



HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering
Laboratory of Computer and Information Science

Janne Aukia

Bayesian clustering of huge friendship networks

Master's thesis submitted in partial fulfillment of the requirements for
the degree of Master of Science in Technology

Espoo, 11th September 2007

Supervisor: Professor Samuel Kaski
Instructor: Janne Sinkkonen, Ph.D.

Author	Janne Aukia	Date	11th September 2007
		Pages	viii + 88
Title of thesis	Bayesian clustering of huge friendship networks		
Professorship	Computer and Information Science	Code	T-61
Supervisor	Professor Samuel Kaski		
Instructor	Janne Sinkkonen, Ph.D.		
<p>Because of the recent growth in popularity of social websites, such as MySpace, Facebook and Last.fm, there is an increasing interest in ways to analyze extremely large friendship networks with even millions of nodes. These huge networks provide a practical test ground for new network algorithms. The network analysis methods can also be applied to other networks than social networks, such as interactions between proteins and links between web pages.</p> <p>Social networks have typically structure: there are dense groups of nodes and some nodes have disproportionately many connections. The structure emerges, because friendships are not formed randomly. Instead, people tend to become friends with those who are similar to themselves. This can be called homophily. There are also other factors that guide the formation of friendships, such as geographical location and membership in common activities.</p> <p>The M0 algorithm finds clustering structure in networks with homophily by Bayesian statistical inference. The algorithm is based on a generative model for creating the edges of a network based on latent components. The model parameters are inferred using Gibbs sampling. Because of homophily, the nodes that belong to the same cluster are likely to have similar traits.</p> <p>In this master's thesis, an effective implementation of the M0 algorithm is introduced, which uses a balanced binary tree for storing the component probabilities. The implementation can be used on networks with even millions of nodes. The algorithm is tested on a range of well studied small networks and on a friendship network with over 600 000 users crawled from the Last.fm service.</p> <p>The algorithm finds meaningful structures in networks of various scales and the results are comparable to those obtained with hierarchical clustering methods. The strength of the method is the fuzzy assignment of nodes to clusters, where a node can belong to a number of clusters simultaneously. However, the choice of model hyperparameters is often inconvenient.</p>			
Keywords	clustering, friendship network, Bayesian inference, latent variable model		

Tekijä	Janne Aukia	Päiväys	11. syyskuuta 2007
		Sivumäärä	viii + 88
Työn nimi	Bayesilaisen klusteroinnin soveltaminen erittäin suuriin ystävyysverkkoihin		
Professori	Informaatiotekniikka	Koodi	T-61
Työn valvoja	Professori Samuel Kaski		
Työn ohjaaja	FT Janne Sinkkonen		
<p>Sosiaalisten verkkopalveluiden, joita ovat esimerkiksi MySpace, Facebook ja Last.fm, vii-meaikaisen suosion kasvun myötä kiinnostus erittäin suurten ystävyysverkostojen analysointiin on kasvanut. Näissä verkoissa on jopa miljoonia solmuja, joten ne tarjoavat hyvän testiympäristön uusille verkkoalgoritmeille. Verkkojen analysointimenetelmiä voidaan hyödyntää myös muihin kuin sosiaalisiin verkkoihin, kuten proteiinien välisiin vuorovaikutusverkkoihin ja verkkosivujen välisiin linkkeihin.</p> <p>Sosiaalisilla verkostoilla on tyypillisesti rakenne: niissä on tiheitä solmuryhmittymiä, ja joillakin solmuilla on suhteettoman paljon yhteyksiä. Rakenne syntyy, koska ystävydet eivät muodostu satunnaisesti. Ihmiset sen sijaan tapaavat ystävyystyä samanlaisten ihmisten kanssa. Tätä voi kutsua homofiliaksi. Ystävyksien syntyyn vaikuttavat myös muut tekijät, kuten maantieteellinen sijainti ja yhteisiin aktiviteetteihin osallistuminen.</p> <p>M0-algoritmi löytää klusterirakenteen homofiilisista verkoista bayesilaisen tilastollisen inferenssin avulla. Algoritmi pohjautuu generatiiviseen malliin, jossa verkon sivut luodaan latenttien komponenttien perusteella. Mallin parametrien tilastollisessa päättelyssä käytetään Gibbs-otantaa. Homofilian vuoksi samaan klusteriin kuuluvilla solmuilla on todennäköisesti yhteisiä piirteitä.</p> <p>Tässä diplomityössä esitetään M0-algoritmille tehokas toteutus, joka käyttää tasapainotettua binääripuuta komponenttien todennäköisyyksien tallennukseen. Toteutus toimii jopa miljoonien solmujen verkoilla. Algoritmia testataan joukolla aiemmin tutkittuja pieniä verkkoja ja Last.fm-palvelusta kerätyllä ystävyysverkolla, jossa on yli 600 000 käyttäjää.</p> <p>Algoritmi löytää merkityksellisiä rakenteita monenkokoisista verkoista, ja tulokset ovat vertailukelpoisia hierarkisilla klusterointimenetelmillä saatujen tulosten kanssa. Menetelmän vahvuus on solmujen sumea klusterointi, jossa solmu voi kuulua samanaikaisesti useaan klusteriin. Hyperparametrien valinta on kuitenkin usein hankalaa.</p>			
Avainsanat	klusterointi, ystävyysverkko, bayesilainen päättely, latenttimuuttujamalli		

Acknowledgements

This Master's thesis was conducted at Xtract Ltd.

I would like to express my gratitude to my instructor Doctor Janne Sinkkonen and to my supervisor Professor Samuel Kaski. They have given helpful advice and comments during various phases of the work. Additionally, I want to thank Norman Casagrande from Last.fm for helpful answers about the data set used in the thesis, and Last.fm for providing access to the data set.

Special thanks go to Xtract Ltd for making it possible to spend time and effort on this thesis, and to all the people there for a working environment where something fun or interesting is always happening.

Finally, I would like to thank my fiancée Anni for exploring the world together with me, and for love and support during the long evenings spent writing the thesis.

Helsinki, 11th September 2007

Janne Aukia

Contents

1	Introduction	1
1.1	Background	1
1.1.1	Concepts of interest	1
1.1.2	Aims of the research	2
1.2	Problem setting	3
1.3	Contributions of the thesis	4
1.4	Structure of the thesis	4
2	Friendship and network structure	5
2.1	Common properties of networks	5
2.2	Analyzing social networks	9
2.3	Friendship networks	10
2.4	Formation of friendships	11
2.4.1	Challenges in defining friendships	11
2.4.2	Focus theory	12
2.4.3	Homophily and interests	12
2.4.4	Theories of proximity	15
2.4.5	Relationships between the theories	16
3	Data analysis by clustering	17
3.1	Data analysis	17
3.2	Clustering	18
3.3	Clustering of network elements	19
3.3.1	Clustering coefficient	20
3.3.2	Clustering quality	20
3.4	Algorithms for network clustering	22
3.4.1	Deterministic methods	22
3.4.2	Probabilistic methods	24
3.4.3	Conclusions on the clustering algorithms	25
4	Bayesian inference	26
4.1	Bayesian probability theory	26
4.2	Hierarchical generative models	27
4.3	Parameter inference	28

4.3.1	Sampling	29
4.3.2	The EM algorithm	30
4.4	Challenges with sampling	30
4.4.1	Convergence of sampling	30
4.4.2	Label switching	30
4.5	Distributions	31
5	A latent component model for networks	34
5.1	Model introduction	34
5.2	The finite mixture model	35
5.2.1	Joint distribution	36
5.2.2	Conditional link probabilities	37
5.3	The infinite mixture model	39
5.3.1	Joint distribution	40
5.3.2	Conditional link probabilities	40
5.3.3	Marginal likelihood	42
5.4	Hyperparameter values	43
5.5	Implementation of the algorithm	43
5.5.1	Simple implementation	43
5.5.2	Hash table implementation	44
5.5.3	Binary tree implementation	46
6	Experimental setup	49
6.1	Material	50
6.1.1	Small test networks	50
6.1.2	Last.fm data set	51
6.2	Methods	55
6.2.1	Algorithm implementation and analysis tools	55
6.2.2	Assessing algorithm convergence	55
6.2.3	Finding optimal hyperparameters	55
6.2.4	Analyzing small networks	56
6.2.5	Clustering Last.fm friendship network	56
7	Results	57
7.1	Finding optimal hyperparameter values	57
7.2	Clustering small networks	59
7.2.1	Networks with known community structure	61
7.2.2	Convergence of the small networks	63
7.3	Clustering Last.fm friendship network	65
7.3.1	Full Last.fm network	65
7.3.2	Last.fm USA network	66
7.3.3	Likely and unlikely tags	68
7.3.4	Convergence of the clustering for Last.fm networks	70
7.3.5	Close-up view of the Last.fm Denmark network	70

CONTENTS	iv
8 Conclusions and discussion	73
8.1 Evaluation	73
8.2 Applicability of the results	75
8.3 Future work	76
8.3.1 Validation	76
8.3.2 Improvement	76
8.3.3 Performance optimization	77
8.3.4 Further possibilities	77
Bibliography	77

Abbreviations

CMC	Computer Mediated Communication
CNM	Community algorithm by Clauset, Newman and Moore
CONCOR	Convergence of iterated correlations
CPU	Central Processing Unit
DP	Dirichlet Process
EM	Expectation Maximization
EO	Extremal Optimization
GN	Community algorithm by Girvan and Newman
IT	Information Technology
LDA	Latent Dirichlet Allocation
M0	The latent component model presented in this paper
MCMC	Markov Chain Monte Carlo
mPCA	Multinomial Principal Component Analysis
N06	The spectral community algorithm by Newman
NF	Newman's Fast community algorithm
NP	Nondeterministic Polynomial-time
SNA	Social Network Analysis

Notation

x	Real number or a real-valued vector
$p(x)$	Probability mass or density of x
$p(x y)$	Probability of x given y
$p(x, y)$	Joint probability of x and y
$ x $	Element count in vector x
$x \propto y$	x is proportional to y
$x \sim y$	x is asymptotically equal to y
$x \equiv y$	x is equivalent to y
$x^{(t)}$	The value of x in sample t
$Dir(x)$	Dirichlet distribution with a symmetric parameter x
$Mult(\cdot)$	Multinomial distribution
$DP(\cdot)$	Dirichlet Process
$\Gamma(\cdot)$	Gamma function
$\mathcal{O}(\cdot)$	Asymptotic complexity of an algorithm
$x^{[n]}$	Pochhammer symbol
Q	Modularity of a network
c	Clustering coefficient
d	Mean degree of a network
m_z	Probability distribution over nodes for component z
θ	Probability distribution over components
α	Hyperparameter for the distribution over components
β	Hyperparameter for the distribution over nodes in components
L	Set of edges
Z	Set of components for edges
N	Number of nodes
E	Number of edges
n_z	Number of edges with the component z
k_{zi}	Number of edges adjacent to node i with the component z
v_i, v_j	First and second end point

List of Figures

2.1	A sketch of some of the different disciplines studying networks. . . .	6
2.2	Illustrations of an undirected network, a directed network and a weighted undirected network, each with 7 nodes and 15 edges.	7
3.1	Example of a computer generated network with local structure.	19
4.1	Relationship between statistical inference and generative processes. . .	28
4.2	Dirichlet distribution for three dimensions with different parameter values.	33
5.1	A demonstration of the principles behind the M0 model.	35
5.2	Plate model representation of the M0 generative process.	37
5.3	Updating of the partial sum tree.	47
6.1	Last.fm users in different countries and the most common tags for all users in Last.fm.	54
6.2	Last.fm friendship degrees have a heavy tail while the age distribution peaks at around 21 years.	54
7.1	An artificial doughnut-shaped network, which demonstrates the effects of different hyperparameters.	57
7.2	Results from running the clustering with a range of hyperparameter values.	60
7.3	Two visualizations of the clustering result for the Karate network. . .	62
7.4	Clustering result for the Football network.	63
7.5	Convergence of the small networks.	64
7.6	The whole Last.fm main component clustered into five clusters and plotted together with the tag counts in each group.	66
7.7	The whole Last.fm main component clustered into five groups with the countries of the group members.	67
7.8	The Last.fm users from United States belonging to the main component clustered into eight groups with their tags.	68
7.9	Convergence of the full Last.fm network and of the United States subset of Last.fm network.	71
7.10	A close-up view of the results from clustering the Danish Last.fm users.	72

List of Tables

6.1	Small networks used in testing the algorithm.	50
6.2	Networks crawled from Last.fm used to test the algorithm.	53
7.1	Optimal hyperparameter values for the small networks in terms of modularity.	58
7.2	Network modularity with different algorithms.	59
7.3	The most likely and unlikely tags for each of the components in the Last.fm United States network.	69

Chapter 1

Introduction

This chapter motivates the need for creating new methods for analyzing social networks and presents the aims of the research in this context. These aims are formalized in the form of research questions. Finally, the contributions of the thesis and the structure of the forthcoming chapters are detailed.

1.1 Background

The subject of this thesis is interdisciplinary. This is why the aim is to combine information and methods from multiple scientific disciplines, with the hope that a broader picture of the problems at hand will emerge. Furthermore, this approach is interesting for a researcher, because seemingly unrelated subjects studied by physicists, sociologists and computer scientists intertwine naturally in network research.

1.1.1 Concepts of interest

Traditionally, sociologists have analyzed social networks by interviewing people about their social relationships or observing the social behavior of a group of people. The typical size of networks studied has been tens or hundreds of persons (Newman, 2003a). However, in recent years two developments have made it possible to study the social interaction of much larger numbers of people than before: The emergence of new communication channels, such as mobile phones and online services, and the availability of IT infrastructure for storing and analyzing large amounts of data.

Working with large networks is computationally hard. In the past, good algorithms have been created for analyzing small social networks with only tens or hundreds of nodes, but using them on larger networks is infeasible. The current challenge is to create algorithms that can be used on networks with millions of nodes, are fast and are able to mine meaningful information.

Methods for analyzing networks are not limited to only the study of social interactions. Many different systems can be represented as network structures, such as metabolic networks in biology, or optimization of large infrastructures in technical problems (Newman, 2003a). Concepts and algorithms that have been found to be useful in studying one type of networks are often applicable to other kinds of networks as well.

Social networking services, such as MySpace and Facebook, have gained huge popularity in recent years. Users of these services can create profiles for themselves and list their friends for other users to see and browse. Based on these *friends lists*, a network can be created, where nodes are persons and edges friendships between them. These *friendship networks* provide a good test ground for network algorithms, because they make it possible to compare the network topology with personal attributes, such as demographics and interests, on a much larger scale than what has been possible before.

1.1.2 Aims of the research

In data mining, the amounts of data are often so large that it is difficult to see any relevant content by just looking at them. Unsupervised learning methods try to make large amounts of data more easily analyzable by reducing the complexity or by finding a smaller number of dimensions that represent the relevant structures. Clustering is an unsupervised method, where the aim is to group similar elements together. The results of clustering can be used for visualization and further analysis.

One property affecting the formation of clusters in networks is that individuals sharing some common traits, tend to link to each other. This tendency of individuals to associate with similar others is called *homophily*. Homophily in many different forms has been observed in a large number of social networks (McPherson et al., 2001). Moreover, homophily leads to the formation of communities in friendship networks that tend to consist of people who share interests.

This master's thesis presents an algorithm called M0 for finding clusters and traits in a network with homophilic structure using Bayesian statistical inference. The main idea behind the algorithm is that there are a number of mutually exclusive traits, which explain the formation of edges in the network. Each node in the network has its own proportions of the traits, that is, a node may have more than one trait. The model behind the algorithm assumes that each edge in the network represents one trait. Thus, the more two nodes have traits in common, the more likely they have an edge between them.

Several methods have been developed for network clustering. However, many of the popular approaches do not take noise in the data into account, or it is difficult to understand the assumptions behind the models. Because of the generative foundation of the M0 algorithm, it can relatively easily be made to handle noisy and missing data. Moreover, instead of giving just one fixed clustering of the network, Bayesian estimation of the model gives probabilities for the different values of the clustering parameters, and an explanation of the process that could have generated the structures in the network.

Despite the advantages, the generative framework has some shortcomings. Notably, the models often include parameters that are difficult to find values for, the methods require an understanding of Bayesian estimation, and it is often difficult to make efficient implementations of the methods.

In this thesis, the aim is to discuss the processes that guide the formation of social contacts between individuals and the ways traits affect social structures. The M0 algorithm is presented and discussed from this perspective. Finally, the algorithm is tested on a range of networks, including a friendship network from Last.fm social network service. Specifically, it is evaluated how well the algorithm is able to reveal traits from the networks.

1.2 Problem setting

The research problem is stated as follows:

By using only friendship network topology, can individuals be clustered into groups that differ in traits such as interest, language and gender?

