explicitly tied to parameters and are equal to their expected values [55]. The functional form of the joint distribution can be represented as

$$\Pr(\mathbf{Y}=\mathbf{y}) = \frac{1}{\kappa}\exp\left\{\sum_{T\subseteq C}\theta_T g_T(\mathbf{y})\right\} \tag{5.4}$$

where

- C is the set of all paris of nodes that could possible be joined by an edge; almost exclusively this is all pairs of nodes, so $|C| = \binom{n}{2}$
- $T \subseteq C$ is a subset of the possible edges and is often referred to as a configuration
- θ_T is the parameter corresponding to configuration T
- $g_T(\mathbf{y}) = \prod_{(i,j)\in T} y_{ij}$ is a network statistic, which is equal to 1 only when the configuration T occurs in $\mathbf{y}$
- The summation in the exponent is referred to as the negpotential function

$$Q(\mathbf{Y}) = \sum_{T\subseteq C} \theta_T g_T(\mathbf{y}) \tag{5.5}$$

- $\kappa = \sum_{\mathbf{Y}} \exp\left\{ \sum_{T\subseteq C} \theta_T g_T(\mathbf{y}) \right\}$ is the normalizing constant
- Define $\xi \equiv \{\mathbf{y} : \Pr(\mathbf{Y} = \mathbf{y}) > 0\}$ to be the support

The negpotential function is given much attention because defining the negpotential defines the joint distribution, up to a constant. Specifying a particular set of parameters, θ_T, or equivalently network statistics, $g_T(\mathbf{y})$, to be included in the negpotential specifies the ERGM. Often these statistics represent counts of subgraph features, such as the number of edges, number of triangles, or number of edges in a particular block, and the corresponding coefficients would represent density, transitivity, and block effect, respectively. A majority of model development for the ERGM has focused on the parameters or statistics to be included in the negpotential function. There have been four main phases in the development of the ERGM. First, the proposal of a model of form Equation 5.4 for network analysis. Next, the relaxing of an independence assumption to allow for more complex dependence structures, followed by the proposal of parameters not motivated by dependence. The most recent wave of development involves the introduction of new parameters designed specifically to address model degeneracy, a common issue with the application of an ERGM to realized networks.

Many studies point to the seminal work of Holland and Leinhardt [55] as the first introduction of the ERGM. This work proposes a log-linear model, termed the p_1 model, in the form of Equation 5.4 to model the dyads of a directed social network. A dyad is defined as a pair of nodes and the possible ties between them, which in an unweighted, directed graph could be 0, 1, or 2. Network analysis at the time was only descriptive, focusing on aspects such as the degree distribution or the distribution of attributes on the nodes. In contrast, Holland and Leinhardt [55] were able to stochastically model the patterns of relationships. An advantage of the p_1 model is a simultaneous estimation of a parameter to represented reciprocity, or a tendency for a edge to be reciprocated, and a parameter for differential attractiveness which occurs when a few nodes attract a comparatively large number of edges. The authors noticed both reciprocity and differential attractiveness appeared in social networks at a rate greater than if edges were allowed to form at random.

One disadvantage of the p_1 model is that it assumes independence between the dyads. This assumption results in a likelihood which is a product of probabilities for each dyad, which was necessary to estimate the parameters through standard statistical techniques. Relaxing this assumption was identified by the authors to be difficult at the time. The next development of the ERGM was introduced by Frank and Strauss [31] who incorporated a more general definition of dependence. They adopted Besag's [13] methods from spatial statistics and applied them to social networks. It is because of this consideration of complex dependence that Frank and Strauss [31] is also referenced as the origin of the ERGM. Although the paper proposes a general dependence structure, Frank and Strauss give the most attention to the Markov graph, which we discuss in detail below.

One important contribution of Frank and Strauss [31] was defining a dependence graph. The nodes of a dependence graph represent the potential edges from the original graph. An edge between nodes in the dependence graph indicates the two corresponding random variables are conditionally dependent. The Markov graph has an incidence definition of dependence, thus, two edges are conditionally dependent if they share a common node. In the dependence graph this implies that there are no edges between disjoint sets of node indices. Stated another way, let $\{i, j\}$ and $\{m, n\}$ represent two potential edges in the original graph and thus two nodes in the dependence graph. If $\{i, j\} \cap \{m, n\} = \emptyset$ then there will not be an edge between these nodes in the dependence graph. The set of random variables on which a particular random variable is conditionally dependent, and thus is linked in the dependence graph, will be called its neighborhood. For example, in a model with Markovian dependence, the neighborhood of for a given edge, $Y_{i,j}$ can be stated as $\{Y_{r,s} : (i, j) \cap (r, s) \neq \emptyset\}$. We use this definition of neighborhood throughout this section, even though some literature uses other graph features for the neighborhood.

To apply Besag's [13] the methods to social networks, we use the Hammersly-Clifford Theorem. This important theorem has been stated and proven in various forms and in multiple references. (For the original see [22], for one similar to what we use see [23]) First, define a clique to be a single random variable or a set of random variables such that every pair within the set is pairwise mutually conditionally dependent, given the rest of the graph. Besag [13] showed that, under mild conditions, the negpotential function can be expanded uniquely over ξ as a summation over configurations of random variables, i.e., that the negpotential takes the form shown in Equation 5.5. The Hammersly-Clifford Theorem states that the parameter, θ_T, will be non-zero only if the random variables in the corresponding configuration T form a clique. Thus, the incidence dependence of [31] imply that the non-zero parameters will correspond to triangles and k-stars in the dependence graph. A k-star configuration results from k edges which all share a common node. Although the order of the stars can be $k = 1, \ldots, n-1$, we consider only $k = 1, 2$, making a more manageable number of parameters. The triad model consists of a term for 1-stars, i.e., number of edges or density, 2-stars, and triangles. In addition, the model includes an assumption of homogeneity so that all isomorphic graphs are equivalent.

The parameters, θ_T, included in the negpotential function of the Markov graph are the result of the Markovian dependence structure. The next refinement in ERGM specification began with Wasserman and Pattison [131] who proposed a slew of possible parameters or statistics motivated by graph topology. Models of this type were named p^* in honor of the p_1 model [55]. Wasser-

man and Pattison [131] contains four tables of possible parameters with a statement that those listed are just a subset of the possibilities. They recommend five methods for choosing the parameters for the negpotential. One of the suggested methods is to consider the resulting conditional dependence between possible edges. However, even in the other methods when the dependence between edges may not be explicitly modeled, the choice of statistics induces a conditional edge dependence structure. The authors describe some of the induced conditional dependencies, but for others admit that identifying the sets of conditionally dependent variables is not immediate and that some models induce "arbitrary complexity". The authors use trial and error to determine the best fitting model for an example dataset, although, according to Besag [16] do not "reach any particular conclusion". Varying the parameters in the negpotential leads to changes in the underlying dependence structure. Thus, although the authors acknowledge that the conditional dependence structure is determined by the choice of parameters, they often fail to identify the induced structure or recognize the effect of changing this structure. More recently, Goodreau [39] pared down the list of parameter-choosing methods by describing two approaches: considering the nature of the dependence or the combination of parameters that fit the empirical network best.

The extension to include arbitrary statistics in the negpotential function, $Q(\mathbf{Y})$, proposed in [131] provides a straightforward method for considering exogenous attribute information in an ERGM. The parameters and statistics discussed up to this point have been endogenous, or functions of the graph itself, e.g., number of edges or transitivity. Exogenous attributes do not depend on the structure of the graph and can be incorporated at the level of the individual nodes, edges, or as symmetric functions of nodal covariates. An example of how to include an exogenous attribute of a node is through a main effect term which allows for a different value in the negpotential dependent upon the covariate value of the node [39]. A similar example for pairs of nodes is assortative mixing. This parameter attempts to capture the increased probability of edges to form between nodes within a particular attribute class [40]. When this effect is uniform across attribute classes, it is referred to as uniform homophily. For example, consider a friendship network between grade-school aged children. A uniform homophily term could represent the higher probability of friendships to form between children within the same grade. Differential homophily allows for a different parameter to describe this effect within the different attribute classes. For the grade-school example, a differential homophily term could be used to describe how friendships are more likely to form within gender *and* how females tend to make more friends than males [59]. As a final example, if the covariate is continuous, such as age, the absolute value of the difference can be included as a statistic, and the corresponding parameter would allow the probability to vary monotonically with the value of the absolute difference [39]. Attribute-based parameters and statistics do not affect the dependence structure. Thus, these parameters lead to a dyad-independence model if no other terms are included to account for a more complex dependence.

An approach that considers the affect of the dependence structure of higher-order statistics are the two extensions to the ERGM termed "neighborhood-based" models presented in [95]. The first model uses what the authors refer to as social settings, or groupings of nodes based on some condition, typically an attribute value. A particular model may have multiple social settings, and these social setting may overlap. Two edges are considered conditionally independent if they do not occupy a common setting, although it is not implied that they are conditionally dependent if they do. The purpose of this approach is to limit the number of conditionally dependent random

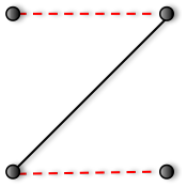


Figure 5.2. Configuration which leads to partial conditional dependence.

variables, or the size of the neighborhoods. The second model includes configurations of at least four nodes such that every pair of edges are part of a path of length three. An example of such a configuration is shown in Figure 5.2. The dependence structure induced by this form of statistics is termed partial conditional dependence. Two random variables are partially conditionally dependent if their dependence status is determined by the state of a third random variable. For the example shown in Figure 5.2, the two dashed, red lines will lie on a path of length three and therefore be conditionally dependent only if the solid, black edge is realized.

There are other extensions of the ERGM including to networks with multiple relations [96], networks with values on the edges [105], or including the attributes into the dependence graph to predict node-level attributes given the graph topology [103]. We do not discuss these extensions in detail. The final, significant contribution to the study of the ERGM we discuss is alternating or geometrically weighted statistics of [121] and [104]. We postpone full consideration of this new specifications until Section 5.1.8 on degeneracy because these terms address this specific issue.

Although the ERGM is expressed as a joint distribution, it can also be expressed as a conditional log-odds. This is often how the model is presented as it allows for a more natural parameter interpretation. Consider the random variable representing a potential edge, Y_{ij}. The model in Equation 5.4 implies the following full conditional distribution

$$\text{logit}[\Pr(Y_{ij} = 1 | \mathbf{Y}^c_{ij} = \mathbf{y}^c_{ij})] = \sum_{T \subseteq C} \theta_T \delta_g(\mathbf{y})_{ij} \tag{5.6}$$

where $\mathbf{Y}^c_{ij}$ represents all edges in the network other than edge ij and $\delta_g(\mathbf{y})_{ij}$ is the vector of change statistics. The change statistics is computed as

$$\delta_g(\mathbf{y})_{ij} = g_T(\mathbf{y}^+_{ij}) - g_T(\mathbf{y}^-_{ij})$$

where $g_T(\mathbf{y}^+_{ij})$ is the value of the network statistic if edge ij is forced to be present while the rest of the graph remains unchanged and $g_T(\mathbf{y}^-_{ij})$ is the analogous value when edge ij is constrained to be absent. The change statistic is interpreted as the effect on the network statistics if the particular random variable is changed from 0 to 1, while all other random variables remain constant. As an illustrative example, if $g_T(\mathbf{y})$ represents the density of the graph, the change statistic is 1 for all potential edges, because the addition of an edge will always increase the density by 1.

A parameter, θ_T, of Equation 5.6 is interpreted as the increase in conditional log-odds of a network as a result of a unit increase in the corresponding statistic. The conditional probability for Y_{ij} depends on the rest of the graph, $\mathbf{Y}^c_{ij}$, only through the change statistic. The interpretation is from the point of view of the edges. If the model includes a homogeneity assumption, as is usually the case, the parameter values indicate the type of edges which are most probable, e.g., those that close triad, those that connect two nodes within the same attribute class, etc. Individual parameters interpretation is heavily dependent upon the other terms included in the model. For example, if the model includes two k-star statistics for $k_1 < k_2$, then the interpretation of the parameter for k_2 is the effect on the conditional log-odds of the network due to k_2-stars adjusted for the number of k_1 stars [65]. Although this is partially due to the fact that there are $\binom{k_2}{k_1}$ k_1-stars within a k_2-star, a confounding of interpretation occurs even if the statistics are not nested.

5.1.7 Estimation & Goodness-of-Fit

Exact maximum likelihood estimation (MLE) has never been a viable option for the ERGM. Computing the MLE would require repeated evaluation of a normalizing constant that involves a sum over all possible graphs. As the number of possible graphs grow super-exponentially with the number of nodes, $|\mathscr{G}| = 2^{\binom{n}{2}}$ this is computationally intractable even for trivially small networks [102]. For example, the number of simple graphs that can be formed from only 10 nodes is over 35 trillion. Thus, since the inception of ERGM, approximation methods have been devised to estimate the parameters. Unfortunately, model proposals have outpaced the estimation methods. The discussion below will focus on the more commonly used techniques for estimating the parameters of an ERGM: maximum pseudo-likelihood, two approximate maximum likelihood estimation techniques based on Monte Carlo simulations, and some recently proposed methods in Bayesian estimation.

Basag developed maximum pseudo-likelihood estimation (MPLE) for lattice [13] and non-lattice [14] data for applications of spatial statistics. Frank and Strauss [31] and Strauss and Ikeda [125] introduced MPLE to social networks and the ERGM. The pseudo-likelihood (PL) function is an approximation to the joint distribution and is computed as the product of the full conditional distributions. For the ERGM, this is the product of the conditional distributions shown in Equation 5.6. The problematic normalizing constant cancels out in the conditional distributions, and thus the PL function is always available in closed form.

A reason for the widespread use of MPLE was the ease and speed at which the PL can be maximized. Strauss and Ikeda [125] proved that the MPLE is equivalent to the maximum likelihood of a logistic regression where the data plays the role of both the dependent and independent variables. Using standard statistical software the PL can therefore be maximized through an iteratively reweighed Gauss-Newton least squares procedure. However, Geyer and Thompson [33] point out that the PL function is not the true likelihood for any model. The two functions are simply computationally identical. Standard errors reported from the logistic regression fit are not applicable to the PL maximizers because logistic regression presumes the independent variables are fixed and dependent variables are random. For the PL scenario both are random since they both are copies

of the data.

Although the PL approach is fast and provides an intuitive solution to the intractable normalizing constant, its use has been heavily criticized in the network analysis literature. The most common complaints are that the PL overestimates dependence and structural effects [81], underestimates standard errors, and performs poorly in practice. These issues are exacerbated when the dependence between random variables is strong. If random variables are independent, the PL is equivalent to maximum likelihood (ML). In a case study using DNA fingerprinting, Geyer and Thompson [33] compared their MLE approach to MPLE and found that MPLE estimated much higher values of a dependence parameter, produced unreasonable probabilities, and overall provided a very bad fit to the data. Robins et. al. [104] compared standard errors between ML and PL and found that on average PL standard errors were smaller, but could differ from ML by three to four times in either direction. Other identified issues with the PL method are that it is not admissible for a squared error loss function because it is not a function of complete sufficient statistics [119], it can produce infinite values even if the function converges [47], and, becauase it is approximate, it can fail to indicate when the model has become degenerate [104]. Lastly, the properties of the MPLE, specifically within the context of the ERGM, are not well understood and there is no asymptotic theory to base confidence intervals and hypothesis tests [65].