To address this problem, research subquestions are specified:

1. Based on previous research, is it plausible that friendships can be used to predict traits of the individuals?
2. Can the M0 method be used in finding clusters from a friendship network?
3. Based on the clustering, is it possible to make some conclusions about the traits affecting the friendships of the individuals?
4. Does the algorithm find local clusters in a network (communities) or more diffuse latent component structures (traits)?
5. How well does the method compare with other approaches in terms of quality of the clustering results and algorithm speed?

1.3 Contributions of the thesis

This thesis presents an optimized implementation of the M0 clustering algorithm, which can be used even with huge networks. The algorithm was conceptually developed by Janne Sinkkonen (Sinkkonen et al., 2007). The author was responsible for the empirical work related to the implementation, optimization and testing of the algorithm, as documented in this thesis. The algorithm is compared to sociological models, and testing of the algorithm is carried out both on small networks and on a large friendship network.

1.4 Structure of the thesis

After this introductory chapter, Chapter 2 starts with explanation of the basic concepts of networks to give a common ground for discussing networks in general, and social networks in particular. Friendship networks are introduced, as well as theories about friendship from sociology.

Chapter 3 deals with data analysis and clustering and introduces approaches for network clustering. In chapter 4 the concepts of generative models and Bayesian parameter inference for the models are presented, and some background understanding on typical distributions and sampling challenges is given to support the understanding of the algorithm.

The algorithm and its implementation are presented in Chapter 5. First, two different methods for estimating the model parameters are introduced: one for a fixed number of parameters, and another that automatically adjusts to the number of parameters. Next, three different algorithms based on the model are given, and the efficiency of the algorithms is discussed.

Chapter 6 presents the experimental setup for testing the algorithm and Chapter 7 details the results of the experiments. Finally, in Chapter 8, conclusions are drawn based on the experiments, and directions for future research are given.

Chapter 2

Friendship and network structure

This chapter covers the basic concepts related to networks and methods for analyzing them. Friendship networks are introduced and the formation of friendships and the effect of similarity between individuals in social networks is discussed in detail. An understanding of friendship networks is necessary to be able to create algorithms, which model meaningfully these real-world structures, and to evaluate their effectiveness. Luckily, research done in sociology for already decades can be used to develop this understanding.

2.1 Common properties of networks

In recent years, there has been a strong interest in studying networks, ranging from sociology and epidemiology to applied physics and the study of biological networks. Because of the varying backgrounds of the researchers and different scientific communities, information flow between the disciplines has been slow. To give an idea of the wide range of approaches used, a sketch of the different disciplines working with networks is presented in Figure 2.1.

The basis of network research is formed of *graph theory*, which originates from the famous paper by Euler (1736) about the bridges in Königsberg and how they connect the different parts of the city (Newman, 2003a). Sociologists have studied social networks from the beginning of the 1930's. They have made significant contributions to the statistical methods for analyzing networks and to the empirical analysis of social networks.

In the last ten years, applied mathematicians and physicists have joined in the network research, trying to develop a general theory of networks. This recent school of research is often referred to as *complex networks research*, and it is concerned with the similarities and differences of the various types of networks (Costa et al., 2007).

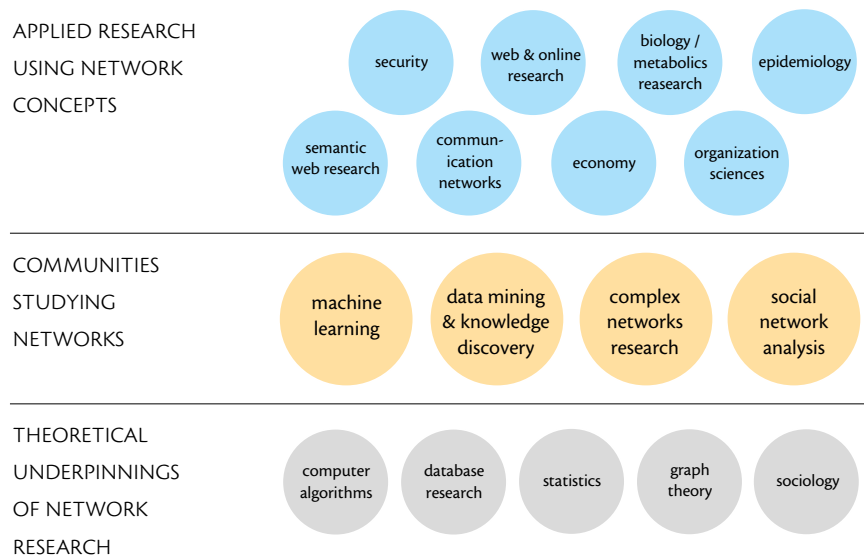


Figure 2.1. *A sketch of some of the different disciplines studying networks. In the bottom are some of the fields that provide the theoretical background that the network sciences build upon. In the middle are the scientific communities doing methodological studies, and in the top the sciences that utilize the methods created by the network communities. Of course, in practice, the distinctions between the different disciplines are neither strict nor well specified.*

It is useful to define some basic concepts about networks to have a clear vocabulary for dealing with the intuitive concepts. Formally, a network is a set of pairwise connected entities. In graph theory, these items are called *vertices*, and the connections between vertices are called *edges*. Together the sets of vertices and edges form a *graph*.

The number of edges connected to a node is called the *degree* of the node. A network with an edge between every node is called *fully connected*, and for a network with N nodes the number of edges $E = N(N - 1)/2$. Real-world networks are typically not densely connected, but instead sparse, that is, it is very unlikely that there is an edge between two nodes selected by chance. In sparse networks $N \sim E$ (Clauset et al., 2004).

Because of networks have been studied by many sciences, naming of even the most basic concepts is far from clear. Vertices are often called *nodes* in computer science, while in social sciences they are typically referred to as *actors*. Edges are sometimes called *links* in computer science, *bonds* in physics and *ties* in social science. This thesis uses the terms “node” and “edge” for the sake of consistency.

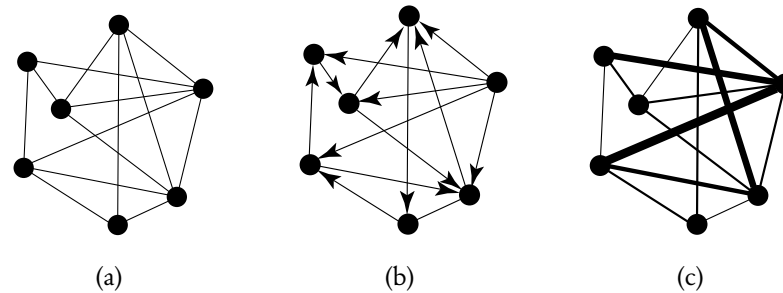


Figure 2.2. Illustrations of an undirected network (a), a directed network (b) and a weighted undirected network (c), each with 7 nodes and 15 edges. Illustration adapted from Boccaletti et al. (2006).

Any process where individual elements interact can be pictured as a network. Some typical classifications of networks can be seen in Figure 2.2. The edges of a network can be *directed* or *undirected* (Newman, 2003a). In the Figure 2.2(b) showing the directed network, arrows indicate the direction of the link. In directed networks, there is a distinction between directed (one-way) and mutual (undirected, two-way) connections between nodes. Some processes are more easily represented with directed edges, for example hyperlinks between web pages, while others, such as roads connecting cities, are naturally represented as undirected connections.

Another typical way to enrich a network model is to add weights to network edges to describe the strength of the connection. For example, in social networks weights may represent the amount of communication between persons. In the weighted network of Figure 2.2(c), the line thickness corresponds to the edge weight.

In addition to adding the link directions and weights, the simple model for networks can be extended in several ways. For example, a network could contain different types of edges for describing different kinds of relationships between nodes (Newman, 2003a). In social networks, different relationships could be for example *friendship*, *kinship*, *information transfer* or *emotional support*. Furthermore, many types of attributes can be associated to the nodes.

Many models of networks consider the network as unchanging or at least as a static snapshot of a process at a certain point of time. However, dynamic networks with new edges and nodes being added and old ones fading away have also been studied (Berger-Wolf and Saia, 2006, Kempe et al., 2000, Kleinberg, 2003, Leskovec et al., 2005) and recently even rich models for network growth have been proposed (Qamra et al., 2006, Zheng and Goldenberg, 2006). Studying networks that change in time leads to additional challenges for data analysis because of the larger amounts of data and the added complexity.

Often, when a network starts to form, it first consists of many isolated components not connected to each other. However, after a while most of these components connect forming a *giant component*, which is the connected subgraph that contains the majority of all the nodes. A network is said to *percolate* when a giant component forms. The mathematical *percolation theory* deals with the formation of connected subgraphs in random networks (Albert and Barabasi, 2002).

The potential *scale-free* structure of networks has received much attention (Holme, 1994, Newman, 2003a). Networks are said to be scale-free if their degree distribution follows the power-law, that is $p_k \sim k^{-\alpha}$ for some constant exponent α , where p_k is the fraction of nodes in the network that have degree k (Newman, 2003a). The name scale-free comes from the fact that the power-law distribution has no natural scale (Dorogovtsev and Mendes, 2003, page 12), i.e., the shape of the distribution is the same in all scales. A degree distribution following the power-law is linear when plotted on a log-log scale.

A weaker statement than to say that a network follows the power-law is to state that a network has a *heavy tail* (i.e., a fat tail), which roughly means that there are some nodes with many connections and many nodes with just a few connections. Even though many networks are said to have a scale-free structure, in reality, they would fit also other distributions having a heavy tail (Clauset et al., 2007).

In the classical *small-world experiments* by Stanley Milgram in the 1960's, randomly selected individuals were asked to make acquaintance chains to each other (Travers and Milgram, 1969). In the experiments, it was noted that the mean distance between individuals was less than six. Watts and Strogatz (1998) noted that short average connections between people form in networks with high amounts of dense subgraphs (or large *clustering coefficient*, covered in more detail in Section 3.3.1) and where the shortest distances between nodes are short on average. They defined these networks as *small-world networks*. Networks which are scale-free are also small-world. However, even networks that do not follow power-law can be small-world, as long as they fulfill the two requirements mentioned above (Amaral et al., 2000).

Several models have been created to explain what types of underlying processes affect the emergence of networks and their statistical properties. A simple but useful model is the *random graph* studied by Rapoport (1957), and Erdős and Rényi (1959). In this model, undirected edges are created randomly between a fixed number of nodes. However, this model does not explain the heavy tails observed for many different types of networks. An improvement to this is the *preferential attachment* model proposed by Barabási and Albert (1999) where edges are more likely to be added to nodes that already have many edges. This leads to a scale-free node degree distribu-

tion. Newman (2003a) provides a good overview of the many other models that have been generated to explain different aspects of network emergence.

2.2 Analyzing social networks

A *social network* is a representation of social relationships such as friendship, kinship, or information exchange between a set of individuals. The term social network is said to originate from Barnes in 1954 (Wasserman and Faust, 1994). In the simplest form, a social network is a static picture of the social connections between individuals at a certain point of time. In sociology, the field of studying social networks is called *social network analysis* (SNA).

The aim of social network analysis is to make a model of the social interactions between individuals, and then study how this structure affects the functioning of the individuals and groups in the network (Wasserman and Faust, 1994). To collect the data on a social network typically requires sending questionnaires to people, asking them to describe their relationships, or observing their behavior (Newman, 2003a).

Traditionally, sociologists have studied only small and well-bounded networks, with largest networks typically having less than one hundred nodes and almost never more than a thousand (Newman et al., 2006). Nowadays, by examining mobile or online communication, huge social networks can be analyzed with thousands or even millions of nodes.

A problem with data that is collected automatically from communication between individuals is that it is often in a raw form and it may have lots of errors and other deficiencies. A typical challenge is that the data sets describe communication only via one communication channel, such as email or a certain online forum. This selective sampling can make the data biased, and observations made based on one communication channel might not be generalizable.

While social networks study is currently one of the most active fields in network analysis, many different kinds of networks can be analyzed using similar methods. These include biological networks, such as the interaction between proteins, communication networks, such as networks of computers connected to the Internet, and networks in economics, for example for analyzing trade between countries.

There are a number of software applications that can be used in social network analysis. Popular applications include *Pajek*, *UCINET*, and *NetMiner* (Huisman and van Duijn, 2005). With them, measures such as centrality and clustering can be computed. However, researchers who wish to create new metrics need to implement applications

of their own. This is why a large number of applications intended mainly for the personal use of a small number of researchers exist.

2.3 Friendship networks

People are good at forming friendships. Typically as soon as individuals are gathered together, for example in a party or at a workplace, they start interacting with others. Based on these interactions, friendly relationships and friendships start to form. These relationships can be pictured as a friendship network where nodes (individuals) are connected if they are friends with each other.

Typically sociologists and ethnographers have analyzed friendship networks by interviewing people and asking, whom they consider to be their friends. Nowadays, in addition to traditional friendship research methods, it is also possible to study friendships in online communities based on automatically collected data. People communicate with each other via different types of channels, such as email, chat rooms, and message boards. They also participate in online events, and join in different communities.

Together, the different methods for online communication are covered by the term *computer mediated communication* (CMC), used by researchers from fields such as linguistic studies, anthropology and communication studies (Herring, 2002). The online communication between users provides huge amounts of friendship data for researchers to analyze.

The friendship networks people have in the physical world and in online communities are not the same (Boyd, 2006). Some connections between people are present in both the online and the physical world, while others are only online or *in real life*. There are lots of people who do not want to create explicit friendship networks online. Moreover, an online friendship is often a much weaker connection between two persons than a friendship in the physical world. For example, some users collect huge amounts of online “friendships”, but do not know any of the people personally. Thus, based on online friendships, one should not make too hasty conclusions on the social relations between the users.

A practical method for analyzing relationships in online communities is to observe how people communicate via different types of channels (Haythornthwaite and Wellman, 1998, for example). Those who communicate a lot with each other typically have a close social connection. Thus, this communication information can be used as an indicator of the friendships between the persons.

Another approach for studying online contacts between people is to have the users of an online service themselves list, who they consider to be close acquaintances. Luckily, this type of information can be easily gathered from social networking sites, such as MySpace, Friendster, and Orkut. In these online services, an important function is that users can list, who are their “friends”, and ask other users to form friendships with them. Based on these lists a network of friendships between the users can be formed.

The best approach is to combine multiple data sources. In addition to the two aforementioned approaches, there are several other methods to infer relationships between users, such as studying links between web pages of users (Park, 2003) and observing memberships of users in online communities or their participation in discussions (Paccagnella, 1998). Also traditional methods from social network analysis, such as questionnaires and interviews of users, can be combined to the other approaches to obtain a better picture of the relationships between the individuals.

2.4 Formation of friendships

Having a good grasp of how networks of friendships evolve requires an understanding on the processes that give rise to friendships. Below, first the concept of friendship is covered briefly and next the processes that govern the formation of friendships are analyzed from three perspectives: using *focus theory* (Feld, 1981), via the concept of *homophily* (Lazarsfeld and Merton, 1954), and by discussing the effect of physical proximity.

2.4.1 Challenges in defining friendships

Even though everybody has an idea of what friendship is, it is rather difficult to give a precise definition. This is because the concept of friendship is ambiguous and vague with many different overlapping meanings (Van De Bunt et al., 1999). In addition, friendship is conceived in a multitude of ways in different societies (Keller, 2004).

The formation of friendships is a complex process. Friendships change over time, as old friendships weaken or die while new relationships emerge. Individuals have the chance to affect their friendships by investing different amounts of effort into them. Friendships may form via a number of social processes, such as when a mutual acquaintance introduces two persons to each other, i.e., via *transitive linking* (Ebel et al., 2003). Furthermore, people may have different levels of friendship with different people (Van De Bunt et al., 1999).

2.4.2 Focus theory

According to Feld (1981), a problem with social network analysis is that patterns are extracted from social tie structures without taking into account the social structures, in which people operate. Focus theory proposed by Feld (1981) aims to provide a theoretical model of how people form friendships. It is based on the idea that people take part in a number of foci, which can be “*social, psychological, legal or physical entities.*” Some typical foci given as examples are family, workplace and voluntary organizations. The study of social networks should take into account the foci in which people participate.

A focus is *constraining* because it leads individuals, who belong to it, to spend time and energy in participating in the activities of the focus (Feld, 1981). This makes possible the formation of friendships, because two individuals that share a focus are more likely to interact with each other than two random people. According to the theory, the more the people interact, the more likely it is that they will develop positive feelings towards each other.

Some foci involve more constraint while others constrain their members less. If there is no constraint, the focus does not exist. For some foci, such as families, the members of the focus are forced to interact much and often. Other foci require less interaction from their members, for example city neighborhoods. The more constraining a focus is, the more likely it is that two individuals, who belong to the focus, are connected to each other.