Besag [16] suggests that the MPLE is not likely to perform well for the ERGM unless the dependence between edge variables is weak. Camino and Fried [19] and Hoff et. al. [53] argue that ERGMs are more global than local, while Handcock [47] argues that the PL considers only local information. The PL function does not take into consideration the parameter space or normalizing constant. Thus, if this space is constrained or if the normalizing constant is an important aspect of the model, then PL is likely not to be an adequate estimation technique. Basag [15] describes MPLE as "a simple tool from another era" that will eventually become obsolete and recommends simulation because of recent increases in computing capabilities [16].

As an alternative to PL, two Monte Carlo approaches can approximate the parameters of an ERGM. The first, based on the work of Geyer and Thompson [33], is Markov Chain Monte Carlo-Maximum Likelihood Estimator (MCMC-MLE). It is a stochastic approximation of the log-likelihood and a maximization of the approximation. Hunter and Handcock [58] were the first to apply it to the ERGM. Snidjers [119] proposed the second method. He approximates the MLE through a stochastic approximation to the moment equation. It is based on the Robbins-Monro algorithm, a stochastic version of the Newton-Raphson algorithm. Both methods assume the ability to sample graphs from a specified ERGM. This is most directly accomplished through a Gibbs sampler using the full conditional of the log-odds shown in Equation 5.6 to update random variables in turn. Other methods update groups of variables at a time [119] or use a pure Metropolis algorithm [58].

To calculate the MCMC-MLE, first consider the log-likelihood

$$\ell(\boldsymbol{\theta}) = \left\{ \sum_{T \subseteq C} \theta_T g_T(\mathbf{y}) - \log(\kappa(\boldsymbol{\theta})) \right\}$$

where the normalizing constant has been written to emphasize its dependence on the unknown

parameter vector, θ. The key idea behind this approach is that the parameter values that maximize $\ell(\theta)$ are equivalent to those that maximize the log of the likelihood ratio

$$r(\theta,\theta^0) = \sum_{T\subseteq C}\left[(\theta_T - \theta_T^0)g_T(\mathbf{y})\right] - \log[\kappa(\theta) - \kappa(\theta^0)] \tag{5.7}$$

for some fixed and constant θ^0. The difference of normalizing constants can be approximated by generating a sample of M graphs from the ERGM with parameter values θ^0 and noticing that

$$\exp[\kappa(\theta) - \kappa(\theta^0)] = E_{\theta^0}\left\{\exp\sum_{T\subseteq C}(\theta_T - \theta_T^0)g_T(\mathbf{y})\right\}$$

Thus, an approximation of the likelihood ratio in Equation 5.7 is

$$\hat{r}_M(\theta,\theta^0) = \sum_{T\subseteq C}\left[(\theta_T - \theta_T^0)g_T(\mathbf{y}_{\text{obs}})\right] - \log\left\{\frac{1}{M}\sum_{i=1}^{M}\exp\left[\sum_{T\subseteq C}(\theta_T - \theta_T^0)g_T(\mathbf{y}_i)\right]\right\}$$

and maximization of this equation approximates the MLE. [58] also proposed a method for approximating the standard errors of this estimate, performing a likelihood ratio test and extended this method to the curved exponential family which is sometimes necessary for the degeneracy-combating parameters discussed in the following section. Another extension of the MCMC-MLE approach of [33] was proposed by [49] to estimate parameters while accounting for measurement error resulting from a partial observation of the network.

The other main approach presented in [119] is based on the Robbins-Monro algorithm which solves equations of the form $E_\theta[\mathbf{Z}] = 0$ for a random vector $\mathbf{Z}$. To estimate the parameters of the ERGM, the random vector takes the form $\mathbf{Z} = g(\mathbf{y}) - g(\mathbf{y}_{\text{obs}})$. The iteration step is

$$\hat{\theta}^{(n+1)} = \hat{\theta}^{(n)} - a_n D_n^{-1} Z(n)$$

where $\Pr(Z(n)|Z(1),\ldots,Z(n-1)) \equiv \Pr(Z|\theta = \hat{\theta}^{(n)})$. The a_n is known as the gain sequence and is a sequence of positive values that converge to 0. If $a_n = 1/n$ and $\mathbf{Z}$ has an exponential family, the optimal choice of D_n is the derivative matrix [65]. For the ERGM this is given by

$$D_{j,k} = \frac{\partial^2\kappa(\theta)}{\partial\theta_j\partial\theta_k}$$

The three phase algorithm presented in [119] is based on this approach and includes a check for validity and a method to compute the covariance matrix of the estimator.

Bayesian methods of estimation have not been extensively considered for the ERGM despite the fact that they are appropriate for quantifying uncertainty of the estimated model parameters and for formally comparing competing models [19]. [134] used an empirical Bayes approach to the p_1 model of [55] which estimates the model parameters through a Newton-Raphson step and then uses an EM algorithm step to estimate the covariance parameters. The full algorithm cycles through both steps and hence was termed the EM-Newton algorithm. [36] extended this to a fully

Bayesian approach and also considered a model which includes a block effect parameter, but did not consider a model with a dependence structure beyond dyad-independence.

Computational complexity is the main reason that Bayesian estimation techniques have lagged behind for the ERGM. Usual MCMC algorithms are able to sample from posterior distributions as long as they are known up to a constant. The ERGM is doubly intractable because it is not possible to evaluate the normalizing constant of the posterior or the likelihood. This was overcome by [19] who adapted the exchange algorithm to network analysis. The algorithm involves augmenting the data with $\mathbf{y}'$ and parameters with θ' and sampling from the augmented posterior density

$$\pi(\theta', \mathbf{y}', \theta|\mathbf{y}) \propto \pi(\mathbf{y}|\theta)\pi(\theta)h(\theta'|\theta)\pi(\mathbf{y}'|\theta')$$

where the posterior of interest, $\pi(\theta|\mathbf{y})$ is a marginal distribution. The distribution $\pi(\mathbf{y}'|\theta')$ is the same exponential family form as $\pi(\mathbf{y}|\theta)$ and the auxiliary density, $h(\theta'|\theta)$, can be any arbitrary distribution with dependence on θ. The algorithm first samples the augmented data, $\mathbf{y}'$, and parameters, θ', through a Gibbs sampler and then proposes a swap between θ and θ'. The significance of this method is that in the Metropolis acceptance probability, all intractable normalising constants cancel. A further difficulty arises when trying to sample $\mathbf{y}'$ as it requires exact sampling [60]. [19] propose sampling the augmented data using a"tie no tie" sampler which is computationally faster than a single dyad updating approach. In order to improve mixing and speed of convergence, [19] also recommend using population MCMC which combines multiple, simultaneous chains. Another similar approach was presented in [68] which samples from an extended sample space using the linked importance sampler auxiliary (LISA) Metropolis-Hastings algorithm. Like [49], this work also focuses on networks that are only partially observed and thus the LISA algorithm is part of a larger sampling scheme that also samples from the distribution of missing values.

One area which has not been fully developed for the ERGM, or network analysis in general, is assessing the fit of the model to an observed network. [113] indicates four groupings of existing model assessment techniques: ground truth comparison, link prediction, model comparison, and graphical goodness-of-fit. The first two approaches are possible only when the network is assumed to be know and are often used to compare two generative methods. Model comparison includes likelihood based measures, such as AIC [39], or Bayesian model selection such as selecting the model that minimizes the expected predictive deviance [36]. [41] warn against using the model comparison measures because they fail to indicate the manner in which the model is misspecified. Rather, the recommendation is for the graphical goodness-of-fit method presented in [57]. In this approach, a large number of graphs are simulated from the model with the estimated parameters. From these simulations a distribution of structural aspects are computed. Examples of structural aspects to consider are the number of realized edges, the number of triangles, or the average path length between pairs of nodes. The structural aspects can be the sufficient statistics included in the model or not. If the value of these structural aspects from the observed network appears to be a common value of the corresponding distribution obtained from the simulations, the the model is determined to be a good fit. The model is determined to fit poorly if the structural aspect was not a statistic included in the model. An observed value of a statistic included in the negpotential that is unlikely based on the distribution of simulated structural aspects is an indication of model degeneracy, an issue which we now discuss.

5.1.8 Degeneracy

A complication that has been recognized and well documented with the application of the ERGM to empirical networks is that of model or inferential degeneracy. The term degeneracy has been used to refer to a variety of undesirable model behaviors. Most of these behaviors are a result of the model placing a large amount of probability on a very small subset of the theoretically possible networks. Often these highly plausible networks are radically dissimilar and frequently include the complete (all edges realized) and empty (no edges realized) graphs. The phenomena has led to the definition of a "useful model" in [47] and [48] as one for which the probabilistic structure places a large amount of probability on graphs that could reasonably be produced by the underlying process. [48] lists three interrelated capabilities a useful model should possess: simulation, parameter estimation, and model assessment. These three features are lacking or extremely difficult in degenerate models. Because the ERGM cannot be degenerate in the strict sense [114], this behavior has also been referred to as near-degeneracy [104]. This issue is not unique to the ERGM as a similar behavior has also been recognized in a more general class of models for interactive systems [124] and is similar to long-range dependence observed in the Ising model [119].

The inability of degenerate models to simulate reasonable graphs can result in a failure to estimate model parameters. As described in the previous section, parameter estimation requires the use of a sampling scheme due to the intractable normalizing constant. If a large amount of probability is placed on a few, disparate graphs, the algorithms mixes very slowly, hardly moving for millions of steps [114]. One proposed solution to the slow mixing is the inclusion of a graph complement step in the Markov chain. At each step and with a small probability, the value of all random variables is reversed, e.g., for binary valued graphs $\mathbf{Y}^{(t+1)} = 1 - \mathbf{Y}^{(t)}$ [119]. Although this allows the algorithm to explore extreme modes of the distribution, this does not diminish the fact that the model is degenerate. The issue of degeneracy is not a failure of the algorithm or statistical inference technique, but rather the underlying or stationary distribution, [114], [121] or as a result of model mis-specification [40]. The MCMC-MLE algorithm fails to find an estimate when the subset of plausible graphs do not resemble the network of interest. The estimation technique requires that the model produce the observed values of the sufficient statistics [121] and even with the graph complement step, the algorithm may continually jump over the observed values.

If an estimation algorithm does converge and estimates are obtained, simulated graphs from the fitted model could still fail to reproduce much of the graph structure observed in the realized network. One instance of this phenomena was explained via change statistics by [121]. When the ERGM is simulated with a Gibbs sampler, the conditional distributions are stated as logistic regression with the change statistics playing the role of the independent variables, as shown in Equation 5.6. The edge values are updated one at a time, either by cycling through all edges in the graph or through random selection. Consider a model with a moderately-sized, positive value of the triangle or any of the $k \geq 2$ star statistics. Changing one random variable to 1 could induce a large increase in the change statistics of other random variables, causing these edges to be realized with high probability, which in turn creates an increase in the change statistics for other edges, and so forth. This effect was termed an "avalanche of change" by [121] and would quickly force the algorithm to the complete graph with little probability of moving away. The graphical goodness-of-fit method of [57], described in the previous section, was motivated by this commonly observed

lack of fit. In addition, [41] and [104] have suggested as a method to detect model degeneracy to check if the observed model could possibly have been produced by the fitted model.

Most of the work on diagnosing the cause of model degeneracy has identified a parameter space issue. We discuss in detail below three approaches to identifying the offending parameter values. Other potential causes include a large sample problem as discussed in [124] and [125] who prove that for a fixed set of parameters, the expected density approaches 1 as the number of nodes increases to infinity. Likewise, [115] warn against applying an ERGM to "large" graphs without explicitly defining "large", citing neighborhoods which grow as a function of the number of nodes as the culprit for degenerate models. Another suggested reason for the degeneracy is the dominating global nature of the ERGM over local structure which is especially especially relevant when attempting to find the MPLE [47]. [68] proved that a model which contain nested, degenerate models will also be degenerate. This implies that the issue of degeneracy can not be remedied by adding additional parameters into the model.

The first attempt to identify the subset of the parameter space which leads to degenerate models to be discussed relies the theory of discrete linear exponential families and the geometry of the parameter space. [47], [48] extend results from [9, see Cor 9.6] and [102] makes use of Shannon's entropy to identify the problem areas. To explain the method in [47] and [48], let $g_{T1}(\mathbf{y}), \ldots, g_{T2}(\mathbf{y})$ represent the statistics chosen to be included in the negpotential function, $Q(\mathbf{Y})$ of Equation 5.5, and let C represent the convex hull of the combination of possible values of the statistics computed from all possible graphs, $\{g_{\mathbf{T}}(\mathbf{y}) : \mathbf{y} \in \mathscr{G}\}$. [47] proved that the MLE does not exist if the observed values of the statistics, $g_{\mathbf{T}}(\mathbf{y}_{\text{observed}})$, fall on the relative boundary of C. This situation, he argues, occurs quite frequently in practice. Similarly, define C_1 to be the convex hull of the space formed by the statistic values computed from M simulated graphs, $\{g_{\mathbf{T}}(\mathbf{y}_1^*), \ldots, g_{\mathbf{T}}(\mathbf{y}_M^*)\}$. If $g_{\mathbf{T}}(\mathbf{y}_{\text{observed}})$ falls on the relative boundary of C_1, the MCMC-MLE does not exist. Therefore, if $C_1 \subset C$ there are observed statistics values for which the MLE exists, but the MCMC-MLE does not. This led to the proposed solutions of a Bayesian analysis where the prior distribution restricted all its mass to the non-offending areas of the parameter space.

Both [47] and [102] identified the problem region with a case study: a model with the density and 2-star parameter to a network with $n = 7$ nodes and a model with the density and triangle parameters to a network with $n = 9$ nodes, respectively. [47] identify the degenerate region through the convex hull method discussed above while [102] uses Shannon's entropy to quantify the amount of degeneracy where lower entropy corresponds to more degenerate parameter values. Both works concluded that the degenerate regions have a nicer, more identifiable form in the mean value parameterization rather than the natural parameterization of the exponential family. If we refer to the region of the parameter space for which degeneracy is not an issue as the effective parameter space [47], in both of the case studies this area was found to be much smaller than the theoretical parameter space. Although the case studies represent only two possible ERGM specifications on unrealistically small networks, [102] claims that the results can be extended to any ERGM with nodes labeled arbitrarily and no node level information included.

[114] also recognizes the issue of degeneracy as related to the discrete exponential family model, characterizing the issue as one of stability. He separately defines a stable distribution and a stable sufficient statistic. A distribution is stable if the maximum values of the negpotential func-

tion, $Q(\mathbf{y})$, over all possible graphs is bounded by some constant times the number of degrees of freedom, N, for large N. A stable sufficient statistic is one with a maximum value that is bounded in a similar manner. For the ERGM, the degrees of freedom, N, are equal to the number of possible edges, or $N = n(n-1)/2$ for a simple graph. The unstable distributions are characterized by excessive sensitivity and near degeneracy. Sensitivity is defined as distributions with unbounded nearest neighbor log-odds, where two possible graphs are considered nearest neighbors if the graphs are equivalent with the exception of a single edge. Six different ERGM specifications were analyzed to identify the regions of the parameter space which exhibited instability. All models contain a density term, the first three contain the 2-star, triangle, and both; the remaining three contain each one of the geometrically-weighted statistics, which we discuss in detail below. Regions of instability are identified for all six models. The first two are stable only when they are generalized to a Bernoulli graph, i.e., when the 2-star and triangle parameters are zero. The area of stability for the third model with a density, 2-star, and triangle term is also very restrictive. For the models with the geometrically weighted terms, the area of stability is non-negligible, providing some optimism; however, care is still needed to avoid the unstable parameter regions.

Identification of degenerate parameter space regions of an ERGM has also been examined by researchers outside the disciplines of statistics and sociology. [17] examine the mixing time for an MCMC of the ERGM specified with a density and triangle parameter. A high and low temperature regime are defined and identified for the parameter space. The authors showed that the algorithm mixes exponentially slow in the low temperature regime and as $\Theta(n^2 \log n)$ in the high temperature regime. However, the authors also showed that models with parameter values in the high temperature regime are asymptotically independent and thus are not appreciably different from the Erdös-Rényi graph. This coincides with the region of instability found for the model by [114] for the same ERGM specification. [93] and [94] investigate the degeneracy issue with analysis techniques from statistical physics for two ERGM specifications: the specification with a density and two-star parameter, called the two-star model, and the specification with a density and transitivity parameter, called the clustering model. For the two-star model, the authors used the Hubbard-Stratonovich transform and saddle-point expansions, to determine the region of the two-parameter space where degeneracy occurs. High and low density phases which are separated by a coexistence region are identified in a phase diagram. This coexistence region corresponds to a symmetry-broken phase. The symmetry breaking that describes the coexistence phase results in similar behavior as the stability defined by [114]. The separation of these three phases corresponds to a conventional continuous phase transition. As a result of this analysis the authors concluded that the degeneracy problem, at least in this particular ERGM, is analogous to a phenomenon in physics known as phase separation. The coexistence region was also identified for the clustering model in [94]. For this particular model, the degenerate region corresponds to parameter values that indicate a moderate number of triangles. The finding led the authors to conclude that without some augmentation, the clustering model will never adequately describe a real-world network.

In addition to the case studies mentioned above, special attention has been given to degenerate ERGM specifications that include a term for transitivity, and to a lesser extent to those with k-star terms. Inclusion of a term that accounts for transitivity has been shown to be important because the closure of triangles is a main feature that separates realized networks, at least for social networks, from those generated independently and at random [121]. However, incorporation of a term that

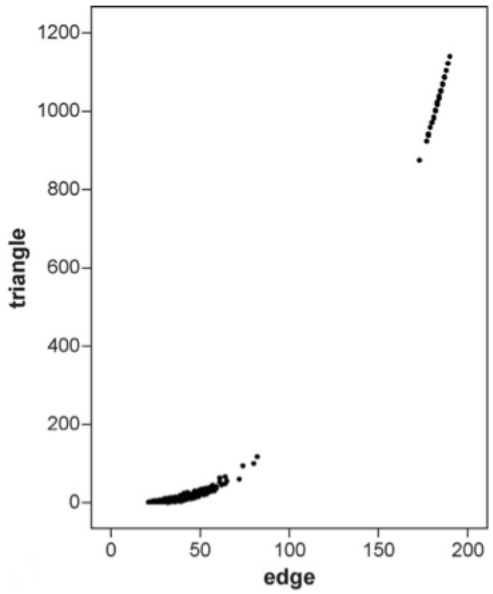


Figure 5.3. Scatterplot of number of edges against the number of triangles from a simulation study conducted by [104] for an ERGM with density parameter fixed at -1.5 and triangle parameter ranging from 0 to 1.

successfully reflects the amount of transitivity in the observed network has been a difficult task. This problem was explored through simulations in [104] who found that for an ERGM that included only a density and triangle term, an increase in the value of the triangle parameter does not correspond to a smooth increase in the number of triangles in the simulated graph. Instead what occurs is a tendency to resemble either a high or low density Erdös-Rényi graph, in agreement with the conclusion of asymptotic independence of the same model presented in [17]. The number of triangles in the simulations of [104] occur uniformly throughout the graph and form as a function of less or more edges with a dramatic jump, shown in Figure 5.3, between the two extremes. Therefore, the model is unable to adequately describe a graph with a moderate number of triangles. If the parameter estimate is obtained with MCMC-MLE, the parameter is estimated to be the value at which the jump occurs. The probability distribution of the statistic is bimodal with a combination of the low density graphs to the left of the jump and the high density graphs to the right [121].

One proposed reason for the difficulty in the representing transitivity with only a triangle parameter is the existence of other effects contributing to the formation of triangles [121]. For example, three relationships could form independently between three students in the same class, not because of a shared relation, but because of the shared covariate value. Even if a term is included in the model to explain the covariate effect, the two processes are not acting on the edges independently although they are modeled as such [41]. Another reason for the difficulty with modeling

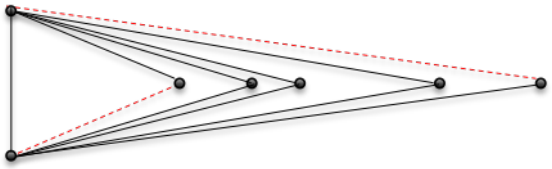


Figure 5.4. An example 5-triangle.

transitivity through a count of triangles is the manner in which the model introduces triangles does not correspond to how they appear in observed networks. The model wants to place the increasing number of triangles uniformly through the network, where it has been observed that groups of triangles tend to form dense "clumps" of triangles [104] . These clumps of edges are not completely connected [121] and thus the more triangles to which an edge is a part of, the less likely it is to be part of a new triangle. These observations have led to the formation of a new specification for representing transitivity, the alternating k-triangles statistic.

The alternating k-triangle statistic was proposed by [121] in an effort to model transitivity but avoid the avalanche effect that leads to degeneracy. The authors claim that transitivity in observed networks is important but complex, and that a statistic that merely counts the number of triangles is overly simplistic. A k-triangle is a set of k triangles that share a common edge, often referred to as the base; see Figure 5.4 for an example 5-triangle. The formula for the alternating k-triangle statistic is

$$AKT_\lambda(\mathbf{y}) = 3T_1 + \sum_{k=2}^{m-2} (-1)^{k+1} \frac{T_k(\mathbf{y})}{\lambda^{k-1}}$$

where T_k is the number of k-triangles. The increasing denominator gives decreasing probability to higher-order triangles as per the empirical observation noted above. The value of λ controls the type of transitivity. Larger values of λ lead to smaller probability given to higher-order triangles where smaller values lead to a localized effect [58]. The alternating signs of increasing terms also aim to prevent large cliques of edges.

The alternating k-triangle statistic is included in the ERGM as a term in the negpotential function with a single parameter coefficient. The value of λ can either be considered known or estimated. [121] consider the value fixed citing that the conclusions obtained remained constant for various values of λ. If this value is to be estimated, the ERGM is no longer within the standard exponential family as the negpotential is not a linear function of the parameters. [58] present the details for estimation in a curved exponential family for an ERGM where λ is to be estimated using MCMC approximation to the likelihood. This work does recommend estimating this parameter unless theoretical considerations imply a particular value.

[121] show that an ERGM with the alternating k-triangle statistic satisfies the partial conditional dependence of [95]. Thus, the dependence graph for this particular model is realization dependent. The authors argue for a more generalized dependence structure than the Markovian

dependence claiming that edges that are not incident could possibly be conditionally dependent. [104] claim that considering cliques of size greater than three is necessary to avoid degeneracy. The alternating k-triangle statistic allows for edges that, if realized, would create a four-cycle to be conditionally dependent. To make this connection more clear, the two red, dashed lines in Figure 5.4 are not incident, but would be conditionally dependent under the partial conditional dependence structure.

The authors ([121]) admit that inclusion of the alternating k-triangle statistic makes interpretation of model parameters more difficult and that it does not lead to a simple representation of dependency [115]. As a demonstration of the parameter interpretation, consider the case study of collaboration relationships between $n = 36$ partners in a law firm presented in [121]. One model fit to this data included the alternating k-triangle statistic and multiple attribute-based statistics. A significant coefficient for the k-triangle term is interpreted as evidence of a triangle formation process beyond what could be explained by considering only attributes. When an additional term is included to to represent the degree distribution, the significant k-triangle parameter indicates that transitivity is not the result of popularity of nodes. [104] interpret a significant and positive k-triangle parameter as indication of a core-periphery structure resulting from transitivity, rather than popularity. Although the k-triangle statistics appears "contrived", it is argued that a contrived statistic is necessary due to the complex processes that work together to create the static view of the network.

[56] proposes an alternative formulation for the k-triangle statistic based on a shared partner statistics which leads to more clear interpretation of parameters. The edgewise shared partner statistic, $EP_k(\mathbf{y})$ counts the number of edges that are realized with both nodes connecting to exactly k other nodes. The alternating k-triangle statistic is then equivalent to the geometrically-weighted edgewise shared partner statistic,

$$\text{GWESP}_\theta(\mathbf{y}) = e^\theta \sum_{i=1}^{m-2} \left\{ 1 - (1 - e^{-\theta})^i \right\} EP_i(\mathbf{y})$$

where $\theta = \log \lambda$ in the original formulation. This parameterization is particularly useful when the value of λ is to be estimated because of the desired restriction to positive values. With this formulation of the statistic, a significant, positive parameter would be interpreted as the more edges two nodes have in common, the less the motivation is to form more common edges.

The alternating k-triangle statistic was novel in its approach because rather than counting local configurations, the statistic attempts to summarize an entire distribution of subgraph counts [59]. Two other distribution-summarizing statistics were also presented by [121] with the same goal of decreasing the effects of degeneracy. The first is the alternating k-star statistic

$$AKS_\lambda(\mathbf{y}) = \sum_{k=2}^{m-1} (-1)^{-k} \frac{S_k(\mathbf{y})}{\lambda^{k-2}} \tag{5.8}$$

where $S_K(\mathbf{y})$ counts the number of k-stars. The alternating k-star statistic is an attempt to model the degree distribution. The Markov model of [31] proposed including terms for all k-stars, restricting it to only the first two to decrease the number of parameters from $n-1$ to 2. The alternating k-star

statistic also restricts the number of parameters from $n-2$ to 2, if the value of λ is to be estimated. Therefore, inclusion of the $ASK_\lambda(\mathbf{y})$ can be seen as modeling all k stars but maintaining a reduction of the parameter space. As the issue of degeneracy has been identified as a parameter space issue, limiting attention to a subset of the parameter space can lead to an increased probability that the MLE exists.

The other statistic introduced in [121] is the alternating k-twopath statistic

$$AKP_\lambda(\mathbf{y}) = \sum_{k=1}^{m-2} (-1)^{-k} \frac{P_k(\mathbf{y})}{\lambda^{k-2}}$$

where $P_k(\mathbf{y})$ counts the number of k-twopaths which are similar to a k-triangle, but without the base. The purpose of the alternating k-twopath statistic is to be used in conjunction with the alternating k-triangle statistic. If both statistics are included the k-triangle parameter can be interpreted exclusively as transitivity, rather than as transitivity and the necessary conditions for transitive closure. [56] provided equivalent geometrically weighed versions of $ASK_\lambda(\mathbf{y})$ and $AKP_\lambda(\mathbf{y})$ as well.

The most recent adaptation to counter degeneracy is the hierarchical ERGM of [115]. In this work, the issue of model degeneracy is attributed to large and growing neighborhoods, where a neighborhood of an edge refers to the set of edges on which the original edge is conditionally dependent. For a simple graph with Markovian dependence where all $\binom{n}{2}$ edges are possible, each potential edge has a neighborhood of size $2(n-2)$. Thus, as the number of nodes increases, so does the size of the neighborhoods.

The first feature of the hierarchical ERGM is a partitioning of the nodes into K local "neighborhoods". These "neighborhoods" are non-overlapping, although edges can form between them. This interpretation of "neighborhood" is more consistent with the concept of a community; thus, in order to avoid confusion, the rest of this description reflects this terminology. The community structure is not necessarily an observable feature of the graph, nor are the number of communities required to be known a priori. The second feature of the proposed model is local dependence and global independence. The distribution function of the graph is separated into a within- and between-community probability mass function (PMF). If $\mathbf{Y}_{(kl)} = \{Y_{ij} : i \in \mathcal{N}_k, j \in \mathcal{N}_l\}$ represents the edges that could form between nodes in community k and nodes in community l, the conditional probability function for the entire graph $\mathbf{Y}$ given the community membership, $\mathbf{X}$, is written as

$$P(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x}) = \prod_{k=1}^{K} P(\mathbf{Y}_{(kk)} = \mathbf{y}_{(kk)}|\mathbf{X} = \mathbf{x}) \prod_{k<l}^{K} P(\mathbf{Y}_{(kl)} = \mathbf{y}_{(kl)}|\mathbf{X} = \mathbf{x})$$

Each PMF can include a different set of sufficient statistics and thus parameters to estimate and each set of parameters for the between- and within-community PMF induce a different dependence structure on a subset of the graph. A Bayesian inference method is presented to estimate the number of communities, node community membership, and all sets of parameters. This computationally expensive procedure requires an approximation to the prior and an exchange algorithm to sample from the posterior.

5.1.9 Latent Variable Models

The subclass of probabilistic network models known as latent variable models encompasses a broad class of models that are hierarchical in nature. The common thread between latent variable models is the edge variables are specified as conditional distributions and considered conditionally independent given some latent variable, such as block membership or position within a social space. [113] claim that any model which treats the nodes as exchangeable can be represented as a latent variable model. In contrast to the ERGM where the interpretation of the model identifies highly probably edges, the focus of an analysis from a latent variable model is on the role and position of individual nodes of the graph. The goal of incorporating a latent structure is to account for some of the variability of the topological features in the observed network [38]. While the ERGM is really only one a single model with extensions, the latent variable category includes a variety of models. We discuss the random effects models, latent block models, and latent space models.

Random Effects Models The simplest random effects model is the p_2 model, a random effects version of the p_1 model of [55] introduced by [127]. Both the p_1 and p_2 models were developed for directed social networks and contain parameters to describe density, reciprocity, and node-level expansive/productive effects and popular/attractive effects. In the p_1 model the node-level parameters are modeled as fixed effects with a potentially unique value for each node. For identifiability purposes, constraint must be placed on this set of parameters. In contrast, the p_2 model treats node-level parameters as crossed random effects and aims to estimate the parameters of the underlying distribution from which they could have reasonably been drawn [38]. Covariates can be incorporated into the p_2 model, at the node-level with additional modeling of the attractiveness and productivity parameters and/or by modeling the density and reciprocity parameters as functions of dyad-level covariates. The coefficients of the covariates are modeled as fixed effects and hence the p_2 model can be viewed as a generalized linear mixed model [136]. The remaining node-level variability not explained by the covariates is incorporated into the correlated random effects. If there is no covariate information, the p_2 model is merely a more parsimonious version of the p_1 model [113].

Parameter estimation of the p_2 model was originally conducted via an Iterative Generalized Least Squares algorithm [127]. Because p_2 is a nonlinear model, the algorithm requires a linearization step based on a Taylor series expansion of the likelihood function around the current estimate of the parameters. [136] introduced three Bayesian methods for parameter estimation of the p_2 model. All approaches implement a Gibbs sampler, with separate updating steps for the random effects, covariance matrix, and fixed parameters. The full conditional distributions of the random and fixed effects cannot be directly sampled; thus, the three separate approaches explore three different proposal distribution for the required Metropolis-Hasting steps.

The other area in which random effects have been utilized for network analysis is by incorporating them into a generalized linear model (GLM) framework, first introduced by [51]. The purpose of the random effects are to model higher-order dependence. For example, within-node dependence can be represented as random intercepts on the random effects terms. Traditionally,

a GLM assumes observations are conditionally independent given regression coefficients, or fixed effects, whereas the model presented in [51] assumes conditional independence given the random effects terms. An estimation scheme is also presented where the regression coefficients, fixed effects, and covariances of the random effects are estimated with a standard Bayesian analysis.