The model explains the formation of clusters in networks. Clusters tend to form around different foci, in which the individuals participate. Feld (1981) states that clusters with few members tend to be strong while clusters with many members are often loose in structure. However, the opposite is also possible (Feld, 1981).

Feld (1981) emphasizes that although foci tend to produce structures in social interactions, not all structures arise from foci, but people may also meet by chance. However, interaction formed around foci has structural significance.

2.4.3 Homophily and interests

People tend to communicate with those who are in some way like them. This tendency to interact with similar people is called homophily (Lazarsfeld and Merton, 1954) and is well expressed with the saying “Birds of a feather flock together.” The opposite of homophily is *heterophily*, which occurs, when people with different attributes are likely to be connected to each other. An example of this would be a dating network, where

most edges would be between males and females and the network would exhibit strong heterophily by gender.

The forming of homophilic relationships is one of the first properties analyzed by the early social network researchers (Macskassy and Provost, 2004). The original definition on homophily by Lazarsfeld and Merton (1954) made a distinction between homophily by status and by value. *Status homophily* means that individuals with similar status in the social hierarchy tend to associate with each other. *Value homophily* means that people, who are intrinsically similar in some way, have a tendency to associate with each other.

According to McPherson et al. (2001), the effect of homophily is additive in social interactions. That is, people who are similar to each other in more than one factor will have an greater probability of being connected.

The concept of *assortative mixing* in networks is closely related to homophily. Assortative mixing is the tendency for nodes in networks to be connected to other nodes that are like them in some way (Newman, 2003b). A special case of assortative mixing is mixing by node degree. This means that nodes with lots of neighbors tend to be connected to others with many connections. Confusingly, often the term assortative mixing is used specifically when talking about mixing by node degree.

Newman (2003b) has found that social networks tend to exhibit assortative mixing by degree, while communication networks and biological networks tend to have disassortative (i.e., dissortative) mixing by degree, which means that nodes with many connections tend to link to nodes with only a few.

One explanation to homophily is given in the field of social psychology by the *similarity effect* (Byrne, 1971). It refers to the tendency of people, who have similar attitudes, to be attracted towards each other. In other words, there is a causal relation between similarity and attraction. Byrne (1971) has studied this causal relation in controlled experiments and has found out that the amount of attraction between two persons is a positive linear function of the proportion of personality characteristics they share. That is, the more attitudes two persons share, the stronger the attraction.

Another explanation to the formation of homophilic relationships is the *self-categorization theory* by Turner et al. (1987), which states that people categorize themselves and prefer to interact with those who belong to the same category as themselves. Moreover, communication is easiest with similar people because they have a smaller risk of misunderstanding each other (Ibarra, 1992).

A third process leading to homophily in social relationships is that people who have a social relationship, have an effect on each other. Thus, the interests and attitudes of

one person may affect those of the other and this may lead to the two persons sharing interests and attitudes. In a way, this is the opposite of the similarity effect, because the causal relationship is reversed so that the relationship between two individuals leads to similar attitudes.

Homophily not only connects similar people but at the same time separates different people (Yuan and Gay, 2006). Newman (2003b) notes that strong homophily tends to break a network into separate communities. For example, in a social network with people speaking different languages, the network might break up into communities speaking the same language, with only those speaking multiple languages connecting the communities.

Empirically homophily has been evidenced in a large number of social networks based on attributes such as age, gender, and status. According to McPherson et al. (2001), in empirical studies race has been found to be the strongest factor in social relationships. After race, the strongest factors are age, religion, education, profession, and gender.

Homophily can be also seen in the online world. Studies of an online dating system showed that, just like in the offline world, the users of the system sought much more often people that are similar to them than what chance would predict (Fiore and Donath, 2005). Adamic and Adar (2003) noticed that students in MIT and Stanford tend to link from their homepages to others who are similar to them. Adamic et al. (2003) analyzed Club Nexus, an online community at Stanford University. They found out that users had different degrees of homophily depending on the terms they used to describe themselves. Users describing themselves as “sexy” had a much higher chance of connecting to others with the same description than those describing themselves as “intelligent” or “responsible”.

In marketing research, it has been noted that the more homophilic a relationship between two individuals is, the more likely it is to cause word of mouth recommendations (Brown and Reingen, 1987). Moreover, Brown and Reingen noted that strong links were more likely to cause recommendations than weak links. Bruyn and Lilien (2004) have noted that the similarity of values and interests between sender and receiver affects the appeal of word of mouth communication from the sender. However, they state that this effect is stronger for personal products requiring a degree of confidence (e.g., a physician) than for impersonal products (such as a television). For impersonal products, perceived authority or social status, such as age or expertise on the topic, have more influence (Bruyn and Lilien, 2004).

2.4.4 Theories of proximity

In addition to similar interests or attributes, also physical proximity explains the formation of communication networks between individuals. This is because people, who live close to each other, have more chances to interact and meet with each other. The tendency to form connections to those who are physically close to oneself is often called *the propinquity effect* in social psychology (Kadushin, 2004, see for example).

The effect of proximity has been observed in a number of different environments. Zahn (1991) has noted that in an office setting, when physical distance between two individuals increases, the probability of them engaging in face-to-face communication becomes significantly smaller. By studying the communication between engineers in R&D organizations Allen (1977) stated that when distance between two persons grows to over 30 meters, the chance of communication becomes as small as if they were miles apart.

Online communities do not have all the burdens of geographical communication, because in theory it is as easy to communicate online with someone who is next to you as it would be with someone who is on the other side of the world. Cairncross (1997) among others has even suggested that communication technology makes the effect of distance vanish as people can communicate effectively over large distances.

Nevertheless, the tendency to form relationships with those who are geographically close, has also been observed in online communities. By studying the online communities of bloggers using LiveJournal, Liben-Nowell et al. (2005) noted that approximately 70 percent of friendships can be explained by geography.

People from different countries or even different cities tend to have separate subcultures and even speak different languages. This is why even in online environments it can be easier to communicate with those who are physically close, than with those, who live in more distant places, even though one might never meet the people face-to-face.

Another explanation for the geographical nature of online communication is that online communities reflect also social interactions of the physical world. Based on interviews with users, Boyd (2006) tells that in Friendster and MySpace, two common reasons for friendship with other users are that they are actual friends, and that they are acquaintances, family members or colleagues.

2.4.5 Relationships between the theories

Each of the three theories about friendship formation mentioned above, that is, the focus theory, homophily, and the propinquity effect, can be described in terms of traits of individuals. For the focus theory, the traits of a person are the different foci the person participates in. Because of the constraints on the foci, some traits are stronger while others are weaker. Homophily is clearly related to traits, since the more similar two persons are in terms of a number of features, the more likely they are to interact. From this point of view, the propinquity effect is just a special case of homophily, where the closer people are in physical space, the more likely they are to interact.

These various kinds of traits explain together a large part of friendships formed by a person. Some traits for an individual could be, for example, *"lives close to London"*, *"enjoys classical music"*, *"is part of a focus consisting of his family"* and *"is male"*. Although the features are quite different, they are likely to be additive, that is, the more two people share traits, the more likely they are to form friendships.

The general nature of the traits serves as a motivation for the M0 model presented in Chapter 5. However, sometimes it is difficult to describe exactly what traits have caused some properties of networks, since traits related to demographics, interests and geography are often overlapping. These challenges are given thought to when analyzing the experimental results in Chapter 7.

Chapter 3

Data analysis by clustering

Finding meaningful structure in networks is a type of *data analysis* problem, where the idea is to take a network data set and distill some meaningful information from it, such as a clustering of the network or some statistical properties of the network, and then to interpret the network data based on the results. This chapter starts by introducing the basic concepts of data analysis and clustering. Next, clustering of networks is discussed and a number of deterministic and probabilistic clustering algorithms are presented.

3.1 Data analysis

Berthold and Hand (1999) give a concise definition for data analysis: “*Data Analysis is the process of computing various summaries and derived values from a given collection of data.*” The important point of this definition is that it stresses the word *process*, that is, data analysis is not one single event of applying a statistical method, but instead, it involves making models, trying different approaches and, based on the results, adjusting the models to better describe the problem.

Tukey (1977) made a classical separation of data analysis into two categories: *confirmatory* and *exploratory*. In confirmatory data analysis, the aim is to give answers to certain specific questions (hypotheses) about the data, such as “*Can this attribute be predicted from the value of these?*” In contrast, a typical question in exploratory data analysis would be “*Are there any interesting structures in the data?*” (Berthold and Hand, 1999). A core part of exploratory data analysis methods is using graphics to understand underlying structures of data. Tukey (1977) considered that not enough emphasis was given to exploratory methods.

Data analysis has its origins in statistics, but many of the same problems are studied in machine learning and data mining communities, which originate from computer science and engineering. Traditionally, machine learning considers data analysis as the

process of *learning* the parameters or the missing values of a model. In statistics this is called *parameter estimation*. Research performed and methods used in statistical and machine learning communities overlap considerably and nowadays machine learning can be seen as a modern combination of statistics and computer science.

A classical but still often useful distinction in machine learning is that between *supervised* and *unsupervised* learning (Chapelle et al., 2006). In supervised learning, the system is first taught by using a *training set* with matching input and output values. After the system has been taught, it can be used to estimate the output values from new inputs (Dietterich, 2003). The opposite is *unsupervised learning*, where no training set with correct output is known (Dietterich, 2003, Ghahramani, 2004).

Nowadays, it is often difficult to say, whether a problem is closer to a supervised or an unsupervised learning, since many of the approaches lie somewhere in between, where the model is being taught with labeled data at the same time as properties about the unlabeled data are learned. These types of methods, which lay halfway between supervised and unsupervised learning, are sometimes referred to as *semi-supervised* learning (Chapelle et al., 2006, Zhu, 2005).

3.2 Clustering

In general, clustering means grouping unlabeled observations (Maimon and Rokach, 2005, page 1270). The idea with clustering is to find natural groups of items, which are in some way similar to each other and share some common properties. In social networks, clusters are often called communities, cohesive subgroups or cliques, while in biological protein interaction networks they are called functional modules (Palla et al., 2005). Clustering methods can be either hard or soft. In *hard clustering*, an item is assigned to just one cluster, while in *soft clustering* (i.e., *fuzzy clustering*) each item can belong to multiple clusters with different strengths (Jain et al., 1999).

Clustering methods have been widely studied in data mining and machine learning. The data that is clustered can be any set of items with discrete or continuous attributes associated to them. These items can be seen as data points in a high-dimensional space. The aim of the clustering algorithms is to find groups of items, which have in some way similar attributes, or are close to each other in the data space. Typically a distance measure is used in the clustering process to compare which items are similar to each other and should be assigned into the same cluster (Jain et al., 1999).

Clustering can be seen as unsupervised learning, since it is a function that takes the items to be clustered as input and outputs the cluster labels for each item. Typically, clustering is used in exploratory data analysis, where the aim is to see, whether the clus-

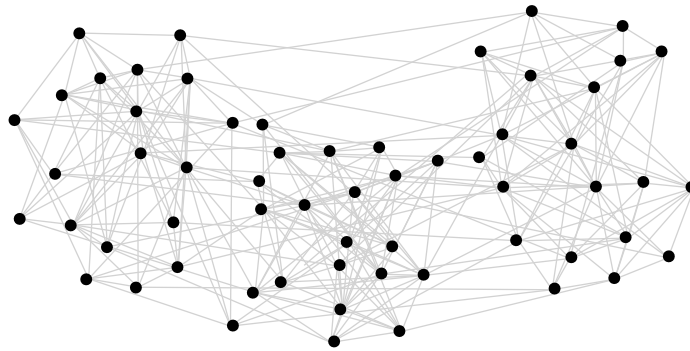


Figure 3.1. *Example of a computer generated network with local structure.*

ters can give interesting viewpoints to the data being analyzed. However, in practice different data analysis methods are often mixed together. For example, a data set can first be clustered to give initial insights into the data. These insights can be then used as the basis of making hypotheses and confirming them with statistical tests.

3.3 Clustering of network elements

The methods used to cluster network elements differ from those used in clustering data items in high-dimensional spaces. This is because there is no underlying feature space in which distance measures between nodes could be directly applied. However, some of the methods from clustering data items can be also adapted for clustering in networks (Newman, 2003a).

The idea behind clustering network elements comes from the empirical observation that edges in real networks are seldom placed randomly. Instead, networks usually contain groups of nodes that are strongly connected to each other, so that there are a lot of edges between nodes that belong to the same group and only a few between nodes that are from different groups. The aim of the network clustering algorithms is to divide the network so that nodes in the same cluster have many connections between them and nodes placed in different clusters have only a few connections.

A network which would seem to have some kind of local structure is shown in Figure 3.1. The visualization has been generated with a *spring embedding* algorithm, where forces have been added between nodes and then the system has been relaxed (Fruchterman and Reingold, 1991). It seems like the network would separate into three clusters. However, the clustering of the network would be different depending on what clustering method is used, because the methods are based on different assumptions about the nature of the clusters.

3.3.1 Clustering coefficient

Before running any clustering algorithm, it would be interesting to know about a network, whether there is community or cluster structure in it. One measure for this is the clustering coefficient of Watts and Strogatz (Newman, 2003a, Watts and Strogatz, 1998). It quantifies the concept of *transitivity*, which means that friends of a person are likely to be friends with each other as well.

Clustering coefficient is defined for a node i as:

$$c_i = \frac{\text{number of triangles connected to node } i}{\text{number of triples centered on node } i}. \quad (3.1)$$

Clustering coefficient c_i describes the connectedness of the neighborhood of node i . The numerator is the total number of shared neighbors between node i and each of its neighbors. The denominator is the maximum amount of neighbors they could have in common. Thus, $c_i = 1$ when the node and all its neighbors are fully connected. For nodes with degree 0 or 1, c_i is assigned 0 (Newman, 2003a).

The clustering coefficient for the whole network is the average

$$c = \frac{1}{n} \sum_i c_i. \quad (3.2)$$

For a random network consisting of N nodes that are randomly connected by E edges, the clustering coefficient c_{rand} is d/N , where d is the mean degree of the network (Dorogovtsev and Mendes, 2003, page 16). Since $d = 2E/N$,

$$c_{rand} = \frac{2E}{N^2}. \quad (3.3)$$

Confusingly, there are two different definitions for the clustering coefficient. The other variant and the relationship between the measures is discussed in Newman (2003a). To make the concepts even more vague, in complex networks research clustering coefficient is sometimes referred to as transitivity of the network although, as mentioned earlier, transitivity has a more general meaning in social network analysis.

3.3.2 Clustering quality

Networks can be partitioned into discrete groups in a number of ways. The best approach for validating whether an algorithm has found the correct group structure,

would be to compare the clustering result to a known group structure of the network. The problem is that usually this kind of comparative data is not available. Therefore some statistical measure has to be used to estimate the quality of the clustering.

Modularity Q is a measure for assessing the quality of a group division of a network. The idea of modularity is that a good division into groups is such that nodes have many edges within groups and only a few between them. The modularity for a community is calculated by subtracting from the number of edges within a community the expected value of the same number for randomized graphs (Clauset et al., 2004). Modularity of a network is simply the sum of the modularities of all communities.

Q is always between zero and one. In theory, zero would mean that the group division of the network is purely random, while a value larger than zero would mean that the group division represents real structure in the network. However, in practice, due to fluctuations, even random networks may exhibit some group structure and have a non-zero Q (Guimera et al., 2004).

For a set of groups in a network, a *connection matrix* is defined, where each member e_{ij} represents the number of edges from group i to group j divided by the total number of edges in the network. The connection sum a_i for component i is defined as $a_i = \sum_j e_{ij}$ and represents the proportion of all edges that connect to vertices in group i .

When C is the total number of groups, Q can be computed by

$$Q = \sum_{i=1}^C (e_{ii} - a_i^2). \quad (3.4)$$

In a network where edges are placed randomly between vertices, without regard to the communities of the vertices, $e_{ij} = a_i a_j$ (Gustafsson et al., 2006), and thus $Q = 0$.

There are some problems with Q . One is that in a network with E edges, it may fail to find clusters which have less than $\sqrt{E/2}$ internal edges (Fortunato and Barthelemy, 2007). Work-arounds have been found for this (Kumpula et al., 2007, Muff et al., 2005), but no standard alternative has yet emerged.

Modularity has been defined only for a hard clustering. Calculating Q for soft clustering requires that each node is first assigned to its strongest cluster. However, in this way, a node may belong only to one cluster, which is not a viable assumption in smooth clustering.

To assess results from smooth clustering of networks, a probabilistic generalization of Q can be generated. However, based on preliminary tests, the probabilistic Q would

always seem to be below that obtained by hard assignment. Thus, because the result can be trivially improved by hard assignment of the nodes, the probabilistic Q would not seem to be useful for comparing clustering results. Another approach would be to do a hard assignment of the nodes to multiple communities, as suggested by Zhang et al. (2007). However, this method requires specifying an arbitrary threshold level for the hard assignment.

3.4 Algorithms for network clustering

Clustering of networks has been studied in physics, computer science and mathematical sociology, and many methods for finding meaningful clusters have been proposed. In recent years, one of the main directions has been to generate algorithms that can be used with large networks that have thousands or even millions of nodes.

In social network analysis, homophilic clusters are sometimes called *cohesive subgroups*, while some researchers in complex network research use the term *community*. Both terms can be defined as clusters that are obtained when the network is divided so that there are a lot of edges within groups and little between them.

Also non-cohesive group types are popular in sociology, such as grouping based on *structural similarity* (Kadushin, 2004, Michaelson and Contractor, 1992). In these methods, the idea is to group nodes that link to the same nodes into clusters. In social network analysis, grouping of structurally similar nodes is called also *block modeling*. The aim is to partition a social network into groups or *blocks* of individuals who have similar connections to others in the network (Wasserman and Faust, 1994).

Clustering approaches are often classified as either *deterministic* or *probabilistic* (i.e., stochastic) (Jain et al., 1999). Deterministic algorithms produce always the same output and pass through the same sequence of states, while probabilistic algorithms incorporate randomness in their functioning. Typically probabilistic algorithms assume that the data comes from a mixture of populations, and the aim is to find the distributions of these populations (Berkhin, 2002). Some deterministic and probabilistic clustering approaches are presented below.

3.4.1 Deterministic methods

Most deterministic approaches for clustering can be classified as either *divisive* or *agglomerative*. For networks, divisive clustering works by removing edges that are between tightly-bound groups of nodes (Castellano et al., 2004). This divides the network into disconnected parts. These parts can then be further subdivided to estimate the hierarchical structure of the network. Agglomerative clustering works in the opposite

direction. It starts by assigning each node into its own cluster, and then the clusters that have the shortest distance are joined until all nodes are in the same cluster.

Convergence of iterated correlations (CONCOR) (Breiger et al., 1975) is a commonly used block modeling algorithm. CONCOR is based on calculating correlations for all pairs of nodes based on the similarity of their neighbors. The correlation between the nodes is $+1$ if they have exactly the same neighbors, and -1 if the neighbors are the opposite. In this case, opposite neighbors means that for two nodes n_i and n_j , none of the neighbors of n_i are neighbors of n_j , and all the nodes that are not neighbors of n_i are neighbors of n_j .

In practice, however, the correlation typically lies somewhere between these values. A matrix of the correlations between the nodes is formed and correlations on this matrix are calculated to obtain a second-order correlation matrix. By repeating the process on each successive matrix, the correlations typically finally converge so that all connections get either a value of $+1$ or -1 . The nodes connected with $+1$ are assigned in the same group. The operation can be repeated recursively on the two groups to obtain a hierarchical clustering of the network (Breiger et al., 1975, Cyram, 2005). Streeter and Gillespie (1992) note that CONCOR has several weaknesses. First, it has not been proven that the method actually finds the structurally equivalent nodes. Moreover, it is unclear, what objective function is being optimized in the process.

An effective approach to network clustering proposed by Newman and Girvan (2004) is to maximize the modularity Q of the network. It is believed that optimizing Q of networks globally is an NP-hard problem (Brandes et al., 2006). Still, there are heuristic search strategies, which can be used to restrict the search space while preserving the optimization goal. The GN algorithm proposed by Newman and Girvan (2004) performs the optimization of Q in a divisive fashion.

A fast agglomerative method for optimizing Q , called “Newman’s fast” or NF algorithm, was originally proposed by Newman (2004). An optimized version of the algorithm for clustering networks with hundreds of thousands nodes was presented by Clauset et al. (2004). It was soon noticed that the algorithm does not work well for huge networks. Some further improvements address this problem (Danon et al., 2006, Wakita and Tsurumi, 2007). The improved version by Wakita and Tsurumi (2007) has been tested with a social network with over five million nodes, and even the resulting Q is better than with the algorithm by Clauset et al. (2004). Moreover, a method proposed by Pujol et al. (2006), which combines spectral analysis and modularity optimization, seems to provide an effective method for analyzing even large networks.

3.4.2 Probabilistic methods

A large part of literature on probabilistic methods originates from *stochastic block modeling* ideas in sociology and psychometrics (Airodi et al., 2007). The idea with stochastic block modeling is that links between two nodes depend only on the groups of the individuals, and the groups are considered as independent and identically distributed (i.i.d.) random variables (Getoor and Diehl, 2005). Nowicki and Snijders (2001) present a stochastic block model, which supports arbitrary many clusters and directional, weighted edges. However, according to Daudin et al. (2007), the estimation method of Nowicki and Snijders (2001) cannot be used on networks with over 200 nodes.

Several extensions to the stochastic block model have been presented. Kemp et al. (2004, 2006) modify the model by adding Bayesian priors to automatically determine the correct number of clusters. In the mixed-membership stochastic block model by Airodi et al. (2007) a node may belong to a number of groups with different degrees. A variational algorithm is used for finding the approximate posterior. Another extension to the stochastic block model is the mixture model proposed by Daudin et al. (2007).

A social network can also be represented by assigning positions in a latent space to the nodes in the network. The first methods to achieve this were presented in 1970's. The method by Hoff et al. (2002) uses a finite dimensional latent space into which the nodes of the network are projected. (Handcock et al., 2007) extend the model by Hoff et al. (2002) by clustering the nodes in the latent space. The latent space models have been demonstrated on networks with less than hundred nodes, and it remains unclear whether the approaches can be used on large networks.

Another interesting probabilistic approach is the latent group model by Neville and Jensen (2005), which jointly models links, attributes and group memberships of nodes. In the model, the probability of an edge between two nodes depends on the groups of the two objects. The authors state that using the EM algorithm on the model would probably converge into a local maximum. This is why the implementation resorts to first assigning nodes into groups with spectral decomposition and then estimating the missing parameters using the *Expectation Maximization* (EM) algorithm.

One recent method to probabilistic clustering is the mixture model by Newman and Leicht (2006). In the model, the nodes that link to the same nodes are assigned to the same cluster, i.e., the method is based on the structural similarity of the nodes. The likelihood of the model is maximized using the EM algorithm. The model is able to detect both homophilic and heterophilic structures in the network and it can handle both directed and undirected edges.

3.4.3 Conclusions on the clustering algorithms

The deterministic approaches have been demonstrated on a wide range of networks of different types and sizes. However, the methods typically explore only a small part of the search space, and the measures they use do not necessarily model the problems studied realistically. Moreover, noisy data may cause problems with these methods.

Although the stochastic methods presented above can model rich structures in a network, they would seem to have some limitations. All of the methods have been demonstrated with networks that have less than a thousand nodes. For many of the methods, the full posterior distribution is not obtained but instead, a point estimate of a likely solution is provided. Additionally, a number of tricks are used to overcome problems with effective inference of the model parameters.

The M0 algorithm that is presented in Chapter 5 can be seen as a type of stochastic block modeling algorithm. It can be used on networks with hundreds of thousands of nodes and it can approximate a posterior distribution for the cluster memberships without having to resort to any ad-hoc solutions.

Chapter 4

Bayesian inference

One challenge of data analysis is that the data is usually not clean. There is always some missing data, uncertainty and randomness. *Bayesian inference* provides a natural method for dealing with these problems.

In this chapter, some background information on Bayesian probability theory, modeling of processes, and inference of model parameters are given. These provide the basis for the next chapter, where a Bayesian approach for clustering networks is introduced.

4.1 Bayesian probability theory

Probability theory is the mathematical study of random events where uncertainty of events is dealt with by assigning probabilities to them. The degrees of uncertainty are presented with real numbers and the sum of the probabilities of all events is one. In Bayesian probability theory probabilities are not interpreted simply as frequencies of some event occurrences, but instead as degrees of belief.

In Bayesian analysis, the aim is to estimate the posterior distribution of the unknown parameters given the data and the prior density of the parameters. The difference between Bayesian analysis and frequentist analysis is that in Bayesian approach the parameters are given prior probabilities.

The name of Bayesian inference comes from the fact that the Bayes' theorem is used in Bayesian inference to obtain the probabilities of the parameters. The theorem originates from Thomas Bayes, an 18th century theologian and mathematician. Bayes' theorem deals with the relationship of two stochastic events x and y ,

$$p(x|y) = \frac{p(y|x) p(x)}{p(y)}. \quad (4.1)$$

In Bayesian terminology, $p(x|y)$ is called the *posterior* probability, $p(x)$ is the *prior* probability of x and $p(y|x)$ is the likelihood. The theorem can now be presented as (Bishop, 2006)

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{normalizing constant}}, \quad (4.2)$$

or, equivalently,

$$\text{posterior} \propto \text{likelihood} \times \text{prior}. \quad (4.3)$$

Bayesian methods provide means to incorporate beliefs on a certain problem into the process of data analysis. One example would be a coin toss experiment, where there might be some reason to believe that the coin is biased (Berthold and Hand, 1999). The beliefs about the expected behavior of the coin can be taken into account when choosing the prior probability distribution.

There are two main ways to choose a prior in Bayesian analysis. The first is to use some information *a priori* on the parameters, leading to an *informative prior*. This information may come from, for example, experts of the domain that is analyzed. The second case is when there is no information available that can be used in setting up the prior probabilities, or the information is on a generic level. In these cases one can use an *empirical prior*, which is learned from data, or an *uninformative prior* that makes as little assumptions as possible.

Although Bayesian methods are often advocated because of the possibility to assign prior distributions, in reality, the choice of priors is often difficult. The practical strength of Bayesian methodology comes from the possibility of creating *hierarchical models*, which are described below.

4.2 Hierarchical generative models

In machine learning, a probabilistic *generative model* describes, how a data set, or observations, can be constructed randomly from a set of parameters.

Generative models with multiple levels of variables are often called hierarchical models. They consist of observable outcomes, which are conditioned on some hidden parameters, which themselves are given a probabilistic model using some other parameters, called *hyperparameters*. This kind of approach is useful in simplifying complex problems and makes it also possible to use computationally effective strategies to solving the posterior probabilities (Gelman et al., 2003, page 117).

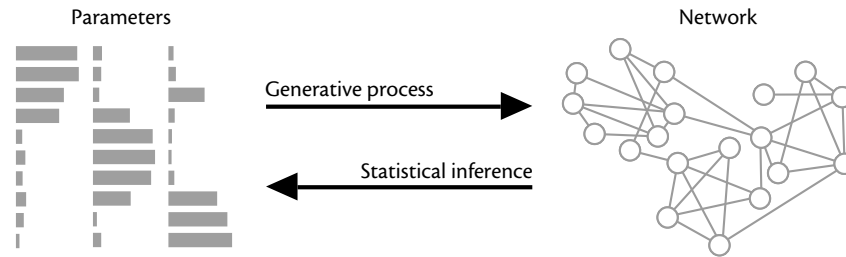


Figure 4.1. *Relationship between statistical inference and generative processes.*

A generative approach for statistical estimation consists of two phases. First, a full probabilistic generative model is defined, in which prior probabilities have been assigned to all of the hidden variables. After the model has been defined, Bayesian inference can be used to obtain the posterior distribution of the parameters. These parameters can either be general, describing the whole data set structure, or more specific information about local structures in the data set (i.e. a network).

As shown in Figure 4.1, a generative process can be postulated to generate a network from a set of parameters. Statistical inference is the reverse process: It can be used to retrieve a distribution over the set of parameters that could have generated the network. However, because of uncertainty, the exact parameters that originally generated the network cannot be obtained.

A good model describes well the data that is being analyzed. It should be possible to incorporate all the information available on the problem into the model. However, in many cases a compromise has to be made between model richness and computational efficiency.

4.3 Parameter inference

In Bayesian analysis, the joint distribution of all the variables of the problem is modeled. Typically only some of the variables are of practical interest. The aim is to obtain the distributions of these parameter values of interest, without regard to the uninteresting *nuisance parameters*. This process of removing the nuisance parameters is called *marginalization*, and it is essentially integration of the joint distribution over the unnecessary parameters y ,

$$p(x|c) = \int_y p(x, y|c) . \quad (4.4)$$

In the case of discrete parameters, this is the same as summing over the parameters. For two variables, of which y is the parameter to be eliminated, the marginalization can be expressed as

$$p(x|c) = \sum_i^{|y|} p(x, y_i|c) . \quad (4.5)$$

In practice, realistic or interesting models tend to be large and consist of many interacting elements, which makes exact inference of the parameters impossible. Sometimes, some of the parameters can be found exactly while others have to be inferred using an approximate method.

The typical approaches for parameter inference are *Markov Chain Monte Carlo* (MCMC) sampling and methods based on the EM algorithm. These two approaches are presented below.

4.3.1 Sampling

The idea with sampling methods is to draw a random sample from the posterior distribution without having to infer the distribution exactly (Bolstad, 2004). *Gibbs sampling* (Geman and Geman, 1984) is a widely applicable MCMC method for sampling the posterior distribution. It can be seen as a special case of the Metropolis-Hastings algorithm (see, e.g., Bishop, 2006). The idea is that the value of each parameter is updated iteratively conditioned on the values of all the other parameters. This chain of values converges to the joint distribution of the parameters.

A property of Markov Chain sampling is that it converges to a *stationary distribution*, which in the case of MCMC methods is the distribution from which we are interested in generating samples (Griffiths and Yuille, 2006). This means that given enough time, we can be certain that the Gibbs sampling procedure converges to the true posterior distribution over the parameters.

Ideally the sampling process is iterated until the estimations converge. Often the convergence is assessed by eye, based on a data plot of some convergence measure. Usually the algorithm is first iterated without taking any samples from it. This is called the *burn in* period of the algorithm. Then a number of samples is taken and posterior probabilities are estimated as an average of the samples.

4.3.2 The EM algorithm

Another approach for finding model parameters is to find the local maximum (mode) of the posterior distribution of the parameters, or alternatively finding the maximum likelihood estimate of the model. When the model is a mixture, these approaches usually lead to the EM algorithm. The EM algorithm (Dempster et al., 1977) alternates between estimating the parameters from the latent variable values (the *E step*), and estimating the latent variables based on the parameter values (the *M step*) (Griffiths and Yuille, 2006).

Variational methods are a generalization of the EM, where mode of a lower bound of the posterior distribution is found, by approximating the posterior distribution with some simple (factorizable) parametrized form.

4.4 Challenges with sampling

In mixture models with symmetry over the components, two common challenges are determining when the algorithm has converged, and avoiding *label switching*, that is, mixing of the components so that the meaning of the components changes during the sampling process.

4.4.1 Convergence of sampling

Several methods have been proposed for assessing whether a Gibbs sampler has reached convergence (Geweke, 1992, Ritter and Tanner, 1992). However, none of the methods can guarantee that the series has converged when the values seem to have reached a stationary plateau. This has been noticed also in practice when using Markov chain methods (Gelman, 1996). Often, to be able to answer questions about convergence, one would have to understand well the distribution that is sampled. Barber (2006) states that when the distribution is well-understood, then usually some exact technique would be preferable and no sampling would be needed.

Gelman (1996) gives two simple and practical approaches for analyzing convergence. The first is to run multiple parallel simulations and compare their results. The second is to run a test for a long time to make sure that it has reached the final plateau.

4.4.2 Label switching

When running simulations where items are assigned to groups or modeled with mixtures of distributions, permuting the component labels does not change the likelihood

or the posterior distribution. This means that in a clustering with C clusters, there are $C!$ different modes in the posterior (Geweke, 2007).

A sampler may end up fluctuating between these modes. This can lead to the ordering of the groups in a simulation to change from one simulation to another, or even in the middle of a simulation. This makes it hard to take averages of the probabilities of group assignments, because the group labels may have changed. This behavior is called *label switching*.

If an algorithm averages distributions for individual parameters over the modes, the result may become total nonsense. However, when using MCMC methods, such as Gibbs sampling, it is typical that no switching between modes occurs. This is because the algorithm typically converges to only one mode. In theory, the Gibbs sampling procedure should explore all of the symmetric modes. However, in practice, the probability of a transition between the modes may be so small that the transition never happens in practice.

Two approaches can be used to overcome label switching. One is to add some kind of identification constraints to the components. This is often difficult to implement in practice. The second is to post-process the results and to identify matching labels in the samples by using some similarity measure and clustering algorithms.

4.5 Distributions

There are a number of versatile probability distributions that can be used in constructing probabilistic models. In the next chapter, a probability model is introduced that includes Dirichlet distributions as the priors of the latent variables. This section presents the Dirichlet distribution, and its relationship to the multinomial distribution.