The generalized bilinear regression model presented in [52] is a direct extension to the generalized linear mixed-effects model discussed above. The extension is to include a bilinear effect into the error structure to incorporate third-order dependence, or dependence between triples of random variables. This bilinear effect is also referred to as a reduced-rank interaction term and is an inner product of latent characteristic vectors. Latent characteristic vectors are then modeled as independent K-dimensional multivariate normal distributions with mean zero and diagonal covariance matrices. This inner product can be viewed as a mean zero random effect. As an application of this method, [52] analyze a valued network of international relations in central Asia.

The methods of [52] are applied in [54] to a network of bilateral trade in an analysis to determine how it is affected by various country attributes, such as capitalism, conflict, cooperation and democracy, among others. Although not statistically novel, through this approach the authors were able to explain three-fourths of the variability in the network by accounting for second- and third-order dependence over the standard gravity model that had traditionally been able to explain only one-half. In addition, through the random effects, correlations were modeled between importer, exporter, and dyadic relations, where they had previously been modeled as independent. One of the more interesting result of the analysis was that bilateral trade was not significantly impacted by conflict between two countries, although cooperation between two countries led to a significantly increased amount of bilateral trade.

5.1.10 Latent Blockmodels

In the most general sense, blockmodels are models which classify the nodes of the graph into groupings, often referred to as blocks. There are two, potentially overlapping, concepts of block structure. This first is that edges are more likely to form between nodes classified within the same block than between nodes in disparate blocks. This is the approach adopted by the computer science community where a block is often referred to as community. One of the main open challenges in this field is that of "community detection", or uncovering these groups of nodes [38]. The second definition of block is motivated by the notion of structural equivalence [28]. Two nodes are defined to be structurally equivalent if the relations between those nodes and all other nodes in the graph is equivalent. A relaxation of the this concept often referred to as stochastic equivalence can be descried as two nodes that relate to similar nodes in a similar manner. With this second definition, the purpose of blocking is to capture the main structural features of the network [122]. Within the statistics literature models that incorporate block structure have been referred to as stochastic blockmodels. This work can be partitioned into either a priori or a posteriori.

The a priori blockmodels were introduced by [129] and are presented as an extension of the p_1 model of [55]. These models do not fit under the heading of "Latent" Blockmodels as one assumption is that the block structure is observed. The block structure is incorporated into the p_1

model by inclusion of a block-specific parameter that accounts for block membership and adjusts the other parameters in the model, such as expansiveness and popularity, for this membership.

The obvious disadvantage of the stochastic blockmodels of [129] is the requirement to know the block membership a priori. When this information is unknown, blockmodels of the second type, a posteriori, are used to infer the group membership. An early attempt was made by [130] who fit the p_1 model to data, grouped nodes based on the estimates of the productivity and popularity parameters, and then, given the group structure, fit a pair-dependent stochastic blockmodel. The pair-dependent stochastic blockmodel specifies the joint distribution of the edge random variables given parameter values and the block membership.

Extensions to the pair-dependent approach were presented in [122] and [88] for undirected, binary graphs with only two blocks and directed, weighted networks and an arbitrary number of blocks, respectively. Both approaches include two assumptions: the number of blocks is known and given block membership, edge probabilities are independent. Block membership is inferred from the pattern of edges and estimated through a Gibbs sampler which alternatively samples the parameters of the model and the block membership of the nodes. Membership probabilities and parameters are assessed through posterior distributions. Parameter identifiability is an issue common for mixture models as the model can only distinguish between different partitions, not the distinct labeling of them. This leads to distinct parameter values resulting in the same probability distribution. One proposed solution is to impose order restrictions on the block probabilities; however, the method was shown to lead to poor group identification when probabilities of different blocks are similar. Therefore, the approach taken in [88] is to restrict attention to posterior distributions of functions which are invariant to relabeling.

Specification of blockmodels requires two components: the block model itself and an index of the nodes and the blocks to which they belong [38]. An extension of the second component is the mixed-membership stochastic blockmodel. For the previously mentioned stochastic blockmodels , there is a one-to-one mapping from node to block. In contrast, the mixed-membership model specifies an array of memberships for each node, so that the block to which the node belongs depends upon the node with which it would potentially join, thus, group membership is "context dependent" [28]. For directed graphs, each node's membership array is of length $2n-2$. The motivation for this model is that the parameters of the block model describe the global features of the graph, while the membership arrays capture the node-specific patterns [38].

5.1.11 Latent Space Models

The main idea of the latent space models is that the nodes of the graph can be represented in a low-dimensional, latent space and the distance between two nodes affects the probability that an edge forms between them. Given the position of the nodes, or rather the distance between them, the probability of the edges are conditionally independent. One purpose of this latent space is to incorporate the effects due to unmeasured covariates [128].

The latent space model was originally developed by [53]. In this work the authors proposed

two models: the distance model and the projection model. The models differ in the way in which the latent space is incorporated into the probability. Euclidean distance is used by the distance model, although the authors point out that any metric could be used. A metric over the latent space inherently incorporates reciprocity, through the symmetry requirement of a metric, and transitivity, through the triangle inequality. Due to its ease of interpretability, the distance model has been more widely used in practice [113]. The projection model incorporates the latent positions of two nodes through the projection of the position of one node onto the direction of the other. This is a measure of similarity two nodes share with respect to some characteristics and depends on the angle the positions create in Bilinear latent space [113]. The projection model has been shown to be most appropriate when the graph is strongly asymmetric. If additional covariate information is available, the model also allows this information to be explicitly incorporated.

As originally conceived, the latent space model is able to capture three important features of networks: transitivity, reciprocity, and homophily of attributes. One aspect of networks that the model cannot account for is that of clustering, also referred to as community structure. To incorporate this feature [50] introduced an extension named the Latent Position Cluster Model. This extension explicitly models clusters in the latent space using a mixture of spherical Gaussian distributions on the positions in the latent space. [70] extended this model even further by combining the approach of [50] and [52] in the Latent Cluster Random Effects Model. This model is able to account for heterogeneity of the nodes, in addition to the four previously mentioned features of networks. Node-specific random effects are added to represent sociality effects in an undirected network, and for directed networks a sender and receiver effect is specified for each node. The dimension of the latent space and the number of clusters is not assumed to be known a priori. Choosing these values is viewed as a model selection problem.

All of the approaches mentioned above can be cast as a generalized linear model as defined by three features: the error model, $\Pr(Y_{ij})$, the linear model, η_{ij}, and the link function $g(\mu_{ij}) = \eta_{ij}$ where $\mu_{ij} = E(y_{ij})$. The models described above are presented as an analysis method for unweighted graphs and thus the error model is $\Pr(Y_{ij}) = \text{Bernoulli}(\mu_{ij})$ with link function $g(\mu_{ij}) = \log\left(\frac{\mu_{ij}}{1-\mu_{ij}}\right)$. It is the linear model, η_{ij}, to which the additional terms are added. As an example, if Z_i represents the latent position of node i and $x_{i,j}$ a dyad-level covariate between nodes i and j, the link function, or conditional log-odds given model parameters (α, β) and latent positions, for the projection model of [53] can be written as

$$\eta_{ij} = \alpha + \beta' x_{i,j} + \frac{Z_i' Z_j}{|Z_j|}$$

The extension to consideration of clustering is to assume the Z_is are realizations from a finite mixture of multivariate distributions and the extension to account for node heterogeneity is accomplished by adding random effects terms to the linear model. When the latent space model is constructed in this manner the extension to a weighted graph is natural by specifying a different error model and link function. This approach was demonstrated by [70] on a network where the edges are counts and thus the error model used is $\Pr(Y_{ij}) = \text{Poisson}(\mu_{ij})$ with link function $g(\mu_{ij}) = \log(\mu_{ij})$.

Estimation of the unknown values of the latent space models can be accomplished through

either maximum likelihood or a Bayesian analysis of posterior distributions. [50] detail both approaches. The maximum likelihood is conducted in two stages. The first stage estimates the distances between the nodes in the latent space, a relatively simple task as the log-likelihood is a convex function of the distances. Multidimensional scaling is then used to determine the positions of the nodes. The second stage involves determining the MLE of the model parameters, (α, β), and the parameters of the Gaussian mixture model, when necessary. Although this approach is fast and simple, the two separate stages imply that information about the first stage is not used in the estimation of the parameters in the second, and vice versa. A more common approach is to estimate all unknowns simultaneously with MCMC sampling.

The Bayesian approach to estimation utilizes a Gibbs sampler to cycle through the model parameters, latent positions, and group memberships, if clustering is of interest. For most parameters, it is possible to specify a conjugate prior and sample directly from the posterior distribution, but not all, and thus some Metropolis-Hastings steps are also necessary. Although this approach incorporates all information into each step, it is more computationally intensive than its frequentist counterpart, and does not scale well for large graphs. To update the latent position of each of the n nodes requires the calculation of $n-1$ terms of the log-likelihood and the updating of the model parameters (α, β) requires all $O(n^2)$ terms be computed (either $n(n-1)$ for directed or $0.5 * n(n-1)$ for undirected) [98]. In practice, this estimation technique has been infeasible for $n > 1000$.

Due to the inability of the Bayesian estimation technique to adequately scale to large graphs, alternative methods have been proposed. [98] propose an approximation to the log-likelihood which is motivated by the approach of case-controls studies. This uses networks' general sparsity, i.e., a row or column in the adjacency matrix contains an order of magnitude more 0's than 1's. In the analogy, the 1's are the cases and the 0's are the controls and the proportion of non-ties, i.e., 0's, are approximated through sampling. The authors show that this approach reduces computation time from $O(n^2)$ to $O(n)$. Variational methods have also been used to approximate the Bayesian estimation of latent space models [60].

Another issue with a Bayesian analysis of the latent space models is that the likelihood is invariant to rotation and reflection of the nodal positions and if a Euclidean space is used, also to translation [60]. In addition, if clustering is considered, the likelihood is invariant to the relabeling of the clusters, or the "label-switching problem" as it is known in mixture models [50]. The proposed solution to these issues requires a post processing of the MCMC output. The former is addressed by performing a Procrustean transformation on the posterior draws so that the result is close to a reference configuration, typically the MLE of the positions centered at the origin [53]. Alternatively, the framework proposed by [50] aims to correct both identifiability issues by minimizing the Bayes risk relative to a Kullback-Leibler divergence. In this approach the goal is to find a configuration that gives edge values closest to the posterior predictive distribution.

V	Set of nodes/vertices
E	Set of edges
n	Number of nodes/vertices
m	Number of edges
$\mathbf{Y}$	$n \times n$ adjacency matrix
Y_{ij}	Random variable representing possible edge ij
$d(i)$	Degree of node i
γ	Power Law exponent
C	Clustering coefficient
$\mathscr{G}$	Set of all possible graphs
G	a graph
$Q(\mathbf{Y})$	Negpotential function
$\delta_g(\mathbf{y})_{ij}$	Change statistic

Table 5.2: Table of notation for Section 5

5.2 Local Structure Graph Models (LSGM)

The local structure graph model (LSGM) is a new class of graph models defined in terms of global structure with interpretable and controllable local dependence. Two defining characteristics of the LSGM are the specification for each potential edge, $y(\mathbf{s}_i)$, of a full conditional distribution, $P(y(\mathbf{s}_i)|y(\mathbf{s}_j); j \neq i)$, and a neighborhood, $N_i = \{j : \mathbf{s}_j \text{ is a neighbor of } \mathbf{s}_i\}$. These two features taken together with an assumption of Markov dependence induce a direct functional dependence between those random variables defined to be neighbors,

$$\Pr(y(\mathbf{s}_i)|y(\mathbf{s}_j); j \neq i) = \Pr(y(\mathbf{s}_i)|y(\mathbf{s}_j); j \in N_i)$$

The probability of the presence of an edge is dependent upon the value of its neighbors.

The features of the LSGM were motivated by the Markov Random Field (MRF) model, a common tool used in the analysis of geo-referenced data such as images or spatially-located data. The modeling goal of both applications is to model dependence between locations that are "close" to each other, where "close" is defined with respect to some metric. Intuitively, the value at a particular location should be influenced by those sites which are nearby and less so by those that are further away. The MRF, through the definition of neighborhoods and conditional specification, allows one to incorporate the spatial dependence between the pixels in an image or the data collection sites and leads to the acknowledgement and specification of a local structure. Typical choices for neighborhoods are the four- or eight–nearest neighbors for data collected on a regular grid or common border for data collected from spatial areas, such as states or counties.

The connection between a MRF and network analysis is not novel. In fact, to any MRF corresponds an acyclic algebraic graph with undirected edges [63]. However, in it common application, the collection of random variables defining the MRF represent the nodes of the graph. In the previously mentions applications, random variables would be assigned to the pixels of an image or

locations on a map because it is the value at the pixel or the status at the location that is of primary concern. The model then assigns each node in the graph a conditional distribution and a neighborhood. In contrast, the LSGM defines the random variables and neighborhoods on the edges of the graph, as it is the existence of the edge that is of interest.

In this regard, the LSGM can be interpreted as incorporating an additional level of modeling to the common MRF approach. To see this, consider the neighborhood structure of the LSGM as represented through a dependence graph, a concept presented in [31]. In the dependence graph each possible edge of the original graph is assigned to a node. A connection between nodes in the dependence graph indicates the corresponding random variables are conditionally dependent. Two example networks and dependence structures with resulting dependence graphs are shown in Figure 5.2. The first example demonstrates the Markovian dependence of the ERGM as two edges are conditionally dependent if they are incident. The second example demonstrates the flexibility in the definition of a neighborhood. For this dependence structure, two edges are conditionally dependent if they connect the same number of red, or odd-numbered nodes. The dependence graph is comprised of three disconnected, yet internally fully connected subgraphs of edges that join the same number of red nodes. Because the nodes of the dependence graph represent the edges of the original graph, the MRF is placed on the nodes of the dependence graph. Thus, the dependence graph corresponds to the spatial locations or image pixels in the common application of a MRF. The additional level of modeling is that the nodes represent the edges of a different graph.

The idea of a neighborhood within network analysis has also been used elsewhere, although the definition of a neighborhood has not been consistently applied. Within the LSGM, a neighborhood contains edges which are conditionally dependent. Neighborhoods are restricted to be symmetric so that $i \in N_j$ implies $j \in N_i$. In this definition, neighborhoods are overlapping and a neighborhood is defined for each potential edge in the network. Other works have used the term neighborhood to partition the nodes of the network. For these approaches, the idea of a neighborhood is more closely aligned with a community or block structure where nodes within the same community are treated either equivalently or similarly [115].