The multinomial and the Dirichlet are two particularly useful probability distributions for modeling statistical problems with nominal data. The main reason for this is that the Dirichlet is the *conjugate prior* for the parameters of the multinomial distribution. This means that in a Bayesian problem, where the likelihood function is multinomial and the prior is Dirichlet, then also the posterior distribution is Dirichlet. This is convenient in constructing generative processes, because it makes the posterior distribution algebraically tractable.

Binomial distribution is a discrete probability distribution that returns the number of successes in a sequence of n independent true/false experiments, when the probability of success is the same in every experiment. However, in many problems independent

experiments may take more than two values. This repeated selection from a set of outcomes can be modeled as a multinomial distribution.

The multinomial distribution can be explained with a series of experiments, where in each experiment the outcome can be one of finite k outcomes with probabilities $\theta = (\theta_1, \dots, \theta_k)$ (Gelman et al., 2003, page 83). The multinomial distribution gives the probability distribution for a certain vector of counts $c = (c_1, \dots, c_k)$ for the outcomes in a sequence of n independent experiments. The probability of a count vector c is

$$p(c|\theta) \propto \text{Mult}(c) = \frac{n!}{\prod c_i!} \prod \theta_i^{c_i}, \quad (4.6)$$

where $\sum \theta_i = 1$.

The Dirichlet is a multivariate generalization of the beta distribution, in a similar way as multinomial distribution generalizes the binomial distribution (Gelman et al., 2003, page 582). It gives the probability density that the probabilities of k outcomes are θ given that the counts for the observations are $\omega_i - 1$. It is defined as

$$p(\theta|\omega) \propto \text{Dir}(\omega) = \frac{\Gamma(\sum \omega_i)}{\prod \Gamma(\omega_i)} \prod \theta_i^{\omega_i - 1}, \quad (4.7)$$

where $\theta_i \geq 0$, $\sum \theta_i = 1$ and $\omega \geq 0$.

In the Bayesian context one may form a hierarchy of distributions, where the counts c are conditioned on the multinomial parameters θ , which are further *Dir*-distributed with parameter ω . The hyperparameter ω can be seen as a vector of "virtual" counts, before seeing the actual counts c (Minka, 2003). The posterior is then

$$p(\theta|c, \omega) \propto p(c|\theta)p(\theta|\omega) = \text{Dir}(\omega + c). \quad (4.8)$$

When using Dirichlet distributions, it is often feasible to assign the same value to the hyperparameters ω , that is, $\omega_i = s$ for all i , where $s \geq 0$. The probabilistic meaning of this would be that *a priori* the counts for all of the observations are the same. This vector of hyperparameters ω is called a *symmetric hyperparameter*. As a notational convenience, one may write simply $\omega = s$ and *Dir*(ω).

Figure 4.2 shows plots of the Dirichlet distribution over three variables on the simplex where the probabilities sum to one. In the left plot $\omega = 0.1$, in the center plot $\omega = 1.0$, and in the right plot $\omega = 10.0$. The value $\omega = 1.0$ corresponds to a uniform probability density and means that the same probability is assigned to any vector θ

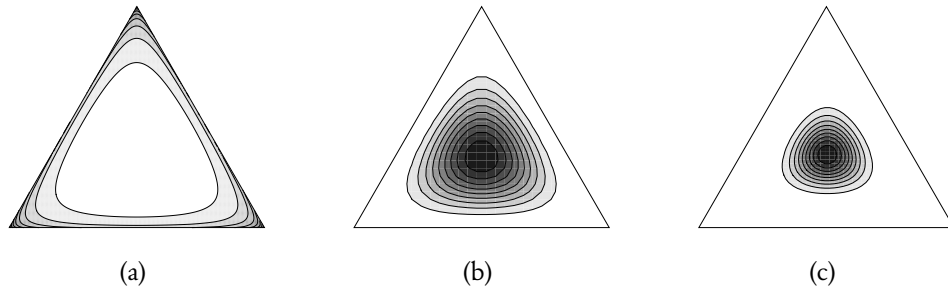


Figure 4.2. *Dirichlet distribution for three dimensions with parameter values $\omega = 0.8$ (a), $\omega = 4.0$ (b), and $\omega = 10.0$ (c). Darker colors indicate larger probabilities.*

that is on the simplex plane (Gelman et al., 2003). As can be seen in the figure, when $\omega < 1$, the probability mass tends to concentrate in the corners while $\omega > 1$ leads to a probability mass that is in the centre of the simplex.

On the other hand, when the number of dimensions is large, even with $\omega = 1.0$, most probability mass is close to the corners of the simplex, even though the probability density is still uniform. This is related to the *curse of dimensionality*, which states that in large-dimensional spaces most of the volume is close to the corners and far away from the center.

Chapter 5

A latent component model for networks

In this chapter a latent component mixture model for networks (the M0 model), is presented, as well as an algorithm based on Gibbs sampling that can be used to effectively infer the model parameters (the M0 algorithm). First, a simple model for a fixed number of components is introduced. This model is extended to a version that allows an infinite number of components. In practice, this means that the number of components required for modeling the problem is automatically selected, although it still depends on hyperparameters.

5.1 Model introduction

The M0 algorithm has been inspired by and is similar in its structure to *Latent Dirichlet Allocation* (LDA) (Blei et al., 2003), also known as *multinomial PCA* (mPCA) (Buntine, 2002), and other latent component models that have been developed for text document analysis and clustering (e.g., Hofmann, 1999). Some of these models use Dirichlet priors on observables in a similar fashion as the M0 model. The model parameters are typically estimated using either variational methods or Gibbs sampling.

It is worth noting that in the text analysis terminology *component*, which corresponds to a latent parameter in the hierarchical model, is different from the meaning of the word in network context, where it is used when referring to directly or indirectly connected groups of nodes. Nevertheless, in this and following chapters the meaning of *latent component* is borrowed from text analysis, and it is used as a common name for the latent parameters of the M0 model, which correspond to both diffuse latent traits and cluster-like structures in the network.

The M0 algorithm is based on a deceptively simple model of network growth. The model creates non-weighted networks having symmetric edges. The idea behind the model is that there are a number of latent components that generate a network. Each edge in the network is created by just one latent component. On a social network this

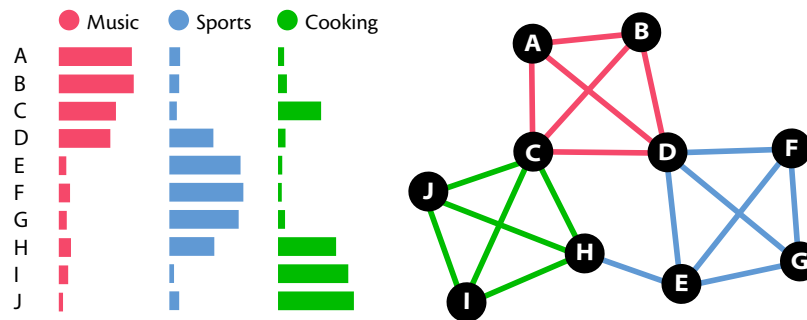


Figure 5.1. *A demonstration of the principles behind the M0 model.*

would mean that, if latent components correspond to traits, each edge represents one trait connecting the two persons.

The ideas behind the generative M0 model are illustrated in Figure 5.1. The columns on the left represent latent component probabilities for nodes (users) while the graph on the right shows the connections between the nodes as a graph. Each user, labeled with A-J, belongs with some probabilities to the three latent components labeled with *Music*, *Sports* and *Cooking*. Based on these probabilities a network can be generated by selecting for each edge first the latent component for the edge and then sampling the edge endpoints from the node probabilities of the latent components. On the right side is a social network between the users which could have been generated from the probabilities.

It is interesting to compare the M0 model to the models of friendship formation discussed in Section 2.4. The latent components in the model can be interpreted as something similar to the foci by Feld (1981). Moreover, the base idea of the model, i.e., users sharing traits are more likely to be connected, is well supported by the theories and observations in sociology.

The model does not cover many properties of friendship networks, such as models that change in time. Moreover, the M0 model cannot represent heterophily in networks. As noted earlier, there are other algorithms, such as the one proposed by Newman and Leicht (2006), which can find both heterophilic and homophilic structures in a network.

5.2 The finite mixture model

Based on the ideas presented above, a generative model for networks can be constructed. In the model, the component probabilities are drawn from a Dirichlet distribution. This type of model is called a *finite mixture model*. In finite mixture models,

data is assumed to have been generated from a mixture with a pre-determined number of components (see, e.g., McLachlan and Peel, 2000, pages 5-6). The term is slightly deceiving, because the model is not only finite but also of fixed-size. Thus, in the inference of the model parameters, the number of components has to be specified.

The following generative model is used to generate a network from the underlying probability distributions:

1. Draw θ from $Dir(\frac{\alpha}{C})$
2. For each component c in C components:
 - (a) Draw m_c from $Dir(\beta)$
3. For each of E edges:
 - (a) Draw a latent component z from θ
 - (b) Draw the first end point v_i from m_z
 - (c) Draw the second end point v_j from m_z .

In this generative model, a multinomial distribution θ is first generated over the C components from the Dirichlet distribution $Dir(\frac{\alpha}{C})$. Then for each c , a multinomial distribution over the N nodes is assigned by sampling the multinomial parameters from the Dirichlet distribution $Dir(\beta)$. After m and θ have been set up, new edges can be generated by first picking a latent component z from θ and then selecting the edge endpoints with probabilities m_z . By repeating this process for E times. A network with E edges is obtained as an output of the process.

The main reason for the form of the hyperparameter for distribution over components, $\frac{\alpha}{C}$, is that it makes it more convenient to derive the conditional link probabilities in Section 5.3.2. In addition, the form is motivated by the fact that in this way, the effect of the prior on the posterior is constant and does not depend on the number of components C (see, Navarro et al., 2006).

Plate models can be used to present relational data graphically (Heckerman et al., 2004). The plate model representation of the generative model for M0 is shown in figure 5.2. In the model, nodes are variables, arrows indicate dependencies between the variables and plates represent replicated structures (Buntine, 1994).

5.2.1 Joint distribution

The first part of solving a Bayesian inference problem is to derive the joint distribution of all the variables in the model. From the joint distribution, all the necessary distri-

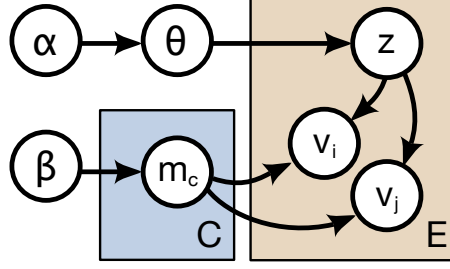


Figure 5.2. Plate model representation of the MO generative process.

butions can be obtained by integrating out (i.e., marginalizing over) the parameters that are not needed. The joint distribution for the Dirichlet prior model is

$$\begin{aligned}
 p(L, Z, m, \theta | \alpha, \beta) &= p(L, Z | m, \theta) \times p(m | \beta) \times p(\theta | \alpha) \\
 &= \prod_l \theta_{z_l} m_{z_l v_i} m_{z_l v_j} \times p(m | \beta) \times p(\theta | \alpha) \\
 &= \prod_{iz} m_{zi}^{k_{zi}} \prod_z \theta_z^{n_z} \times \frac{\prod_{iz} m_{zi}^{\beta-1}}{D(E, \beta)^C} \times \frac{\prod_z \theta_z^{\alpha-1}}{D(C, \frac{\alpha}{C})} \quad (5.1) \\
 &= \prod_{iz} m_{zi}^{k_{zi} + \beta - 1} \prod_z \theta_z^{n_z + \frac{\alpha}{C} - 1} \times \frac{D(E, \beta)^{-C}}{D(C, \frac{\alpha}{C})} \\
 &\equiv \prod_{iz} m_{zi}^{k_{zi} + \beta - 1} \prod_z \theta_z^{n_z + \frac{\alpha}{C} - 1} \times B\left(\frac{\alpha}{C}, \beta\right),
 \end{aligned}$$

where the Dirichlet distribution normalizer $D(X, \xi) = \Gamma(\xi)^X / \Gamma(X\xi)$ (for symmetric parameters ξ), Z is the set of observed components for each edge, L is the set of edges, N is the number of nodes, and C is the number of observed components. Further, v_i and v_j are the endpoints of the edge l , n is a vector containing counts for edges for each component, and k_z is the vector of the number of edges in component z .

For notational convenience, the normalizing parameters depending on α , β , E and C are denoted with $B(\frac{\alpha}{C}, \beta)$. $\Gamma(\cdot)$ is the Gamma function, a generalization of the factorial function with the property $\Gamma(z) = (z - 1)!$.

5.2.2 Conditional link probabilities

In theory, one could obtain the posterior distribution by sampling directly from the joint distribution presented in Equation 5.1, and then averaging over parameters that are not needed. This means simply discarding the nuisance parameters and calculating the average over samples for the parameters of interest. However, in practice, this

would be relatively hard because of the size of the parameter space. Instead, if possible, it makes sense to integrate over unnecessary parameters analytically.

The θ and m parameters are not actually needed, since the main thing that is of interest, in our case, in the inference are the component assignments Z for each of the edges. Based on these assignments, approximations for θ and m can be reconstructed. Following Griffiths and Steyvers (2004), marginalizing over θ and each of the m_z , the joint distribution of the edges and component probabilities is obtained:

$$\begin{aligned}
p(L, Z|\alpha, \beta) &= \int_{m, \theta} p(L, Z, m, \theta) dm d\theta \\
&= B\left(\frac{\alpha}{C}, \beta\right) \int_m \prod_{iz} m_{zi}^{k_{zi} + \beta - 1} dm \int_{\theta} \prod_z \theta_z^{n_z + \frac{\alpha}{C} - 1} d\theta \\
&= B\left(\frac{\alpha}{C}, \beta\right) \prod_z \frac{\prod_i \Gamma(k_{zi} + \beta)}{\Gamma(2n_z + N\beta)} \frac{\prod_z \Gamma(n_z + \frac{\alpha}{C})}{\Gamma(E + \alpha)}.
\end{aligned} \tag{5.2}$$

Typically not all unnecessary variables can be marginalized. Instead, one may integrate away some of the nuisance variables and sample over the others with MCMC methods, such as Gibbs sampling. This is called *Rao-Blackwellization*. As shown by Casella and Robert (1996), this leads to an estimator with a smaller variance than what would be obtained by using only Monte Carlo sampling. Sometimes the Gibbs sampling procedure based on the marginalized equation is referred to as *collapsed Gibbs sampling*.

An efficient collapsed Gibbs sampling procedure is to sample sequentially the component for each edge from the component probability distribution of that edge, given all the other links and component assignments in the network. The latter is obtained by separating one edge, say l_0 , from the product in Equation 5.2. The edge l_0 is associated with one component z_0 and connected to two nodes i_0 and j_0 . k' , n' and E' denote counts as if the link l_0 would not exist at all. Based on this,

$$\begin{aligned}
p(L, Z|\alpha, \beta) &= p(L', Z', l_0, z_0) \\
&= p(L', Z') \times p(l_0, z_0|L', Z') \\
&= B\left(\frac{\alpha}{C}, \beta\right) \prod_z \frac{\prod_i \Gamma(k'_{zi} + \beta)}{\Gamma(2n'_z + N\beta)} \frac{\prod_z \Gamma(n'_z + \frac{\alpha}{C})}{\Gamma(E' + \alpha)} \\
&\quad \times \frac{(k'_{z_0 i_0} + \beta)}{(2n'_{z_0} + 1 + N\beta)} \frac{(k'_{z_0 j_0} + \beta)}{(2n'_{z_0} + \beta)} \frac{(n'_{z_0} + \frac{\alpha}{C})}{(E' + \alpha)}.
\end{aligned} \tag{5.3}$$

Thus, the conditional probability of component z_0 given all the other links and component assignments is proportional to $p(l_0, z_0|L', Z')$, that is

$$\begin{aligned}
p(z_0|L', Z') &\propto p(l_0, z_0|L', Z') \\
&\propto \frac{(k'_{z_0 i_0} + \beta)}{(2n'_{z_0} + 1 + N\beta)} \frac{(k'_{z_0 j_0} + \beta)}{(2n'_{z_0} + \beta)} \frac{(n'_{z_0} + \frac{\alpha}{C})}{(E' + \alpha)}. \tag{5.4}
\end{aligned}$$

5.3 The infinite mixture model

A significant limitation with the finite mixture model is that the number of components has to be known beforehand. In many problems it would be reasonable to assume that the number of components, such as different types of interactions between people, is infinite, although only the most common forms of interaction are observed in practice.

The problems with the finite component model can be overcome by using an infinite mixture model, e.g., a Dirichlet Process (DP) mixture model. In the DP model, instead of drawing components from a finite dimensional Dirichlet distribution, a Dirichlet Process with countably infinite number of components is used.

The Dirichlet Process is a generalization of Dirichlet distribution for an infinite number of components, of which only a finite number are observed in practice. It allows the model complexity to grow when the amount of data increases. The Dirichlet Process was constructed by Freedman (1963) and the statistical theory for the process was detailed by Antoniak (1974), Ferguson (1973).

The generative model presented in Section 5.2 can be converted to the Dirichlet Process model by sampling θ from a Dirichlet Process instead of an Dirichlet distribution:

1. Draw θ from $DP(\alpha)$
2. For each of the ∞ components:
 - (a) Draw m_z from $Dir(\beta)$
3. For each of L edges:
 - (a) Draw a latent component z from θ
 - (b) Draw first end point v_i from m_z
 - (c) Draw second end point v_j from m_z .

What is peculiar about this model is that in theory, m_z is drawn for each of the components in θ , that is, over an infinite component count. However, in the inference

of component probabilities, this does not pose a practical problem, since the components are created first when they are needed, as described in Section 5.3.2.

5.3.1 Joint distribution

In a similar way as for the Dirichlet prior model, the joint distribution for the Dirichlet Process model is

$$\begin{aligned}
 p_{DPP}(L, Z, m|\alpha, \beta) &= p(L|Z, m) \times p(m|\beta) \times p(Z|\alpha) \\
 &= \prod_{iz} m_{zi}^{k_{zi}} \times \frac{\prod_{iz} m_{zi}^{\beta-1}}{D(E, \beta)^C} \times p(Z|\alpha) \\
 &= \frac{\prod_{iz} m_{zi}^{k_{zi}+\beta-1}}{D(E, \beta)^C} \times p(Z|\alpha) .
 \end{aligned} \tag{5.5}$$

In the infinite mixture model, the probability of observing a partition can be obtained by applying the formulas in Tavaré and Ewens (1997),

$$\begin{aligned}
 p_{DPP}(L, Z, m|\alpha, \beta) &= \frac{\prod_{iz} m_{zi}^{k_{zi}+\beta-1}}{D(E, \beta)^C} \times p(Z|n) \times p(n|\alpha) \\
 &= \frac{\prod_{iz} m_{zi}^{k_{zi}+\beta-1}}{D(E, \beta)^C} \times \frac{\prod_z (n_z!)}{(2E)!} \times \frac{(2E)! \alpha^C}{C! \alpha^{[2E]} \prod_z n_z} \\
 &= \frac{\prod_{iz} m_{zi}^{k_{zi}+\beta-1}}{D(E, \beta)^C} \times \frac{\alpha^C \prod_z (n_z - 1)!}{C! \alpha^{[2E]}} \\
 &= \frac{\prod_{iz} m_{zi}^{k_{zi}+\beta-1}}{D(E, \beta)^C} \times \frac{\alpha^C \Gamma(\alpha) \prod_z \Gamma(n_z)}{C! \Gamma(\alpha + 2E)} ,
 \end{aligned} \tag{5.6}$$

where $\alpha^{[2E]}$ is the Pochhammer symbol,

$$\begin{aligned}
 x^{[n]} &= x(x+1)(x+2) \dots (x+n-1) \\
 &= \frac{\Gamma(x+n)}{\Gamma(x)} .
 \end{aligned} \tag{5.7}$$

5.3.2 Conditional link probabilities

As for the finite model, the probability of a component for a certain link can be calculated conditioned on the components of the other links,

$$p_{DPP}(z_0|L', Z') \propto p_{DPP}(l_0, z_0|L', Z') . \tag{5.8}$$

Following Navarro et al. (2006), Neal (2000), by letting $C \rightarrow \infty$ in equation 5.4, the probability of an existing component z_0 is proportional to $p_{DP}(l_0, z_0|L', Z')$,

$$p_{DP}(z_0|L', Z') \propto \frac{(k'_{z_0 i_0} + \beta)}{(2n'_{z_0} + 1 + N\beta)} \frac{(k'_{z_0 j_0} + \beta)}{(2n'_{z_0} + N\beta)} \frac{n'_{z_0}}{(E' + \alpha)}, \quad (5.9)$$

and the probability of a new component z_{new} is proportional to $p_{DP}(l_0, z_{new}|L', Z')$,

$$p_{DP}(z_{new}|L', Z') \propto \frac{\beta}{(1 + N\beta)} \frac{\beta}{(N\beta)} \frac{\alpha}{(E' + \alpha)}. \quad (5.10)$$

The equations above can be used directly in Gibbs sampling by conditionally sampling the component for each edge. This is similar to the finite mixture model, the only difference is that the count vectors for the components have to scale up to accommodate the new components when they are created. Implementation details of the sampling procedures are discussed in detail in section 5.5.

In addition to taking the limit of $p_{DP}(z_0|L', Z')$ as described above, there are a number of other methods to generate samples from a Dirichlet Process. Two approaches are the stick-breaking construction (Sethuraman, 1994) and the *Chinese Restaurant Process* (CRP), which is closely related to the Blackwell-MacQueen Pólya Urn presented by Blackwell and MacQueen (1973).

The Chinese Restaurant Process leads to exactly the same partition as the equations 5.9 and 5.10 above (Navarro et al., 2006). The name comes from Chinese restaurants in San Francisco that have a seemingly infinite capacity. The process works as follows. Each person who enters the restaurant is given a seat next to a table with customers, based on the seatings of all previous customers. People are likely to be placed in the popular tables. However, with probability α , the customer may also be placed in a new table. This process is repeated for all the customers.

The Pólya Urn representation uses a different metaphor of essentially the same process. Colored balls are drawn from a urn with a probability proportional to their mass. One of the balls is black and has the mass α while all the others have a mass of one. Each time a ball is drawn, it is placed back in the urn and another ball of the same color is added. When a black ball is drawn, it is put back in the urn and a ball of a new color is added.

Both of these allegories illustrate the clustering behavior of the DP. The new observations are likely to take the same values as the previous ones. For large α many clusters will form while a small α leads to just a few clusters.

The clustering of the samples drawn from the DP does not depend on the order in which the clusters are assigned to items, i.e., the items are said to be *exchangeable* (Blei et al., 2003). In the CRP setting this means that how the persons entering the restaurant cluster around tables does not depend on the order in which the persons arrive. This is a natural justification for a Gibbs sampling procedure based on the DP.

5.3.3 Marginal likelihood

The posterior probability of the data given the model, in this case $p_{DP}(L|\alpha, \beta)$, is often called either the *marginal likelihood* of the hyperparameters or the *evidence* of the model. It is a useful measure when performing sampling of the posterior distribution, because it can be used to monitor convergence of the sampling process and to evaluate the result of the inference.

The marginal likelihood is obtained by marginalizing over the parameters from the likelihood function. In the case of a sampling process this is estimated by an average over S samples, that is

$$\begin{aligned} p_{DP}(L|\alpha, \beta) &= \int_{m, \theta, z} p(L, Z, m, \theta|\alpha, \beta) dm d\theta dZ \\ &\approx \frac{1}{S} \sum_t p(L|Z^{(t)}, m^{(t)}, \theta^{(t)}), \end{aligned} \quad (5.11)$$

where $p(L|Z, m, \theta) = \prod_{iz} m_{zi}^{k_{zi}}$, as in the joint distribution, and $x^{(t)}$ denotes the value of x in sample t . The likelihood can also be obtained precisely from the joint distribution in equation 5.5 by marginalizing over m and dividing with $p(Z|\alpha)$,

$$\begin{aligned} p_{DP}(L|Z, \alpha, \beta) &= \frac{1}{D(E, \beta)^C} \prod_i \frac{\prod_j^N \Gamma(k_{ij} + \beta)}{\Gamma(2n_i + N\beta)} \\ &= \prod_i \frac{\Gamma(N\beta)}{\Gamma(\beta)^N} \frac{\prod_j^N \Gamma(k_{ij} + \beta)}{\Gamma(2n_i + N\beta)}. \end{aligned} \quad (5.12)$$

In practice, it is often useful to operate with the logarithm of the likelihood, which replaces the products and divisions with sums and subtractions. This yields

$$\begin{aligned} \log p_{DP}(L|Z, \alpha, \beta) &= C (\log \Gamma(N\beta) - N \log \Gamma(\beta)) \\ &\quad - \sum_i \left(\log \Gamma(2n_i + N\beta) - \sum_j \log \Gamma(k_{ij} + \beta) \right). \end{aligned} \quad (5.13)$$

5.4 Hyperparameter values

In the model, the hyperparameters α and β control the Dirichlet distributions from which the component mixture and the components for the edges are drawn.

Parameter α is constant for all latent components, thus the a priori knowledge about the sizes of all latent components are the same and the model cannot represent correlation between the latent components. A large α implies that all the latent components are of the same size, while with a small α the latent components are with a high probability of different sizes.

Likewise, the hyperparameter β is same for all nodes. This means that *a priori* the component distribution of each node is the same. The effect of a large β is that a node is expected to belong to many components, while a small beta leads to the node belonging only to a small number of components. By varying the α and β parameters different properties in the network can be found. Small β values lead to more discrete communities, while large β values find smoother latent component structures.

5.5 Implementation of the algorithm

The algorithm in itself is quite straightforward to implement. However, an effective implementation for huge networks requires optimizations.

Three versions of the implementations are presented. The first one is the simplest possible, which uses arrays for storing vectors that are updated during the iteration process. The second implementation replaces the sampled component structures with hash tables. The third implementation further improves the algorithm performance and memory requirements by using a self balancing binary tree to store the component probabilities.

5.5.1 Simple implementation

The simplest way to implement the M0 algorithm using a Gibbs sampling is to represent all the required node and edge properties as dense arrays and, in each iteration, update these data structures as needed. With this implementation, lookup, insertion, and removal can be performed in constant time.

The simplest implementation of Gibbs sampling requires the following data structures:

- $L[edges]$: a list of edge endpoints
- $Z[edges]$: current latent component of each edge

- $K[nodes, components]$: the component-wise node degree
- $A[components]$: the count of edges in a component
- $P[components]$: latent component probability distribution, used in the iteration

Algorithm 1, presented on a separate page, shows the pseudocode for SIMPLE-GIBBS-SAMPLING, which takes as its input a list of edges and the hyperparameters, and calculates values of Z and K . In practice, multiple samples of Z and K are taken. By averaging over them, distributions over Z and K are obtained.

The algorithm proceeds by iteratively sampling new values for Z for each edge. First the current K values for both edge end nodes and A are updated to subtract the effect of the current edge. Then a latent component probability distribution P is calculated and a new component value is sampled from the distribution and stored in Z .

The worst-case running time of the simple implementation on a network with N nodes, E edges, C components and I iteration rounds is $\mathcal{O}(INC)$. The memory consumption for this simple implementation scales as $\mathcal{O}(NC + E + C)$.

The algorithm requires memory structures for storing the latent components of each of the nodes in the network. This makes the simple implementation infeasible to use with large networks and many components. However, in practice, each node is connected only to a small subset of nodes. The number of components a node can belong to is upper-bounded by the degree of the node.

This approach works well for a small amount of static components, but when the number of components grows, this method slows down because P needs to be calculated for every component. Another problem with large component counts is that the memory requirement for K grows when the amount of components grows.

5.5.2 Hash table implementation

An easy way to improve the memory efficiency of the implementation is to use hash tables over K instead of dense matrices to represent the latent components. Thus, each $K[nodes]$ contains a pointer to a hash table for the components in the node.

Calculating effectively P for an edge in the hash table implementation is somewhat more complicated than with the simple implementation. First, all probabilities for the components in the start node of the edges are calculated. Then, the probabilities for the components in the end node that were not updated for the start node are calculated. Finally, all the other probabilities are updated.

By replacing the dense data structures with sparse hash tables, the average memory consumption can be lowered to $\mathcal{O}(Nd + E + C)$, where d is the mean degree of the

Algorithm 1 SIMPLE-GIBBS-SAMPLING A simple implementation for the M0 algorithm

SIMPLE-GIBBS-SAMPLING(α, β, L)

```

 $t_{nodes} \leftarrow$  node count
for  $c \leftarrow 1$  to components do ▷ initialize data structures
     $A[c] \leftarrow 0$ 
    for  $n \leftarrow 1$  to  $t_{nodes}$  do  $K[n, c] \leftarrow 0$ 
for  $i \leftarrow 1$  to iterations do ▷ main iteration loop
    foreach  $l$  in  $L$  do
         $v_i \leftarrow$  first node of  $l$ 
         $v_j \leftarrow$  second node of  $l$ 
        if  $i \neq 1$  do
             $z_{old} \leftarrow Z[l]$ 
            decrement  $K[v_i, z_{old}], K[v_j, z_{old}], A[z_{old}]$ 
         $p_{tot} \leftarrow 0$ 
        for  $c \leftarrow 1$  to components do
             $p_c \leftarrow$  CALC-PROBABILITY( $A[c], K[v_i, c], K[v_j, c], t_{nodes}, \alpha, \beta$ )
             $P[c] \leftarrow p_c$ 
             $p_{tot} \leftarrow p_{tot} + p_c$ 
             $z_{new} \leftarrow$  SAMPLE-INDEX( $P, p_{tot}$ )
             $Z[l] \leftarrow z_{new}$ 
            increment  $K[v_i, z_{new}], K[v_j, z_{new}], A[z_{new}]$ 
return  $K, Z$ 

```

CALC-PROBABILITY($n_c, k_a, k_b, t_{nodes}, \alpha, \beta$)

```

return  $\frac{(k_a + \beta)(k_b + \beta)(n_c + \alpha)}{(2n_c + 1 + \beta t_{nodes})(2n_c + \beta t_{nodes})}$ 

```

SAMPLE-INDEX(P, p_{tot})

```

 $r \leftarrow$  uniform random number in  $[0.0, p_{tot}]$ 
 $p_{sum} \leftarrow 0$ 
foreach  $p_{cur}$  in  $P$  do
     $p_{sum} \leftarrow p_{sum} + p_{cur}$ 
    if  $p_{sum} \geq r$  do
        return index of  $p_{cur}$ 

```

network. Since $d = 2E/N$, memory consumption scales as $\mathcal{O}(E+C)$. Thus, memory consumption grows linearly with respect to the number of edges in the network. The lookup, insertion and removal of elements from a hash table has $\mathcal{O}(1)$ complexity, which means that the running time of the algorithm is $\mathcal{O}(INC)$ even with the sparse implementation.

5.5.3 Binary tree implementation

The hash table implementation consumes clearly less memory than the simple implementation. However, both the simple and the hash table based implementations suffer from the fact that for each edge, the probabilities for each latent component have to be recalculated, even if they would remain the same. Moreover, the sampling from the set of discrete probabilities can be slow, which is a problem if the number of components is many thousands.

To solve these problems, the component probabilities can be stored in a tree data structure. A tree consists of tree types of nodes: a *root node*, which is the topmost node that doesn't have any parents, *internal nodes*, which have child nodes, and *leaf nodes*, which are on the bottom of the tree and do not have any child nodes.

In the tree, each node represents a component and is assigned a weight which is relative to the default probability of sampling the component. Then for each edge in every iteration, only the probabilities that have been changed need to be recalculated. After the latent component Z for an edge has been sampled from the tree, the tree can be reverted back to its default form. The structure of the tree is described in detail below.

The tree used in the implementation is a self-balancing binary tree called Arne Andersson tree (AA tree) (see, e.g., Weiss, 1998, pages 474-481). Other self-balancing binary trees would be of equivalent performance. Weights have been added to the nodes of the tree, and when the weight of a node is changed, the modifications are propagated up to the parents of the node.

The use of a tree in sampling is similar to the method described by Wong and Easton (1980). The problem is as follows. A random sample of size k from items s_1, \dots, s_n needs to be drawn, where the items s_i are associated with weights w_i . An item should be drawn with a probability proportional to the weight of the item. Wong and Easton (1980) suggest to set up a binary tree of height $\mathcal{O}(\log(n))$ in time $\mathcal{O}(n)$ in a preprocessing step and using this tree as detailed below. Generating the tree takes $\mathcal{O}(\log(n))$, and updating the elements has the same cost.