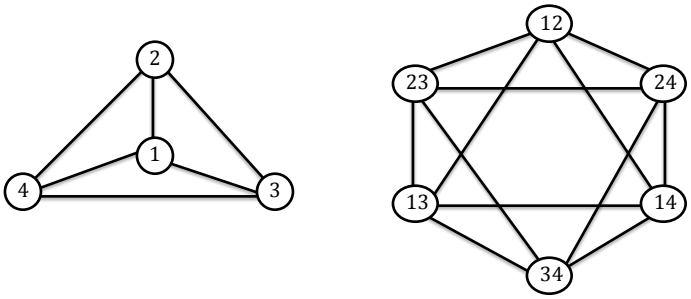
The focus of this article is on simple networks, thus, the conditional distributions are binary distributions. However, this approach can be generalized to weighted graphs by considering a different conditional distribution, such as Beta or Gaussian. The binary conditional distribution expressed in exponential family form is

$$f_i(y(\mathbf{s}_i)|\mathbf{y}(N_i)) = \exp\left[y(\mathbf{s}_i)A_i(\mathbf{y}(N_i)) - B_i(\mathbf{y}(N_i))\right] \tag{5.9}$$

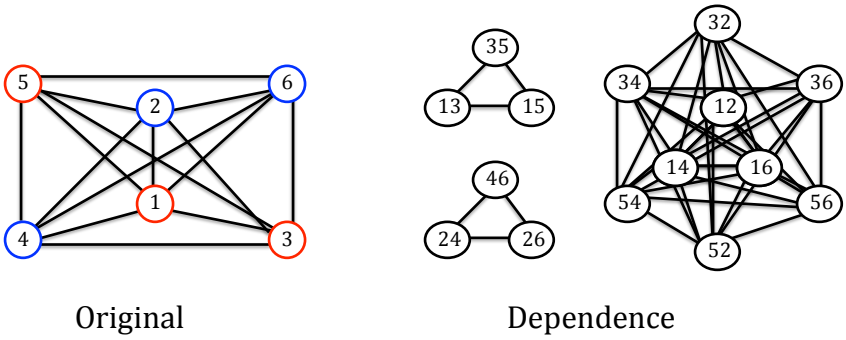
where $\mathbf{y}(N_i)$ represents the values of the random variables in the neighborhood of $y(\mathbf{s}_i)$. The dependence among the random variables is modeled through the function A_i which is referred to as the natural parameter function and B_i, a function of A_i. For binary conditionals, the natural parameter function takes the form:

$$A_i(\mathbf{y}(N_i)) = \log\left(\frac{\kappa_i}{1-\kappa_i}\right) + \sum_{j \in N_i} \eta_{i,j}[y(\mathbf{s}_j) - \kappa_j] \tag{5.10}$$

and the form of B_i is $\log[1+\exp(A_i(\mathbf{y}(N_i)))]$. The sets of parameters, $\{\kappa_i : i \in E\}$ and $\{\eta_{i,j} : i \in E, j \in N_i\}$ represent the global and local structure of the network, respectively, and will be



(a) Incidence definition of dependence. Image recreated from [31]



(b) Two edges are conditionally dependent if they connect the same number of red, or odd numbered, nodes.

Figure 5.5. Two example networks and dependence structures with resulting dependence graphs.

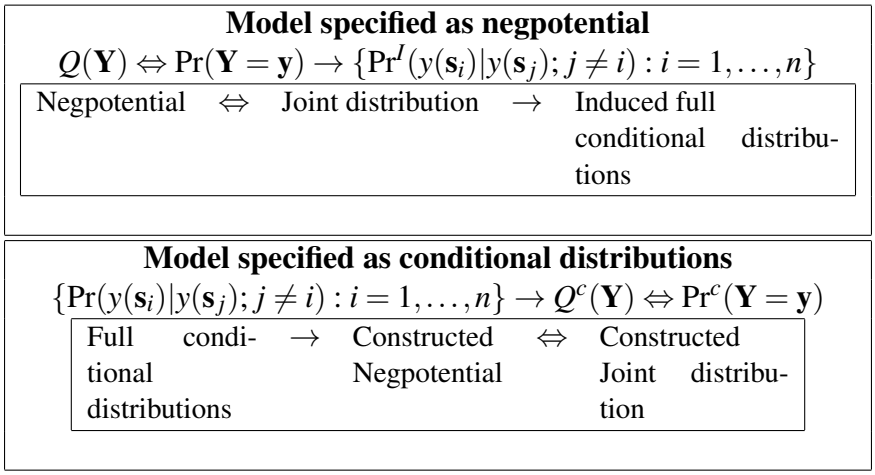


Figure 5.6. Relationship between the negpotential, joint distribution, and full conditional distributions when either the model is specified as the negpotnetial or full conditionals.

discussed in detail below. It should be noted that the parameterization of the natural parameter function in Equation 5.10 is the centered version of [20] and [62]. This parameterization has been shown to separate the global from the local structure leading to a more independent interpretation of parameters, as long as the value of η does not become "too large". We will reserve a discussion of what is meant by "too large" for the section on model parameters.

The specification of any collection of conditional distributions does not necessarily lead to a valid joint distribution on $\mathbf{Y}$, thus there are certain conditions that must be satisfied. [63] detail the requirements for a joint distribution to exist and correspond to the set of specified conditional distributions. For the construction of a LSGM, the condition to check is that term of the summation in the natural parameter function is symmetric for any pair of edges. This implies that neighborhoods and η-parameters must be symmetric, i.e., $i \in N_j$ implies $j \in N_i$ and $\eta_{i,j} = \eta_{j,i}$. If the joint distribution exists, it is uniquely determined by the set of conditionals [4].

Section 5.1.6 discussed how an ERGM is specified by determining appropriate configurations, or parameters and statistics, to include in the negpotential function of Equation 5.5. The negpotential function is given much attention because defining the negpotential defines the joint distribution, up to a constant. Thus, a particular ERGM is specified by defining the joint distribution for all edge random variables. This specification implies a set of induced full conditional distributions but does not directly model them. In contrast, the LSGM is defined by specifying a set of full conditional distributions which then imply a constructed negpotential function and thus joint distribution. This relationship between the two different methods of model specification is demonstrated in Figure 5.6.

The constructed negpotential function for the LSGM has been determined [62] to be

$$Q^C(\mathbf{Y}) = \sum_{i=1}^{n} y(\mathbf{s}_i) \left[\log\left(\frac{\kappa_i}{1-\kappa_i}\right) - \sum_{j \in N_i} \eta_{i,j}\kappa_j \right] + \sum_{i=1}^{n} \sum_{j \in N_i} \eta_{i,j} y(\mathbf{s}_i) y(\mathbf{s}_j) \tag{5.11}$$

This function implies the LSGM fits into the framework of the traditional ERGM. The difference is the way in which the model is specified.

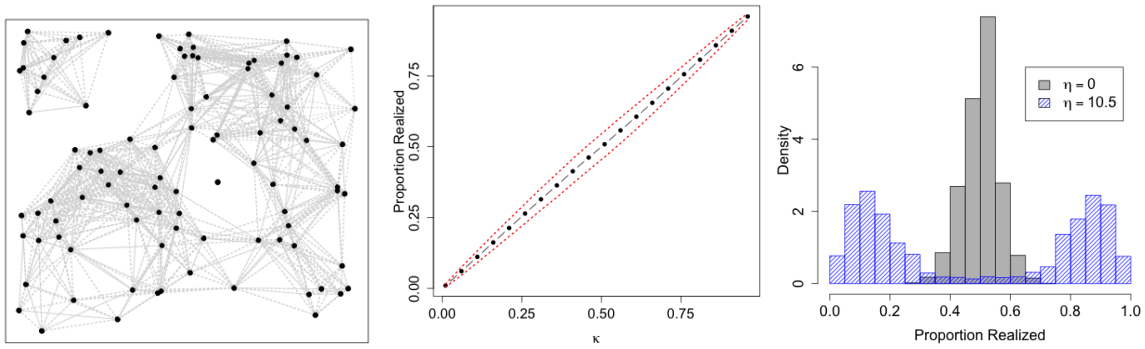
Network features can be partitioned into those that affect the global structure and those that affect the local structure. The existence of these two structures has been recognized in the analysis of empirical networks [115], [39]. The global structure is defined as patterns seen in the overall network, such as density. Features that allow for departures from the global structure at a local level would be classified as local structure. An example of the local structure is transitivity. By specifying the model through conditional distributions the local structure is directly modeled and thus there is greater control over it. According to [13], specification of a model through conditional distributions leads to a more natural generation. Thus, if it is desired to model and understand the local structures in a network, this can be achieved by specifying the model with the LSGM [62].

Again, other network analysis approaches have specified a model through conditional distributions. Most notably, the latent variable models specify the edge random variables as conditional distributions which are independent given some latent variable, such as block membership or position in social space. In this class of models, the edge random variables do not have a neighborhood structure, because there is no modeling of dependence between the edges. The combination of a neighborhood definition and conditional specification for each potential edge of the graph makes the LSGM a unique approach within the realm of network analysis which not only incorporates a local structure, but it allows the modeler to control it.

5.2.1 Model Parameters

The natural parameter function of Equation 5.10 contains two sets of parameters, $\{\kappa_i : i \in E\}$ and $\{\eta_{i,j} : i \in E, j \in N_i\}$. Stated in its most general form allows for a different κ_i for every edge $y(\mathbf{s}_i)$ and a different $\eta_{i,j}$ for every $y(\mathbf{s}_j) \in N_i$. Restrictions must be placed on this set of parameters for model identifiability. The effect of the model parameters will be demonstrated for the simplest case where $\kappa_i = \kappa \quad \forall i \; \eta_{i,j} = \eta \quad \forall i, j \in N_i$. The example network displayed in the first panel of Figure 5.7 will be used to illustrate the features of the model parameters. This network is composed of 97 nodes and 824 possible edges, those of which are displayed in Figure 5.7. If two nodes are not connected in this plot, the probability this edge will be realized is set to zero.

The first parameter considered is κ. The κ parameter represented the large-scale structure of the network and controls the density or proportion of possible edges which are realized. Because this parameter represents a proportion, it has a well defined parameter space over the interval $(0,1)$. This parameter can be interpreted as the marginal mean of realized edges. The second panel of Figure 5.7 displays the resulting proportion of edges realized in the example network when κ is varied over its parameter space and $\eta = 5$ is fixed. Ten-thousand networks were simulated for each κ



(a) Example used to demonstrate effect of model parameters. Gray lines indicate all possible edges.

(b) The effect of varying κ on the proportion of possible edges realized. Points represent the median of 10,000 simulations and red, dashed lines 95% envelopes.

(c) Proportion of the 64 possible edges in the disconnected group in the Northwest corner realized for $\kappa = 0.5$ in 10,000 simulations.

Figure 5.7. Example which demonstrates the effect of model parameters.

value with a burn-in of 10,000 while thinning with 500 simulations between each simulation retained. The points in the plot represent the median proportion realized in the 10,000 networks and the red, dashed lines represent 95% envelopes. This indicates there is a strong, monotonic relationship between the value of κ and the resulting proportion of edges realized with little variability, especially towards the boundaries of the parameter space.

The effect of the second parameter is less obvious. The parameter η represent the local, or small-scale, structure and can be interpreted as a dependence parameter. This parameter controls the extend to which groups behave together or independently. If $\eta = 0$, there is no dependence between edges, and the value of an edge's neighbors does not affect its probability of being realized. This is most clearly seen by examining Equation 5.10. For this case, the probability of edge formation is an independent Bernoulli trial with success probability κ. For larger values of η, the value of the neighbors more strongly influence the probability of an edge realization, thus forcing an edge to be more like its neighbors.

The final panel of Figure 5.7 summarizes the proportion of 64 possible edges in the disconnected NorthWest clump realized for each of 10,000 simulations obtained from two models. Both models have κ of 0.5; the gray, solid histogram represents $\eta = 0$ and blue, dashed histogram a value of $\eta = 10.5$. For the independence scenario of the gray, solid histogram, the probability of each edge is 0.5, regardless of the rest of the network. The resulting histogram is symmetric and centered at κ with few of the simulations resulting in less than 40% of the edges in the group realized or more than 60% realized. When the value of η is large, and thus the dependence is strong, this grouping of edges tends to behave more like a cohesive group, with most edges either present or absent. This is displayed in the histogram as a bimodal distribution with a mode corresponding to the group mostly realized and a mode when the possible edges are predominately absent. Note that the histogram for strong dependence scenario is still centered at the value of $\kappa = 0.5$. This

implies that the marginal mean is preserved over the multiple simulations, even when the local dependence is strong. The centered parameterization of the natural parameter function in Equation 5.10 allows for this behavior.

Often there is a need for additional modeling of the dependence parameter. In the models presented here, this term has been adjusted to account for unequal neighborhood sizes. In its common application, the MRF is applied to situations for which the neighborhoods are of equal or nearly equally-sized. One example is the four-nearest neighbors on regular lattice. If this grid is wrapped on a torus to eliminate edge effects, each random variable will have exactly four neighbors. The approach presented here will not result in equally-sized neighborhoods, but rather a distribution of sizes (see Figure 5.12 in Section 5.3 for an example). So that the summation term in the natural parameter function of Equation 5.10 has a uniform effect on edges of varying neighborhood size, the additional modeling of the dependence parameter is

$$\eta_{i,j} = \frac{\eta}{|N_i| + |N_j|} \tag{5.12}$$

where $|N_k|$ represents the size of the neighborhood of edge $y(\mathbf{s}_k)$. The summation of neighborhood sizes in the denominator assures that $\eta_{i,j} = \eta_{j,i}$, a requirement for the existence of the joint distribution [74].

The parameter space for η is not as clearly defined as that for the marginal mean. The previously mentioned issue of model degeneracy occurs when the local structure dominates the global structure which was recognized by [115] as an issue of large and growing neighborhoods. A large dependence parameters will have the same effect on the probability of edge realization as many neighbors contributing to the summation. For example, the proportion of realized edges in 10,000 simulations of the graph shown in the first panel of Figure 5.7 for parameter values $\kappa = 0.5$ and $\eta = 35$ is shown in Figure 5.8. In this situation, the model places most of its probability on the nearly complete graph, and thus almost all edges are realized in all simulations. A further complication is that the values for which the model breaks down is not consistent between applications. All that can be said is that η should not be "too large". The recommendation by [62] is to simulate from the fitted model to assure that the simulations appear reasonable given the observed network. Further work in this area is a topic of ongoing research and will be further addressed in (paper 2, in preparation).

5.2.2 Optional Features

To specify the LSGM, one must specify the form of the conditional distributions, which for simple graphs is a binary distributions, and for each edge random variable, $y(\mathbf{s}_i)$, a neighborhood, N_i. To aid in the specification of the neighborhood structure two optional modeling features were considered: a potentially latent spatial location of the nodes and a saturated graph.

The first feature is the definition of a spatial location for the nodes. Again, node locations are not necessary to the construction of a network; however, it can aid in the specification of the second feature, a saturated graph, and the required neighborhoods. If the goal is to generate a

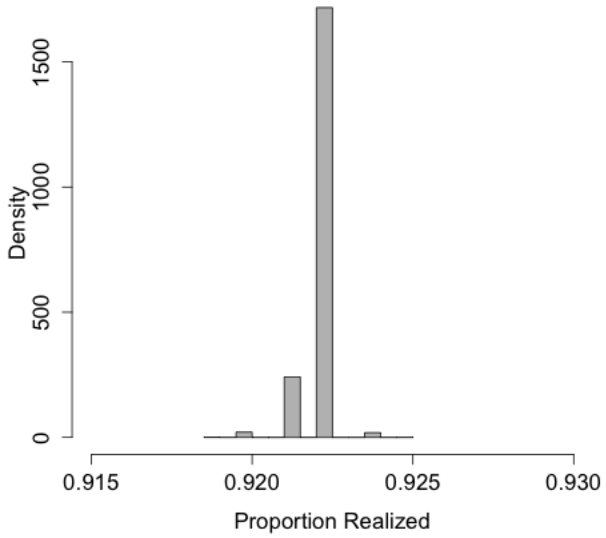
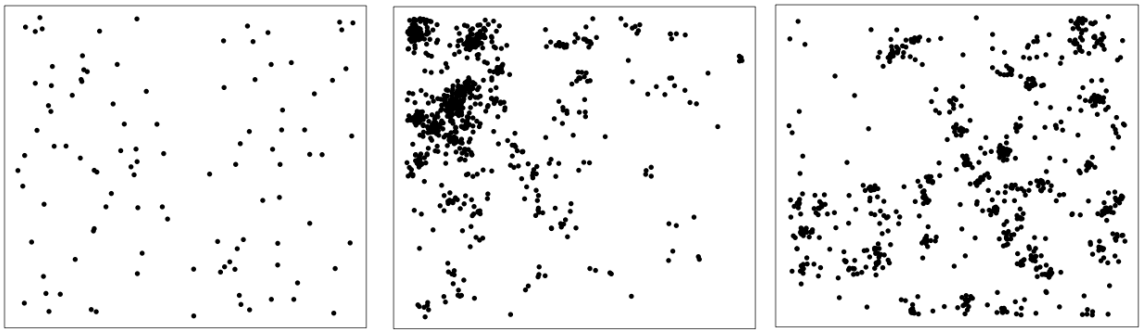


Figure 5.8. Proportion of realized edges in 10,000 simulations when $\kappa = 0.5$ and $\eta = 35$. The proportion realized does not correspond to the marginal mean $\kappa = 0.5$. This is an example of an area of the parameter space where the model is degenerate.

realistic network where the nodes have spatial location, such as locations of power lines in a power grid or routers of the Internet, this information can be naturally incorporated here. This approach will be used in the next section with the analysis of a network of tornadoes. If no location is observed, node location can also be imposed through a latent, random process. This latent process can place nodes randomly in the space through a point process or through a model-based approach which considers attribute information.