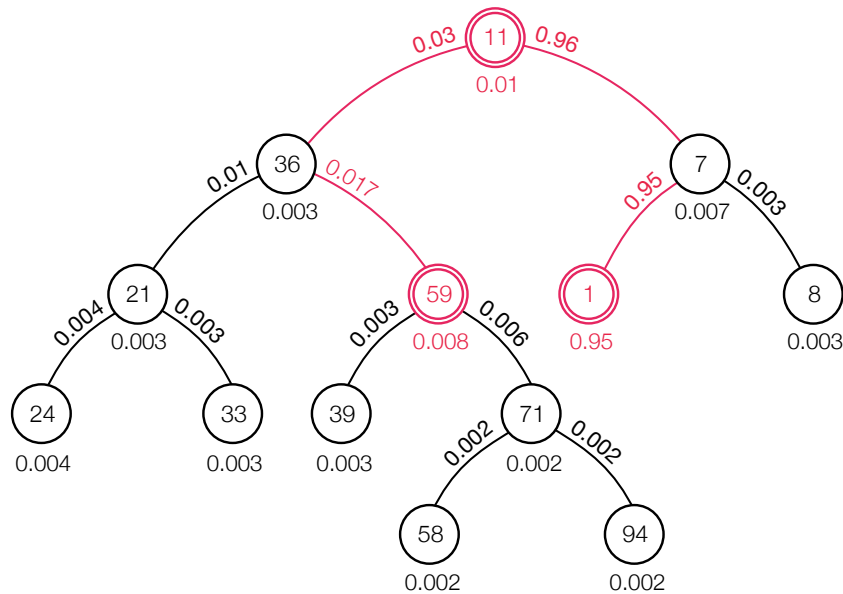


Figure 5.3. *Updating of the partial sum tree. Nodes and weights in black are in their default values. Nodes and weights in red have been recalculated and updated for the edge. Weights for components 1, 11 and 59 have been updated and propagated to their parents, so that the partial sums are correct. Clearly, the most likely component for this edge is 1 and has 98% probability of being sampled. After a component has been sampled for the edge, all the values marked in the figure with red are recalculated, reset to their default values and propagated to their parents.*

In the probability tree, for each node, three floating-point numbers are stored: the weight of the node (probability), the sum of weights for the left children of the node and the sum of the weights for the right children of the node. This kind of tree is called a *partial sum tree*. When any of the values are updated for a node, the corresponding changes are propagated to its parents. Figure 5.3 shows the state of the partial sum tree, when the tree has been updated to reflect the probabilities of the components for a single edge.

With these optimizations to the sampling of the components, the average running time can be lowered down to $\mathcal{O}(IEd \log(C))$. In many networks E and N are of the same order of magnitude, which means that the average running time is essentially $\mathcal{O}(IE \log(C))$. That is, the implementation scales linearly in the number of nodes and logarithmically in the amount of components.

With the binary-tree implementation, latent components for the edges can be inferred for networks with hundreds of thousands of nodes and thousands of components. The binary tree approach works perfectly for the Gibbs sampling, because the tree has to

be generated only in the beginning of the sampling process, and after that only the elements that have been modified are updated in the tree. Blue et al. (1995) provide another method for using binary trees in improving Monte Carlo simulations.

Chapter 6

Experimental setup

In the experiments, the aim is to find out how well the M0 algorithm performs at finding clusters in networks. The following questions form the basis of the experiments:

- What heuristics can be used to select the hyperparameter values for the algorithm?
- How well does the algorithm converge?
- Does the algorithm work on small networks?
 - Do the clusters correspond to known groups?
 - Are the clusterings good in terms of modularity?
- What is the performance of the algorithm on a large friendship network?
 - Is the algorithm memory consumption and running time such that it can be used on networks with hundreds of thousands nodes?
 - Do the clusters correspond to relevant structures, such as interests or nationalities of the users?

The experiments are divided into three parts. First, it is analyzed how the parameters α and β affect the number of communities found and the modularity of the community division in the network. This is done to get a rough picture of the effects of the parameters. Next, based on the results from the first part, the algorithm is tested on several small networks which are known to exhibit modular structure, and the results are compared to those achieved with the hierarchical fast community algorithm. Finally, a large-scale test is performed by analyzing a friendship network from the Last.fm online music recommendation service.

6.1 Material

In addition to the implementation of the M0 algorithm, the material consists of small test networks and a large friendship network collected from Last.fm.

6.1.1 Small test networks

Small and medium-sized networks from several domains are used in testing the algorithm (Table 6.1). The networks have been shown to exhibit community structure. All of the networks can be seen as standard test networks for community algorithms. All of the networks have a clustering coefficient that is magnitudes larger than the same value for a random network (see Equation 3.3), which would indicate that they have community structure. However, the clustering coefficients for the PGP and Email networks are clearly smaller than the clustering coefficients for the other networks.

Table 6.1. *Small networks used in testing the algorithm. In the table, n is the number of nodes in the network, m is the number of edges, d is the mean degree, and c is the clustering coefficient.*

Network	n	m	d	c
Karate	34	78	4.59	0.57
Football	115	613	10.66	0.40
Jazz	198	2 742	27.70	0.62
Celegans	453	2 025	8.94	0.65
Email	1 133	5 451	9.62	0.22
PGP	10 680	24 060	4.68	0.09
Physicists	27 519	116 181	8.44	0.65

The Karate network originates from a study by Zachary (1977) on the social relationships in a karate club. In the study, Zachary observed 34 members of a karate club for two years. During this period, there was some disagreement among the club members which led to the splitting of the club. The instructor of the club left and formed a new club, which around half of the club members joined.

The Football network depicts American football games between Division IA colleges during the fall season 2000. The nodes of the network represent football teams and edges the games between the teams. There is a known community structure for the network in the form of conferences. Teams are divided into conferences with around 8 to 12 teams in each. Games between teams that belong to the same conference are more frequent than between teams that belong to different conferences. The community structure for the network has been originally analyzed in Girvan and Newman (2002).

The Jazz network depicts collaborations among jazz musicians, who performed between 1912 and 1940 (Arenas, 2007, Gleiser and Danon, 2003). The Celegans network is an undirected and unweighted version of the neural network of the worm *Caenorhabditis elegans* (CDG, 2007, Watts and Strogatz, 1998). Email is a network of e-mail interchanges between members of the University Rovira i Virgili (Arenas, 2007, Guimera et al., 2003). PGP represents the giant component of a trust network of mutual signing of cryptography keys (Guardiola et al., 2002). Physicists is a co-authorship network of physicists working on condensed matter physics, originally from the arXiv.org database (Newman, 2001).

6.1.2 Last.fm data set

Last.fm is a personalized Internet radio where users may listen to music that matches their interests. The system builds a profile of the musical interests of a user based on the music the user has been listening to. This profile is used to recommend music to the user. The profile, as well as other information on the user, is also visible on a customizable home page of the user.

There are several social networking features incorporated into the Last.fm service. Users may form groups or communities with other users. They have also the possibility to ask other Last.fm users to be their friends. Friendships are shown to others on the home page of the user. The user can also see what music his friends are listening to in real time from his dashboard. Friendships are created by clicking on the “add as friend” button on the friend’s user page. The friend has to accept the request for a friendship to form. Last.fm friendships are always mutual (two-way), that is, if user A is a friend of user B, then the opposite is also true and user B is a friend of user A.

Users can also tag artists, albums, and tracks to create a classification of music styles. Tags may either describe the music styles (“garage rock”, “electronica”), geographical location (“finnish”) or any other characteristics such as “seen live”.

Last.fm has over 15 million unique active users every month (Lake, 2006). Of these, 780 000 users have listed at least one friend in their profiles (private communication with Norman Casagrande). Thus, only a small minority of users on Last.fm is using the friends-feature of the service. In addition, many people rather belong to groups than form direct friendship connections. This might imply that there are two types users on Last.fm: active users, who make use of the social networking services offered, and those who use the service only as an online radio. In this research only the active users are studied.

Last.fm allows fetching of user profiles, listening habits and friendship information via the Audioscrobbler web services (AudioScrobbler, 2007). The data is licensed with Creative Commons *Attribution, NonCommercial, ShareAlike* license, which means that the data can be used in non-commercial purposes as long as the original source of the data is accredited and derivative works of the data can be made as long as they are licensed with the same terms (Creative Commons, 2007).

Previously music tastes of Last.fm users have been analyzed by Bergstra (2006) for classifying music genres automatically. Liekens (2007) has got interesting preliminary results about the relationships between different music tastes by clustering the music tastes of the users using Principal Component Analysis. However, in these work the social network of the Last.fm users was not used.

Crawled data

A crawler was implemented in Java 1.5 (Java, 2007) to fetch user friendship and profile information via the web services. The information was stored into individual files on the server and tools were implemented for distilling the information from these files into a more easily managed form.

For testing the M0 algorithm, a friendship network with over 650 000 users was crawled from the Audioscrobbler service. This crawl contains over 90% of Last.fm users with friends and is clearly the main component of the Last.fm friendship network.

The Last.fm friendship network was crawled during March 2007 using depth-first search, starting with a single user to fetch a snapshot of the whole network. Then, in the beginning of April 2007, the profile information and the top artists of each user were crawled.

The Last.fm user profiles do not tell which are the tags best matching a user. However, these tags can be calculated based on the tags given to the artists the user has been listening the most. For each user, the top ten artists they have been listening to were crawled, and for each of these artists, their profiles were fetched from the Last.fm service. By weighing the tags given to each of these artists, the top tags for each user were calculated.

Subsets for experiments

In the experiments, the clustering results and the convergence of the algorithm are analyzed for the full Last.fm network as well as the subset of users, who have identified as being from the United States. In addition, for the purpose of visualizing the actual connections between individuals and clustering of their relationships, the subset of

users from Denmark and friendships between them are also clustered. Properties of these networks are shown in Table 6.2.

Table 6.2. *Networks crawled from Last.fm used to test the algorithm. In the table, n is the number of nodes in the network, m is the number of edges, d is the mean degree, and c is the clustering coefficient.*

Network	n	m	d	c
Full Last.fm	675 682	1 898 960	5.62	0.24
Last.fm USA	147 610	352 987	4.78	0.23
Last.fm Denmark	2 374	4 345	3.66	0.32

Properties of data

In general, because Last.fm is an online radio and music sharing site, connections are probably more likely between people with similar music tastes. However, as in other online communities, there are several reasons why a Last.fm user would include other users as his/her friends. One is to use the service for recommending music to friends and receiving recommendations from them. Other reasons include “spying” what music others are listening to, collection of friendships, and projecting social relationships from real-world into the online domain.

When analyzing the Last.fm data, there are some potential pitfalls: the data may be biased, values may be missing, and there may be outliers or incomplete entries (Maimon and Rokach, 2005, pages 1292-1293). Not all users have a public profile available and, because users can specify what they put in their profiles, there are also outliers in the data. Also the tags used for classifying users contain many tags that represent the same information and many different types of tags.

Figure 6.1(a) shows the distribution of Last.fm users in the countries with most users. A clear majority of users have listed *United States* as their country of residence. However, a close second is *[None]*, which means that the user has not listed any country in their profile. As shown in Figure 6.1(b), the most common tag for each user varies greatly and has a long tail. This could imply that the range of music tastes of the users is broad with lots of subcommunities.

The distribution of edges between users is shown in Figure 6.2(a). As is typical in social networks and internet communities in particular, the degree distribution of the network has a heavy tail. This means that most users have few connections, while some have a lot. A similar behavior has been noted by Adamic et al. (2003) in the “Nexus Net” social network. Figure 6.2(b) shows the age distribution of Last.fm users. Most users seem to be around the age of 20. Because the users can choose themselves what age they list in their profiles, some of the ages are outliers.

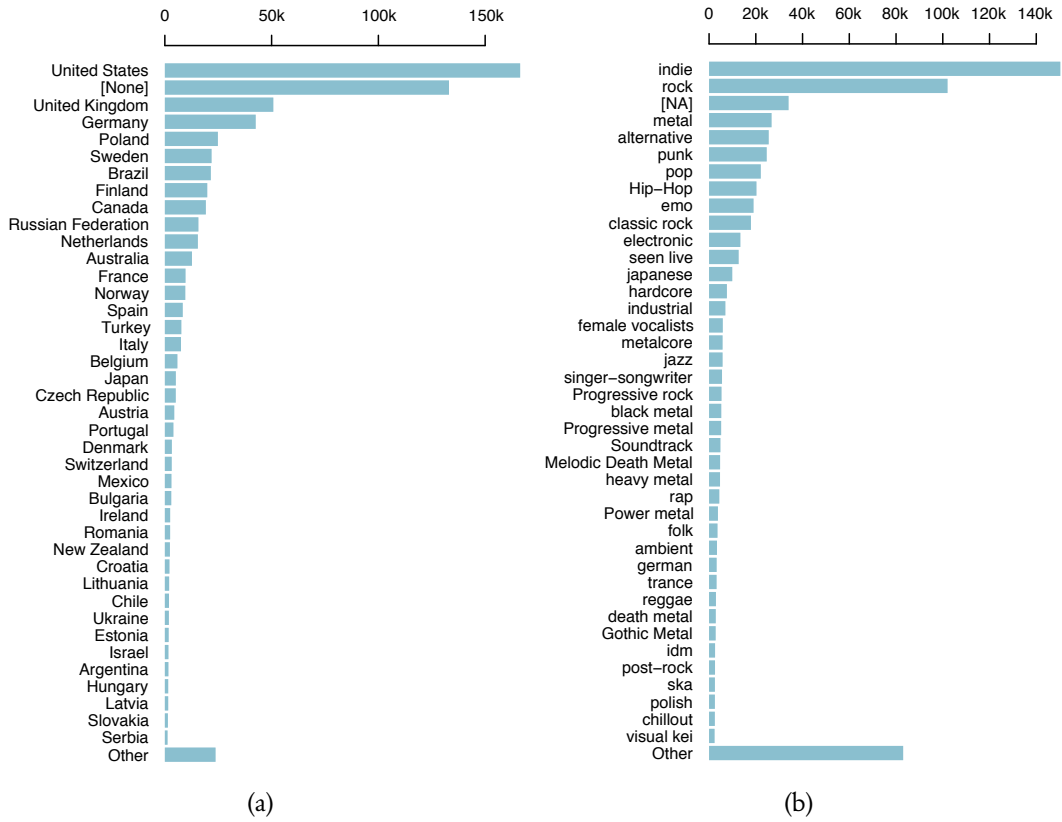


Figure 6.1. *Last.fm* users in different countries (left) and the most common tags for all users in *Last.fm* (right).

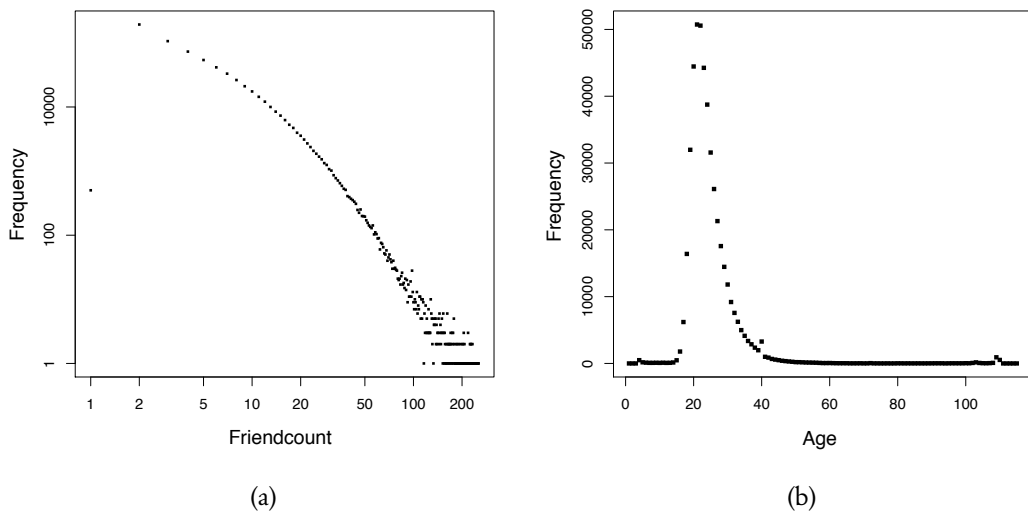


Figure 6.2. *Last.fm* friendship degrees have a heavy tail (left) while the age distribution peaks at around 21 years (right).

Of all Last.fm users who have listed their gender, 66% are males and 34% females. However, there are differences in the gender distribution between countries. For example, in the US, the proportion of females is larger with the amount of males being 60% and females 40%.

6.2 Methods

In this section, the methods that are used in the experiments are presented. The results of the experiments are given in the next chapter.

6.2.1 Algorithm implementation and analysis tools

The M0 algorithm was implemented in the Java 1.5 environment (Java, 2007). GNU Trove library (Trove, 2007) was used for effective data structures, Cern Jet (2007) libraries were used for random number generation, and the Prefuse (2007) visualization library served as basis for the network visualization engine.

The combination of the Dirichlet process model and the binary tree based implementation was chosen as the algorithm to be tested in the experiments, based on initial tests. This is because it does not require the user to specify the number of clusters that are searched for. In addition, the implementation is efficient in terms of memory and computational requirements.