Node locations resulting from three different point processes will be demonstrated. The first and simplest is a homogeneous Poisson point process which results in complete spatial randomness. The number of nodes are chosen from a Poisson distribution with a specified intensity parameter λ. The x and y locations of each node are drawn independently from uniform distributions over the corresponding intervals of consideration. An example of the locations of the 105 nodes resulting from a homogeneous Poisson point process with intensity parameter $\lambda = 100$ is shown in the first panel of Figure 5.9. An immediate extension is to define an Inhomogeneous Poisson point process with an intensity function that depends on node location [23, page 620]. This method allows for a large amount of flexibility in the node placement. An example with intensity function $\lambda(x,y) = \exp(2-4x+3y)$ is shown in the middle plot of Figure 5.9. In this example nodes are most dense in the upper-left hand corner and become sparse as x-coordinate increases and the y-coordinate decreases. The final example is the Neyman-Scott process. Node locations resulting from this method are clustered which can be useful in generating networks thought to contain a community structure. The far-right panel of Figure 5.9 displays the node locations for a Neyman-Scott process generated with a Cauchy cluster kernel [34]. If covariate information is available on



(a) Homogeneous Poisson point process with $\lambda = 100$.

(b) Inhomogeneous Poisson point process with $\lambda(x,y) = \exp(2-4x+3y)$

(c) Neyman-Scott process

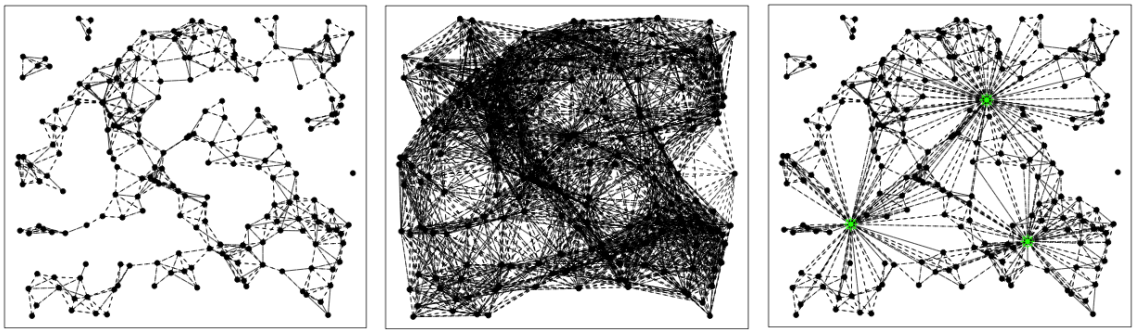
Figure 5.9. Examples of random node placements through different point processes.

the nodes that is assumed to affect the probability of an edge, this can be used to assign location through a model based approach. One possible way in which nodal covariate information can be incorporated into defining nodal spatial location is the unobserved "social space" of [53] and [50].

The other optional feature is the definition and use of a saturated graph. A saturated graph identifies edges with a positive probability of being realized. In many applications it is natural to impose a defined cutoff for which edge formation is possible. For instance, Sensor-Actuator Networks (SAN) have a common transmission range which is the maximum distance possible between two connected nodes [89], and the growth factors and diffusible signaling concentration decreases as a function of distance in biological networks making long distance edges highly improbable [123]. The use of the saturated graph was actually presented in the example network of Figure 5.7.

The saturated graph will be defined in this work in a manner similar to the formation of a unit disk graph [71]. Given a radius, r, an edge between two nodes within distance r will have a positive probability of being realized. Figure 5.10 displays examples of saturated graphs with constant node locations and various radius sizes. The first two panels display the resulting saturated graph when a constant radius size is used for all nodes. Note the vastly different patterns that emerge as a result of the different radius sizes. With a smaller radius size, the graph is not completely connected with two clusters of nodes disconnected from the majority and one isolated node. With a larger radius size, the saturated graph is now completely connected with no isolated nodes and many more possible edges. A common feature of graphs examined in the computer science literature has been the existence of a giant component. One way to allow for this to occur in the saturated graph is to vary the radius size between nodes. The last panel in Figure 5.10 has a radius size of $r = 0.1$ for all except the three nodes highlighted in green with radius size $r = 0.35$.

One advantage to imposing a saturated graph is a decrease in the number of random variables under consideration. For example, there are 213 node in the graphs of Figure 5.10. The saturated graph with the small radius size allows for 668 possible edges. When the radius size in increased



(a) Saturated graph defined with $r = 0.1$.

(b) Saturated graph defined with $r = 0.25$.

(c) Saturated graph defined with $r = 0.1$ for all nodes except the three in green with $r = 0.35$.

Figure 5.10. Examples of saturated graph on same set of nodes for various radius sizes.

to $r = 0.25$, this increases the number of possible edges to 3467 and the combination of radius sizes considers 873 possible edges. If there were no saturated graph and edges between all pairs of nodes were considered, there would be $\binom{213}{2} = 22,578$ random variables to model. In a small example it is possible to consider an edge between all pairs of nodes; however, the direction of current research is to analyze networks with a large number of nodes [28]. Thus, considering an edge between all pairs of nodes will be computationally prohibitive for many of the networks currently of interest. The definition of a saturated graph is an attempt to lessen the effect of this issue.

In addition, the introduction of a saturated graph will affect the size of the resulting neighborhoods. Assume an incidence definition of dependence is used so that two edges which share a node are conditional dependent. In the plots of Figure 5.10, in the absence of a saturated graph, each edge random variable would contain $2(213-2) = 422$ neighbors and thus each summation in the natural parameter of Equation 5.10 would include 422 terms. When using a saturated graph the neighborhood size is not consistent for all random variables, but rather depends upon the number of close edges. The average neighborhood size for the first panel of Figure 5.10 is 12.5, second plot is 68, and for the final plot each edge is affected by, on average, 30.67 neighbors. The largest neighborhood in any of the three examples is only 100. Thus, the use of a saturated graph not only decreases computational time significantly, but also decreases the large and growing neighborhood size effect, which was identified by [115] to contribute to model degeneracy. Note that the saturated graph as presented in the examples presumes the nodes have spatial locations. Although this approach is intuitive and illustrative, it is not the only possibility. Factors other than distance between nodes can be incorporated to eliminate the possibility of edge formation.

5.3 Application

The features of the LSGM will be demonstrated on an example network constructed from the recorded tornadoes which originated within the state of Arkansas during April, 2011. This application was chosen because the nodes have an observed spatial location and the example is able to demonstrate the features of the LSGM while remaining simple. Details of the tornadoes were obtained from National Oceanic and Atmospheric Administration's (NOAA) National Climatic Data Center (NCDC) *Storm Data* severe weather report database. Tornadoes with a documented starting longitude and latitude are included in the analysis. Fifty-nine tornadoes met the criteria.

5.3.1 The Network

The nodes of the network represent the 59 tornadoes which originated within Arkansas during April, 2011. The location of the nodes are the location of the observed point of origin of the tornado. Tornadoes from the same family, or those which arose from the same storm event, are also identified [1]. The location and family information, as indicated by color and number, of the tornadoes are shown in Figure 5.11. For successively occurring tornadoes to be within the same family, both must appear within two hours. This criteria was used to account for overnight storms which spanned more than one day. An exemption was made for events 8, 9, 10, 11, and 12 which all occurred within two hours, but due to their location and ordering could not have physically originated from the same system. Thirteen storm events were identified to have produced at least one tornado.

Edges will connect two tornadoes which originated from the same storm event. The use of a saturated graph will be demonstrated on the Arkansas tornado network. The radius which defines the possible edge formation will be $r = 80$ kilometers. This radius value is motivated by the fact that thunderstorms can travel upwards of 80 kilometers per hour. The resulting saturated graph contains 292 possible edges, 125 of which are realized, i.e., connecting tornadoes from the same event.

To further justify the use of a saturated graph consider if this feature had not been used. Instead of 292 random variables to model, there would be $\binom{59}{2} = 1,711$ edge random variables of interest. Using the incidence definition of dependence, each edge random variable would have neighborhoods of size $2(59-2) = 114$. In contrast, the saturated graph allows for a distribution of neighborhood sizes. This distribution is shown in Figure 5.12 with an average value of 24.43 and the largest neighborhood containing 39 edges. Restricting the neighborhoods is also intuitive. A edge which connects two tornadoes in an area of the state where few tornadoes occurred, e.g., event 3 in Figure 5.11, is not dependent upon edges which occur in an area where many tornadoes occur on the other side of the state, e.g., event 7.

In summary, the Arkansas tornado network consists of 59 nodes which correspond to the starting latitude and longitude of tornadoes within the state of Arkansas during April, 2011. An edge exists between two nodes if they are in the same family, i.e., resulted from the same event. The

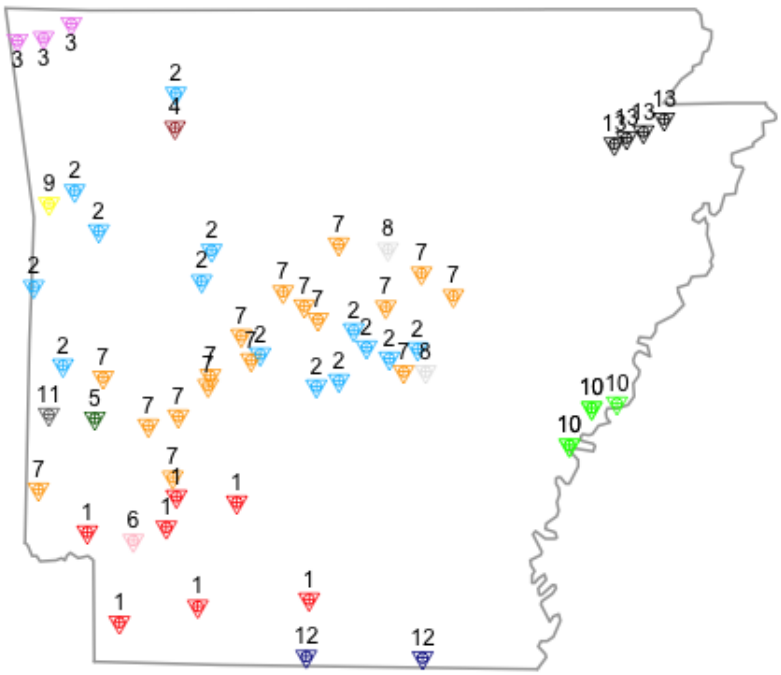


Figure 5.11. Nodes of the network as defined by the tornadoes that originated in Arkansas during April, 2011. Color and numbers correspond to the event in which the tornado occurred.

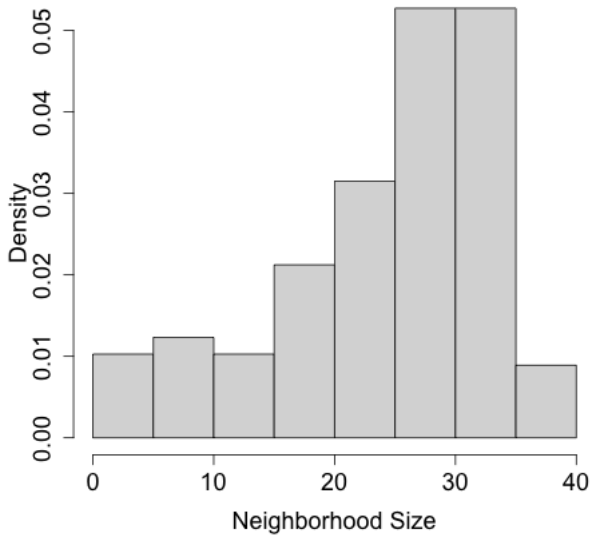


Figure 5.12. Distribution of the neighborhood sizes when a saturated graph of $r = 80$ kilometers is used in the analysis of the Arkansas tornado network.

definition of neighborhoods will be incidence and a saturated graphs will be used restricting edges to connect two tornadoes which are at most 80 kilometers apart.

5.3.2 The Fit of the LSGM

The Local Structure Graph Model with a single marginal mean, κ, and single dependence parameter, η, is fit to the Arkansas tornado network. The dependence parameter will be adjusted to account for unequal neighborhood sizes as in Equation 5.12. Point estimates of the model parameters are obtained through a maximization of the log pseudo-likelihood (PL), the summation of the log of the conditional distributions,

$$\log \mathrm{PL} = \sum_i \{y(\mathbf{s}_i) \log[p_i(N_i)] + (1 - y(\mathbf{s}_i)) \log[1 - p_i(N_i)]\}$$

where $p_i(N_i)$ represents the conditional expectation for $y(\mathbf{s}_i)|N_i$,

$$p_i(N_i) = \frac{\exp(A_i(N_i))}{1 + \exp(A_i(N_i))}$$

The PL function was introduced by [14] and is not a "true" likelihood but rather an approximation to the likelihood function. It provides a fast and computationally tractable method to obtaining approximations to the maximum likelihood estimates when an intractable normalizing constant prevents a direct maximization. Estimates obtained by maximizing the PL function for a MRF

	$\hat{\kappa}$	$\hat{\eta}$
LSGM	0.27 (0.15, 0.75)	8.60 (4.93, 11.07)
Independence	0.43 (0.38, 0.48)	—

Table 5.3. Point estimates, 90% interval estimates and p-value for the proportion of neighborhood model assessment technique for the LSGM and Independence models.

have been shown to be generally consistent and asymptotically normal [46]. Quantifying uncertainty in PL estimators is less known and so interval estimates will be obtained through parametric bootstrap. Networks are generated from the estimated values of κ and η using a Gibbs sampler. Ten-thousand simulations were considered with an equal-sized burn-in period and thinning of 500 iterations. For each simulated network a value of κ and η is estimated. Percentile confidence intervals were obtained by taking the 5th and 95th percentiles of the distributions of estimates. The resulting point estimates and 90% confidence intervals are shown in the first row of Table 5.3.

For comparison purposes, a maximum PL estimate and parametric bootstrap interval are obtained for the one parameter independence model where the dependence parameter is forced to be zero. The results of this fit to the Arkansas tornado network are also shown in Table 5.3. Two methods of model comparison will be presented. The first is an approximation to the likelihood-ratio test using simulations and the second attempts to quantify how the LSGM is able to replicate the local structure of the network.

The likelihood ratio test is a commonly used method to compare two models with nested parameter spaces. Because the likelihood is known only up to a constant for the LSGM, this exact method cannot be used to compare the LSGM to the independence model. However, an approximate approach based on simulations and the PL can be used. It is desired to test the fit of the null model, here independence model, against the fit of the alternative model, or LSGM. The test statistic will be the different in the log-PL value for both methods, or

$$D = \log \mathrm{PL}(\mathrm{LSGM}) - \log \mathrm{PL}(\mathrm{Indep}) \tag{5.13}$$

In order to assess the significance of this test statistic, a reference distribution must be constructed through simulations. This is done by fitting both the LSGM and independence model to the each of the 10,000 networks simulated from the fitted independence model with $\hat{\kappa} = 0.43$. From the fit of both models and for each simulations, the log-PL value is computed. The value of Equation 5.13 is then computed for each simulation, D_h^*, $h = 1, \ldots 10000$. The p-value testing if the fit of the LSGM is significantly better than the independence model is computed as

$$\frac{\sum_{h=1}^{10000} I(D_h^* > D)}{10000} \tag{5.14}$$

where $I(A)$ is the indicator function which takes the value 1 if A is true and 0 otherwise. For the Arkansas Tornado Network the fit of the two models yields $D = 14.06$ with a p-value of 0.0016.

Thus, it can be concluded based on this test that the LSGM fits the Arkansas Tornado Network significantly better than the independence model. This implies that there is a significant amount of local dependence which should be accounted for in the model.

The results of the simulation-based PL ratio test indicate that there is a significant amount of local dependence, or structure, in the Arkansas tornado network. It is now of interest to quantify how much better the LSGM is capturing the local-ness of the network with the inclusion of the dependence parameter. To assess this trait, we will examine how well the simulations from the fitted models are able to recreate the observed neighborhoods. As a reminder, each potential edge is assigned a set of edges on which it is conditionally dependent. The feature of interest will be the proportion of an edge's neighborhood which assumes the same value as that edge. For example, if edge at location $\mathbf{s}_i$ is not present, i.e., $y(\mathbf{s}_i) = 0$, the proportion of neighbors which assume the same value can be computed as

$$q(\mathbf{s}_i) = \frac{1}{(|N_i|)} \sum_{j \in N_i} \left[1 - y(\mathbf{s}_j)\right]$$

and if the edge at location $\mathbf{s}_i$ is realized, i.e., $y(\mathbf{s}_i) = 1$, the proportion is

$$q(\mathbf{s}_i) = \frac{1}{(|N_i|)} \sum_{j \in N_i} y(\mathbf{s}_j)$$

This results in a distribution of proportions $\{q(\mathbf{s}_i), i = 1, \ldots, m\}$ equal to the number of potential edges. For the Arkansas tornado network, this is $m = 292$ with distribution shown in Figure 5.13.

To demonstrate how the LSGM is capturing the feature, consider the mean of the proportion of same neighbors, $\overline{q(\mathbf{s})}$. For the Arkansas tornado network displayed in Figure 5.13 this is $\overline{q(\mathbf{s})} = 0.561$. This average proportion of neighbors assuming the same value as the random variable can be computed for each of the simulated networks from both models. The set of average proportions from each model, $\{\overline{q(\mathbf{s})}^*_{\text{Indep},h}; h = 1, \ldots, 10000\}$ from the independence model and $\{\overline{q(\mathbf{s})}^*_{\text{LSGM},h}; h = 1, \ldots, 10000\}$ from the LSGM can be used as reference distributions to test the significance of the average proportion from the observed network, with p-values computed in a manner similar to Equation 5.14. For a reference distribution obtained from the independence model the p-value is 0.0002. When the reference distribution results from the fit of the LSGM the p-value is 0.7481. From this it can be concluded that independence model is not able to capture the local structure of the network; however, the LSGM, with its neighborhood definition and dependence parameter, is able to recreate this feature.

The previous model assessment techniques indicate that the Arkansas tornado network exhibits a significant amount of local dependence that is appropriately captured by the fit of the LSGM. To see how this fit of the LSGM affects conditional expectations, consider an edge, $y(\mathbf{s}_i)$, with 20 neighbors, $|N_i| = 20$, where each of its neighbors also has 20 neighbors, $|N_j| = 20 \; \forall j \in N_i$. The marginal expectation that this edge will be realized is fixed at $\hat{\kappa} = 0.27$ regardless of the value of the neighbors. However, the conditional probability that this edge will be present, $p_i(N_i)$ is affected by the number of neighbors assuming a positive value. This relationship is plotted in Figure 5.14. When all neighbors of $y(\mathbf{s}_i)$ are absent, the conditional probability that $y(\mathbf{s}_i) = 1$ is only 0.10.

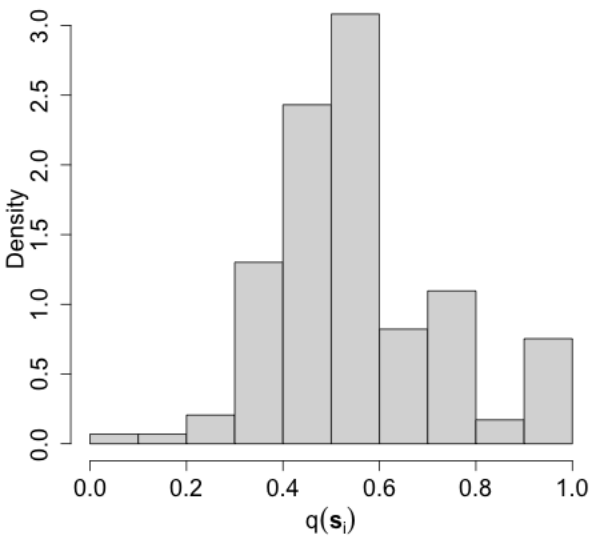


Figure 5.13. Distribution of the proportion of neighbors assuming the same value as the random variable, $p(\mathbf{s}_i)$ for the Arkansas tornado network.

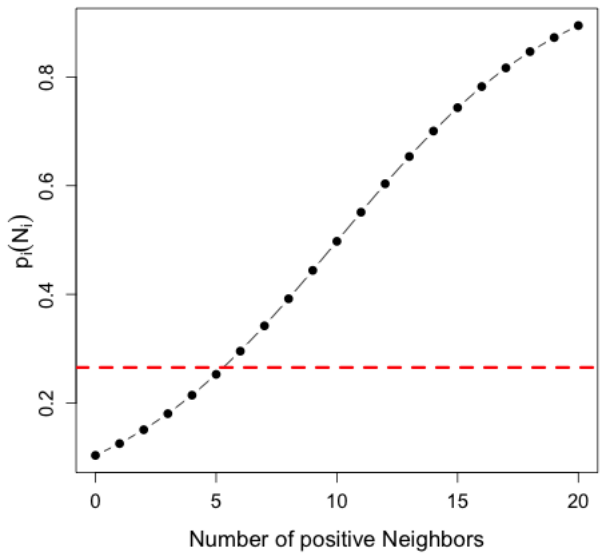


Figure 5.14. Number of positive neighbors against conditional expectation for a random variable with 20 neighbors. The red, dashed, vertical line represents the marginal expectation of $\hat{\kappa} = 0.27$.

This increases monotonically as the number of positive neighbors increases to $p_i(N_i) = 0.89$ when $y(\mathbf{s}_j) = 1 \; \forall j \in N_i$.

5.4 Conclusions

This work has introduced the Local Structure Graph Model, a new model for network analysis. The two main features of the model are an explicit definition of neighborhoods and a specification of the model through full conditional distributions. The LSGM can be interpreted as a MRF with an additional level of modeling or as an alternative approach to specifying an ERGM. The behavior of the model is controlled by two sets of parameters, κ_i which represent the large scale structure and controls the marginal mean of the network, and $\eta_{i,j}$, which captures the local structure and can be interpreted as a dependence parameter. Two optional features were introduced to aid in the specification of the LSGM, decrease computational time, and to lessen the effect of model degeneracy. The optional features are a potentially latent, spatial location of the nodes and a saturated graph, which restricts edges from forming between all pairs of nodes. An application of the LSGM was demonstrated on the network formed by the cited tornadoes in Arkansas during April, 2011. Two simulation-based model assessment techniques indicate the Arkansas tornado network exhibits a significant amount of local dependence that is appropriately captured by the fit of the LSGM.

There are natural extensions that were omitted in this work as to not introduce additional complicating factors. The first extension is the inclusion of node or edge attributes into either the global, local or dependence structure. This can be accomplished through additional modeling of κ, η, or the neighborhood definition, respectively. In addition, the constructed negpotential of the LSGM as currently stated in Equation 5.11 includes an assumption of pair-wise only dependence, a common consideration of the typical application of the MRF model. This assumption implies that cliques of size greater than two are not included in the dependence structure. An extension would be to explicitly account for dependence between triples of random variables by allowing for an additional summation in the expansion of the negpotential function. This will require an extension to the centering approach of [20] as it was developed under the assumption of pair-wise only dependence. The inclusion of attributes and the extension to cliques of size three are the focus of a forthcoming paper.

References

[1] E.M. Agee, J.T. Snow, and P.R. Clare. Multiple vortex features in the tornado cyclone and the occurrence of tornado families. *Monthly Weather Review*, 104(5):552–563, 1976.

[2] William Aiello, Fan Chung, and Linyuan Lu. A Random Graph Model for Power Law Graphs. *Experimental Mathematics*, 10(1):53–66, 2001.

[3] David L Alderson and Lun Li. Diversity of graphs with highly variable connectivity. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 75(4 Pt 2):046102, April 2007.

[4] Barry C. Arnold and S. James Press. Compatible conditional distributions. *Journal of the American Statistical Association*, 84(405):152–156, 1989.

[5] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[6] Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. Scalable k-means++. *Proc. VLDB Endow.*, 5(7):622–633, March 2012.

[7] Sagy Bar, Mira Gonen, and Avishai Wool. A geographic directed preferential internet topology model. *Computer Networks*, 51(14):4174 – 4188, 2007.

[8] Matteo Barigozzi, Giorgio Fagiolo, and Diego Garlaschelli. The Multi-Network of International Trade: A Commodity-Specific Analysis. Working paper, Laboratory of Economics and Management, Sant'Anna School of Advanced Studies, 2010.

[9] Ole E Barndorff-Nielsen. *Information and Exponential Families in Statistical Theory*. Wiley New York, 1978.

[10] Daniel ben Avraham, Alejandro F. Rozenfeld, Reuven Cohen, and Shlomo Havlin. Geographical embedding of scale-free networks. *Physica A: Statistical Mechanics and its Applications*, 330(12):107 – 116, 2003. RANDOMNESS AND COMPLEXITY: Proceedings of the International Workshop in honor of Shlomo Havlin's 60th birthday.

[11] J. W. Berry, B. Hendrickson, R. A. LaViolette, and C. A. Phillips. Tolerating the Community Detection Resolution Limit with Edge Weighting. *Physical Review E*, 83(5), May 2011.

[12] Jonathan W. Berry, Luke K. Fostvedt, Daniel J. Nordman, Cynthia A. Phillips, C. Seshadhri, and Alyson G. Wilson. Why Do Simple Algorithms for Triangle Enumeration Work in the Real World? In *Proceedings of the 5th Innovations in Theoretical Computer Science conference*. ACM, January 2014.

[13] Julian Besag. Spatial Interaction and the Statistical Analysis of Lattice Systems Spatial Interaction and the Statistical Analysis of Lattice Systems. *Journal of the Royal Statistical Society*, 36(2):192–236, 1974.

[14] Julian Besag. Statistical analysis of non-lattice data. *The statistician*, pages 179–195, 1975.

[15] Julian Besag. Contribution to the discussion of Geyer, C.J. and E.A. Thompson, Constrained Monte Carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 657–699, 1992.

[16] Julian Besag. Markov chain monte carlo for statistical inference. *Center for Statistics and the Social Sciences*, 2001.

[17] Shankar Bhamidi, Guy Bresler, and Allan Sly. Mixing time of exponential random graphs. In *IEEE symposium on foundations of computer science*, pages 803–812, 2008.

[18] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[19] Alberto Caimo and Nial Friel. Bayesian inference for exponential random graph models. *Social Networks*, 33(1):41–55, 2011.

[20] Petrua C. Caragea and Mark S. Kaiser. Autologistic models with interpretable parameters. *Journal of Agricultural, Biological, and Environmental Statistics*, 14(3):281–300, September 2009.

[21] Deepayan Chakrabarti and Christos Faloutsos. Graph Mining : Laws , Generators , and Algorithms. *ACM Computing Surveys*, 38(1), 2006.

[22] Peter Clifford. Markov random fields in statistics. *Disorder in physical systems*, pages 19–32, 1990.

[23] N.A.C. Cressie. *Statistics for Spatial Data, revised edition*, volume 928. Wiley, New York, 1993.

[24] Bogdan Denny Czejdo. Network Intrusion Detection and Visualization Using Aggregations in a Cyber Security Data Warehouse. *Int'l J. of Communications, Network and System Sciences*, 05(29):593–602, 2012.

[25] P J Dickinson, M Kraetzl, and W D Wallis. A Graph-Theoretic Approach to Enterprise Network Dynamics - Peter J. Dickinson, Miro Kraetzl, Walter D. Wallis - Google Books. 2007.

[26] Robin Dunbar. Neocortex size as a constraint on group size in primates. *Journal of Human Evolution*, 22(6):469–493, 1992.

[27] M Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 1973.

[28] Stephen E. Fienberg. A brief history of statistical models for network analysis and open challenges. *Journal of Computational and Graphical Statistics*, 21(4):825–839, 2012.

[29] Robert Fisher. The earth mover's distance. `http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/RUBNER/emd.htm`. Accessed: 2014-02-06.

[30] S. Fortunato and M. Barthélemy. Resolution limit in community detection. *PNAS*, 104(1):36–41, 2007.

[31] Ove Frank and David Strauss. Markov Graphs. *Journal of the American Statistical Association*, 81(395):832–842, 1986.

[32] L.C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239, 1979.

[33] Charles J Geyer and Elizabeth A Thompson. Constrained monte carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 657–699, 1992.

[34] Mohammad Ghorbani. Cauchy cluster process. *Metrika*, pages 1–10, 2012.

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

[36] Paramjit S Gill and Tim B Swartz. Bayesian analysis of directed graphs data with applications to social networks. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 53(2):249–260, 2004.

[37] S Gold and A Rangarajan. A graduated assignment algorithm for graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(4):377–388, 1996.

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

[39] Steven M. Goodreau. Advances in Exponential Random Graph (p*) Models Applied to a Large Social Network. *Social networks*, 29(2):231–248, May 2007.

[40] Steven M. Goodreau, Mark S. Handcock, David R. Hunter, Carter T. Butts, and Martina Morris. A statnet tutorial. *Journal of statistical software*, 24(9):1, 2008.

[41] Steven M. Goodreau, James A. Kitts, and Martina Morris. Birds of a Feather, Or Friend of a Friend?: Using Exponential Random Graph Models to Investigate Adolescent Social Networks. *Demography*, 46(1):103–125, 2009.

[42] W R Gray, J A Bogovic, J T Vogelstein, B A Landman, J L Prince, and R J Vogelstein. Magnetic Resonance Connectome Automated Pipeline: An Overview. *IEEE Pulse*, 3(2):42–48.

[43] C. Groendyke, D. Welch, and D.R. Hunter. A network-based analysis of the 1861 hagelloch measles data. *Biometrics*, 2012.

[44] Jonathan L Gross and Jay Yellen. *Graph theory and its applications*. CRC press, 2006.

[45] J. Guo, D. J. Nordman, and A. G. Wilson. Bayesian Nonparametric Models for Community Detection. *Technometrics*, 55(4), May 2013.

[46] Xavier Guyon. *Random fields on a network: modeling, statistics, and applications*. Springer, 1995.

[47] Mark S Handcock. Assessing degeneracy in statistical models of social networks. Technical report, Working paper, 2003.

[48] Mark S Handcock. Statistical models for social networks: Inference and degeneracy. *Dynamic social network modeling and analysis*, 126:229–252, 2003.

[49] Mark S. Handcock and Krista J. Gile. Modeling social networks from sampled data. *The Annals of Applied Statistics*, 4(1):5–25, 2010.

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

[51] Peter D Hoff. Random effects models for network data. In *Dynamic social network modeling and analysis: Workshop summary and papers*, pages 303–312. National Academies Press Washington, DC, 2003.

[52] Peter D Hoff. Bilinear mixed-effects models for dyadic data. *Journal of the american Statistical association*, 100(469):286–295, 2005.

[53] Peter D. Hoff, Adrian E. Raftery, and Mark S. Handcock. Latent Space Approaches to Social Network Analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, December 2002.

[54] Peter D. Hoff and Michael D. Ward. Analyzing dependencies in international relations: commerce, capitalism, conflict, cooperation, and democracy. In *46th Annual Convention of the International Studies Association*, pages 1–20, Honolulu, HI, 2005.

[55] Paul Holland and Samuel Leinhardt. An Exponential family of Probability Distributions for Directed Graphs. *Journal of the American Statistical Association*, 76(373):33–50, 1981.

[56] David R Hunter. Curved exponential family models for social networks. *Social networks*, 29(2):216–230, 2007.

[57] David R. Hunter, Steven M. Goodreau, and Mark S. Handcock. Goodness of fit of social network models. *Journal of the American Statistical Association*, 103(481):248–258, 2008.

[58] David R. Hunter and Mark S. Handcock. Inference in Curved Exponential Family Models for Networks. *Journal of Computational and Graphical Statistics*, 15(3):565–583, September 2006.

[59] David R. Hunter, Mark S. Handcock, Carter T. Butts, Steven M. Goodreau, and Martina Morris. ergm: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of Statistical Software*, 24(3):nihpa54860, 2008.

[60] David R Hunter, Pavel N Krivitsky, and Michael Schweinberger. Computational statistical methods for social network models. *Journal of Computational and Graphical Statistics*, 21(4):856–882, 2012.

[61] Scott Hyde. Properties of the gamma function. `http://jekyll.math.byuh.edu/courses/m321/handouts/gammaproperties.pdf`. Accessed: 2014-01-20.

[62] Mark S. Kaiser, Petrua C. Caragea, and Kyoji Furukawa. Centered parameterizations and dependence limitations in markov random field models. *Journal of Statistical Planning and Inference*, 142(7):1855 – 1863, 2012.

[63] Mark S. Kaiser and Noel Cressie. The construction of multivariate distributions from markov random fields. *Journal of Multivariate Analysis*, 73(2):199–220, 2000.

[64] David Kauchak. Empirical evaluation of dissimilarity measures for color and texture. `http://cseweb.ucsd.edu/classes/fa01/cse291/Dissimilarity.ppt`, 2001. Accessed: 2014-02-06.

[65] E.D. Kolaczyk. *Statistical Analysis of Network Data: Methods and Models*. Springer, 2009.

[66] E.D. Kolaczyk. Tutorial: Statistical analysis of network data. In *2010–11 Program on Complex Networks Opening Tutorials & Workshop*. SAMSI, 2010.

[67] Tamara G. Kolda and Ali Pinar. Feastpack distribution, version 1.1. `http:www.sandia.gov/~tgkolda/feastpack`. Accessed: 2014-02-11.

[68] Johan H Koskinen, Garry L Robins, and Philippa E Pattison. Analysing exponential random graph (p-star) models with missing data using bayesian data augmentation. *Statistical Methodology*, 7(3):366–384, 2010.

[69] Danai Koutra, Joshua T Vogelstein, and Christos Faloutsos. DELTACON: A Principled Massive-Graph Similarity Function. *arXiv.org*, April 2013.

[70] Pavel N Krivitsky, Mark S Handcock, Adrian E Raftery, and Peter D Hoff. Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Social networks*, 31(3):204–213, 2009.

[71] Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. Unit Disk Graph Approximation. In *DIALM-POMC*, pages 17–23, Philadelphis, Pennsylvania, 2004.

[72] A. Lancichinetti, S. Fortunato, and F. Radicchi. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E*, 78:046110, 2008.

[73] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4):1–5, October 2008.

[74] Jaehyung Lee, Mark S. Kaiser, and Noel Cressie. Multiway dependence in exponential family conditional distributions. *Journal of multivariate analysis*, 79(2):171–190, 2001.

[75] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. Kronecker Graphs : An Approach to Modeling Networks. *Journal of Machine Learning Research*, 11:985–1042, 2010.

[76] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution. *ACM Transactions on Knowledge Discovery from Data*, 1(1), March 2007.

[77] E. Levina and P. Bickel. The earth mover's distance is the Mallows distance: some insights from statistics. *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, 2:251–256 vol. 2, 2001.

[78] G Li, M Semerci, B Yener, and M J Zaki. Graph classification via topological and label attributes. *MLG*, August 2011.

[79] Haibin Ling and Kazunori Okada. An efficient Earth Mover's Distance algorithm for robust histogram comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):840–853, June 2006.

[80] S. P Lloyd. Least square quantization in PCM. Technical report, Bell Telephone Laboratories, 1982.

[81] Miranda J. Lubbers and Tom Snijders. A comparison of various approaches to the exponential random graph model: A reanalysis of 102 student networks in school classes. *Social Networks*, 29(4):489–507, October 2007.

[82] David Lusseau. The emergent properties of a dolphin social network. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(Suppl 2):S186–S188, 2003.

[83] O. Macindoe and W. Richards. Graph comparison using fine structure analysis. *Proceedings of the 2010 IEEE Second International Conference on Social Computing*, pages 193–200, 2010.

[84] Daniel Müllner. fastcluster: faster hierarchical, agglomerative clustering routines for r and python. *Journal of Statistical Software*, 53(9), 2013.

[85] Sebastian Neumayer and Eytan Modiano. Network reliability with geographically correlated failures. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[86] M E J Newman, D J Watts, and S H Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(1):2566–72, February 2002.

[87] M.E.J. Newman. Modularity and community structure in networks. *PNAS*, 103(23):8577–8696, 2006.

[88] Krzysztof Nowicki and Tom A B Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001.

[89] F.A. Onat and I. Stojmenovic. Generating random graphs for wireless actuator networks. In *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, pages 1–12, 2007.

[90] Günce Keziban Orman, Vincent Labatut, and Hocine Cherifi. On accuracy of community structure discovery algorithms. *CoRR*, abs/1112.4134, 2011.

[91] L Page, S Brin, R Motwani, and T Winograd. The PageRank Citation Ranking: Bringing Order to the Web. - Stanford InfoLab Publication Server. 1999.

[92] P Papadimitriou and A Dasdan. Web graph similarity for anomaly detection. *Poster*, 2010.

[93] Juyong Park and Mark EJ Newman. Solution of the two-star model of a network. *Physical Review E*, 70(6):066146, 2004.

[94] Juyong Park and MEJ Newman. Solution for the properties of a clustered network. *Physical Review E*, 72(2):026136, 2005.

[95] P. Pattison and G. Robins. Neighborhood–based models for social networks. *Sociological Methodology*, 32(1):301–337, 2002.

[96] Philippa Pattison and Stanley Wasserman. Logit models and logistic regressions for social networks: Ii. multivariate relations. *British Journal of Mathematical and Statistical Psychology*, 52(2):169–193, 1999.

[97] Ofir Pele and Michael Werman. A Linear Time Histogram Metric for Improved SIFT Matching. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision: Part III*. Springer-Verlag, October 2008.

[98] Adrian E Raftery, Xiaoyue Niu, Peter D Hoff, and Ka Yee Yeung. Fast inference for the latent space network model using a case-control approximate likelihood. *Journal of Computational and Graphical Statistics*, 21(4):901–919, 2012.

[99] Claire C. Ralph, Vitus J. Leung, and William McLendon, III. Brief announcement: Subgraph isomorphism on a multithreaded shared memory architecture. In *Proceedinbgs of the 24th ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '12, pages 71–73, New York, NY, USA, 2012. ACM.

[100] Jaideep Ray, Ali Pinar, and C. Seshadhri. Are we there yet? when to stop a markov chain while generating random graphs. In Anthony Bonato and Jeannette Janssen, editors, *Algorithms and Models for the Web Graph*, volume 7323 of *Lecture Notes in Computer Science*, pages 153–164. Springer Berlin Heidelberg, 2012.

[101] W. Richards and N. Wormald. Representing small group evolution,. In *Proceedings of the IEEE Conference on Social Computing*, page 232, 2009.

[102] Alessandro Rinaldo, Stephen E Fienberg, and Yi Zhou. On the geometry of discrete exponential families with application to exponential random graph models. *Electronic Journal of Statistics*, 3:446–484, 2009.

[103] G. Robins, P. Pattison, and P. Elliott. Network models for social influence processes. *Psychometrika*, 66(2):161–189, 2001.

[104] G. Robins, T. Snijders, P. Wang, M. Handcock, and P. Pattison. Recent developments in exponential random graph (p^*) models for social networks. *Social networks*, 29(2):192–215, 2007.

[105] Garry Robins, Philippa Pattison, and Stanley Wasserman. Logit models and logistic regressions for social networks: Iii. valued relations. *Psychometrika*, 64(3):371–394, 1999.

[106] David Robinson. What is the intuition behind the beta distribution? `http://stats.stackexchange.com/questions/47771/what-is-the-intuition-behind-beta-distribution`. Cross Validated Question and Answer Site. Accessed: 2014-01-20.

[107] M. Rocklin and A. Pinar. On Clustering on Graphs with Multiple Edge Types. *Internet Mathematics*, 9(1):82–112, 2013.

[108] Matthew Rocklin and Ali Pinar. Computing an aggregate edge-weight function for clustering graphs with multiple edge types. In Ravi Kumar and Dandapani Sivakumar, editors, *Algorithms and Models for the Web-Graph*, volume 6516 of *Lecture Notes in Computer Science*, pages 25–35. Springer Berlin Heidelberg, 2010.

[109] Matthew Rocklin and Ali Pinar. Latent clustering on graphs with multiple edge types. In Alan Frieze, Paul Horn, and Pawe Praat, editors, *Algorithms and Models for the Web Graph*, volume 6732 of *Lecture Notes in Computer Science*, pages 38–49. Springer Berlin Heidelberg, 2011.

[110] J.P. Rohrer and J.P.G. Sterbenz. Predicting topology survivability using path diversity. *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2011 3rd International Congress on*, pages 1–7, 2011.

[111] Y. Rubner, J Puzicha, C. Tomasi, and JM Buhmann. Empirical evaluation of dissimilarity measures for color and texture. *Computer Vision and Image Understanding*, 84(1):25–43, 2001.

[112] Y. Rubner, C. Tomasi, and L.J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[113] M. Salter-Townshend, A. White, I. Gollini, and T.B. Murphy. Review of statistical network analysis: models, algorithms, and software. *Statistical Analysis and Data Mining*, 2012.

[114] Michael Schweinberger. Instability, sensitivity, and degeneracy of discrete exponential families. *Journal of the American Statistical Association*, 106(496):1361–1370, 2011.

[115] Michael Schweinberger and Mark S Handcock. Hierarchical exponential-family random graph mod-els with local dependence. *Journal of the Royal Statistical Society, Series B*, Under Revision.

[116] Michael Schweinberger, Miruna Petrescu-Prahova, and Duy Quang Vu. Disaster response on september 11, 2001 through the lens of statistical network analysis. Technical report, Technical Report, Department of Statistics, Pennsylvania State University, 2012.

[117] C Seshadhri, Tamara Kolda, and Ali Pinar. Community structure and scale-free collections of erdös-rényi graphs. *Physical Review E*, 85(5), May 2012.

[118] Sean L Simpson, Satoru Hayasaka, and Paul J Laurienti. Exponential random graph modeling for complex brain networks. *PloS one*, 6(5):e20039, 2011.

[119] Tom A. B. Snijders. Markov Chain Monte Carlo Estimation of Exponential Random Graph Models. *Journal of Social Structure*, 3(2), 2002.

[120] Tom A. B. Snijders. Contribution to the discussion of Handcock, M. S., A. E. Raftery, and J. M. Tantrum, Model-based clustering for social networks. *Journal of the Royal Statistical Society*, 170(2):301–354, 2007.

[121] Tom A. B. Snijders, Philippa E. Pattison, Garry L. Robins, and Mark S. Handcock. New Specifications for Exponential Random Graph Models. *Sociological Methodology*, 36:99–153, 2006.

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

[123] Olaf Sporns, Dante R Chialvo, Marcus Kaiser, Claus C Hilgetag, et al. Organization, development and function of complex brain networks. *Trends in cognitive sciences*, 8(9):418–425, 2004.

[124] David Strauss. On a General Class of Models for Interaction. *SIAM Review*, 28(4):513–527, 1986.

[125] David Strauss and Michael Ikeda. Pseudolikelihood Estimation for Social Networks. *Journal of the American Statistical Association*, 85(409):204–212, 1990.

[126] Ana L Teixeira and Andre O Falcao. Noncontiguous Atom Matching Structural Similarity Function. *Journal of Chemical Information and Modeling*, 53(10):2511–2524, October 2013.

[127] Marijtje AJ van Duijn, Tom AB Snijders, and Bonne JH Zijlstra. p2: a random effects model with covariates for directed graphs. *Statistica Neerlandica*, 58(2):234–254, 2004.

[128] Juan C Vivar and David Banks. Models for networks: a cross-disciplinary science. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(1):13–27, 2012.

[129] Yuchung J Wang and George Y Wong. Stochastic Blockmodels for Directed Graphs. *Journal of the American Statistical Association*, 82(397):8–19, 1987.

[130] Stanley Wasserman and Carolyn Anderson. Stochastic a posteriori blockmodels: Construction and assessment. *Social Networks*, 9(1):1–36, 1987.

[131] Stanley Wasserman and Philippa Pattison. Logit models and logistic regressions for social networks:. *Psychometrika*, 61(3):401–425, 1996.

[132] D J Watts and S H Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–2, June 1998.

[133] Richard C Wilson and Ping Zhu. A study of graph spectra for comparing graphs and trees. *Pattern Recognition*, 41(9):2833–2841, September 2008.

[134] George Y Wong. Bayesian models for directed graphs. *Journal of the American Statistical Association*, 82(397):140–148, 1987.

[135] Byung-Jun Yoon, Xiaoning Qian, and Sayed Mohammad Ebrahim Sahraeian. Comparative Analysis of Biological Networks: Hidden Markov model and Markov chain-based approach. *IEEE Signal Processing Magazine*, 29(1):22–34, 2012.

[136] Bonne JH Zijlstra, Marijtje AJ Duijn, and Tom AB Snijders. MCMC estimation for the p2 network regression model with crossed random effects. *British Journal of Mathematical and Statistical Psychology*, 62(1):143–166, 2009.

DISTRIBUTION:

1	MS 0620	Ken Groom, 05642
1	MS 0899	Technical Library, 9536 (electronic copy)
1	MS 0359	D. Chavez, LDRD Office, 1911

Sandia National Laboratories