R statistical environment was used for the analysis of the data and plotting figures from the algorithm runs. Heatmaps were drawn with the “heatplot” command from the MADE4 library (Culhane et al., 2005, MADE4, 2007).

6.2.2 Assessing algorithm convergence

Two different measures are used for assessing how well the algorithm has converged. The first measure is the marginal likelihood for the Dirichlet process model (equation 5.11) presented in Section 5.3.3. The second one is modularity (equation 3.4), which is applied to the network divisions by selecting for each node the most likely cluster based on the sampled posterior probability distribution.

6.2.3 Finding optimal hyperparameters

One challenge with the M0 algorithm, common to all algorithms with hyperparameters, is how to choose the hyperparameter values. In the experiments, for the seven small networks presented in Table 6.1 modularity, likelihood and node counts are

calculated with α and β values ranging from 10^{-7} to 10^3 . These are plotted to give a visual representation of how the hyperparameters affect the clustering.

6.2.4 Analyzing small networks

The algorithm is run with the hyperparameters that provided the best modularity to see how well the algorithm performs compared to four different algorithms (Clauset et al., 2004, Duch and Arenas, 2005, Girvan and Newman, 2002, Newman, 2006) for finding communities, that is, hard clusterings of a network. How well the M0 algorithm can perform as a community algorithm is evaluated by comparing the modularity values obtained with it to the results from the community algorithms.

6.2.5 Clustering Last.fm friendship network

A problem with clustering the whole network is separating the effect of the geography and other traits (homophily) from each other. This is why it is hard to say whether the clustering represents the geography (countries) of the users or their music tastes. To get a better understanding of how well the algorithm can separate users into groups based on their music tastes, the algorithm is first run on the whole network and compared to the tags of the users and then the algorithm is run on the subset of users in the main component who are from the United States.

Chapter 7

Results

In this chapter, the effect of the hyperparameters on the algorithm results is first demonstrated and the optimal hyperparameters in terms of modularity of the clustering are presented. Then, the algorithm is tested with small networks by using the hyperparameters from the previous phase and the results are compared to community algorithms. Finally, the clustering of the Last.fm friendship network is presented.

7.1 Finding optimal hyperparameter values

The component count changes as a function of both the α and β parameter values. Large α and small β lead to more components. The effect of hyperparameters affecting the component count is also demonstrated for an artificial network in Figure 7.1.

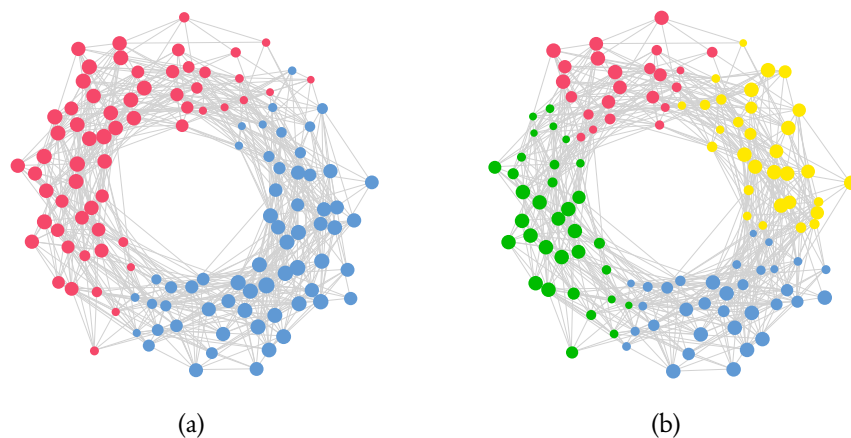


Figure 7.1. *An artificial doughnut-shaped network, which demonstrates the effects of different hyperparameters. On the left, $\alpha = 2.0$ and $\beta = 2.0$ and on the right, $\alpha = 10.0$ and $\beta = 0.0001$.*

Table 7.1. *Optimal hyperparameter values for the small networks in terms of modularity.*

Network	Size n	Clusters	α	β
Karate	34	4	0.025	0.05
Football	115	12	0.01	0.025
Jazz	198	3	0.01	0.5
Celegans	453	6	0.1	0.2
Email	1133	8	0.8	0.15
PGP	10 680	29	0.001	0.025
Physicists	27 519	14	2.0	0.1

Figure 7.2 shows the results of clustering four networks of different sizes with a range of α and β hyperparameter values. As can be seen from the graphs, maximizing the modularity of the solution with α and β tends to favor splits with only a few components, which means smaller α values and larger β values than what would result from maximizing the marginal likelihood. The largest likelihood values seem to be overfitted, because even for small networks the component counts grow quite large.

The results seem to be quite robust to changes in α , while the choice of β affects the results strongly. This could imply that by first selecting a rough value for α and then running more tests for β could be a good heuristic for finding good hyperparameter values. A prior distribution could be also assigned to the hyperparameters.

Based on the marginal likelihood values, reasonable estimate for the parameter values would be $\alpha = 1 \dots 10$ and $\beta = 10^{-3} \dots 10^{-2}$. However, many of these values lead to a large number of components. Moreover, in the figure, the optimal modularity and likelihood are found in different areas, with the β being typically smaller for the optimal modularity.

The tendency of the likelihood to prefer models with a large number of components implies that the complexity of the model is over-estimated and overfitting occurs. This could be avoided by performing further tests on leave-out data.

Table 7.1 displays the optimal hyperparameter values in terms of modularity for each of the networks obtained with the exhaustive search. One observation which can be made from the table 7.2 is that for the networks tested, the optimal hyperparameter values in terms of modularity tend to be smaller for larger networks. In the next section, these hyperparameter values are used to compare the M0 algorithm to community algorithms.

Since the results of the modularity and likelihood differ so dramatically, no general rule can be given on the hyperparameter values leading to optimal clustering. In terms of likelihood, a reasonable guess for the α value would be between 1 and 100 while

Table 7.2. *Network modularity with different algorithms. The hyperparameter values were obtained with exhaustive search, see, Section 7.1.*

Network	Size n	GN	CNM	EO	N06	M0	Clusters	α	β
Karate	34	0.401	0.381	0.419	0.419	0.402	4	0.025	0.05
Football	115	0.601	0.577	-	-	0.603	12	0.01	0.025
Jazz	198	0.405	0.439	0.445	0.442	0.443	3	0.01	0.5
Celegans	453	0.403	0.402	0.434	0.435	0.417	6	0.1	0.2
Email	1133	0.532	0.494	0.574	0.572	0.567	8	0.8	0.15
PGP	10 680	0.816	0.733	0.846	0.855	0.667	29	0.001	0.025
Physicists	27 519	-	0.668	0.679	0.679	0.701	14	2.0	0.1

the β which maximizes the likelihood is in the range from 0.001 to 0.01. For the modularity, optimal α value seems to be between 0.001 and 100 while the optimal β values are between 0.01 and 0.5.

7.2 Clustering small networks

The small test networks were clustered using the optimal hyperparameter values in terms of the modularity of the clustering, which were found by exhaustive search over the hyperparameter α and β values (see, Table 7.1). The results are shown in table 7.2. This table shows the modularity, number of clusters and hyperparameter values for the optimal values of the M0 algorithm. The modularity for the M0 algorithm has been obtained by assigning each node into the component it belongs to with the highest probability.

Table 7.2 also displays, for comparison, the modularities obtained with some popular community algorithms. In the table, GN is the community algorithm by Girvan and Newman (Girvan and Newman, 2002), EO refers to the Extremal Optimization algorithm (Duch and Arenas, 2005), CNM is the fast community algorithm by Clauset, Newman and Moore (Clauset et al., 2004), and N06 is a spectral algorithm presented in Newman (2006). The comparison results are based on those published in Duch and Arenas (2005), Newman (2004, 2006).

As can be seen from the figure, the M0 algorithm seems to perform reasonably well compared to the other algorithms, even though it does not implicitly optimize modularity. For the small networks, only the N06 algorithm performs better than M0. For the Email network, the result is slightly worse than that obtained using EO and N06 algorithms and for the PGP network the result is clearly worse than that obtained using the other algorithms. The bad result with the PGP network could be related to

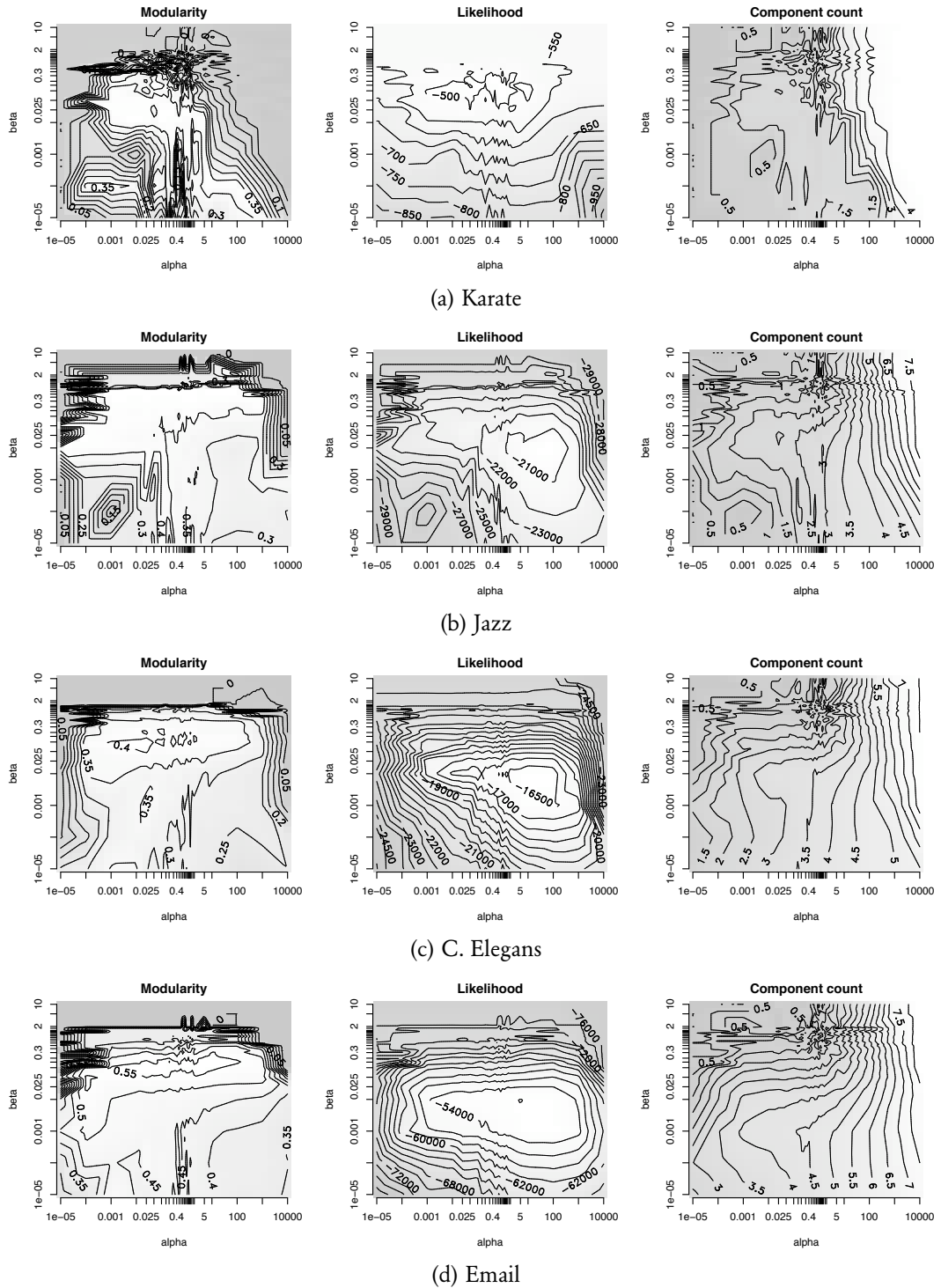


Figure 7.2. Results from running the clustering with a range of hyperparameter values. For each network, the figure on the left shows the modularity of the network split, the figure on the middle displays the likelihood of the result, and the rightmost figure shows the amount of components in the solution on logarithmic scale.

the small clustering coefficient of the network. The results for the Physicist network outperform other approaches.

The strength of the approach compared with divisive methods, such as GN, and agglomerative approaches, such as CNM, is that not only the most likely component for each node is given, but also the probabilities of all the other cluster assignments for each node. In addition, the clustering outcome can be also retrieved on the level of edges, so that each edge is given an explanation in terms of the components, which could have generated it. The main challenge with the M0 algorithm is the choice of hyperparameter values.

7.2.1 Networks with known community structure

To validate the results of a clustering algorithm, it would be ideal to have a network for which the correct division into groups is known. Yet, there are only a small number of such networks commonly used in the literature. The clustering results with the M0 algorithm for two such networks, the Karate network and the Football network, are presented in detail below.

Karate network

The clustering of the nodes in the Karate network is shown in Figure 7.3. The hyperparameter values $\alpha = 0.025$ and $\beta = 0.05$ were obtained by exhaustive search, as presented in Section 7.1.

The algorithm was able to determine almost perfectly the correct division of the network into two groups. However, with these hyperparameter values, the algorithm further subdivided one of the clusters into two smaller clusters, shown with green and blue colors. The cluster assignment of the nodes on the borders of the clusters tends to be fuzzy while the more remote nodes seem to belong almost solely to a single cluster.

In addition, there are some nuisance clusters such as the ones shown in the figure with yellow and pink colors. It would seem that some type of label switching has occurred, because the yellow cluster matches almost exactly the red cluster. This is a problem, because if no label switching would have occurred, node 9 in the middle of the figure would have been assigned correctly because together the yellow and red clusters would have won the majority voting.

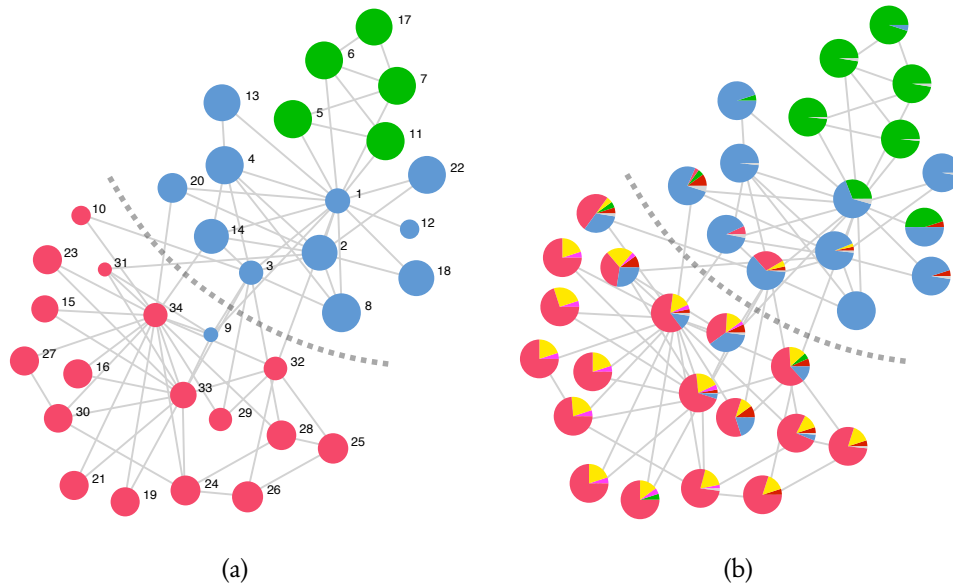


Figure 7.3. *Two visualizations of the clustering result for the Karate network. On the left side, node size represents the certainty of the cluster assignment and node color represents the clustering. On the right side, for each node, a pie diagram is shown, which illustrates the probabilities of the various clusters or components for each node. The dashed line in both pictures shows the “correct” cluster division based on the splitting of the karate club. $\alpha = 0.025$ and $\beta = 0.05$.*

Football network

The clustering results for the Football network are presented in Figure 7.4. As for the other small networks, the hyperparameter values $\alpha = 0.01$ and $\beta = 0.025$ were obtained by exhaustive search (see, Section 7.1). As can be seen from the figure, the algorithm found most conferences correctly, and even the number of clusters is almost correct. However, some of the nodes were clustered into wrong conferences, especially in the top right corner of the figure. Furthermore, two conferences on the right side of the figure were incorrectly combined into one.

The node placement and node ids in Figure 7.4 are similar to those used by Clauset et al. (2006). It is interesting to note that the clustering algorithm by Clauset et al. (2006), which attempts to model statistically the hierarchical clustering of a network, makes similar errors in classification as the M0 algorithm. Both of the algorithms fail to identify the correct clusters in the top right corner. However, the algorithm by Clauset et al. (2006) is able to divide the cluster on the right side of the figure (shown with yellow color) correctly into two parts.

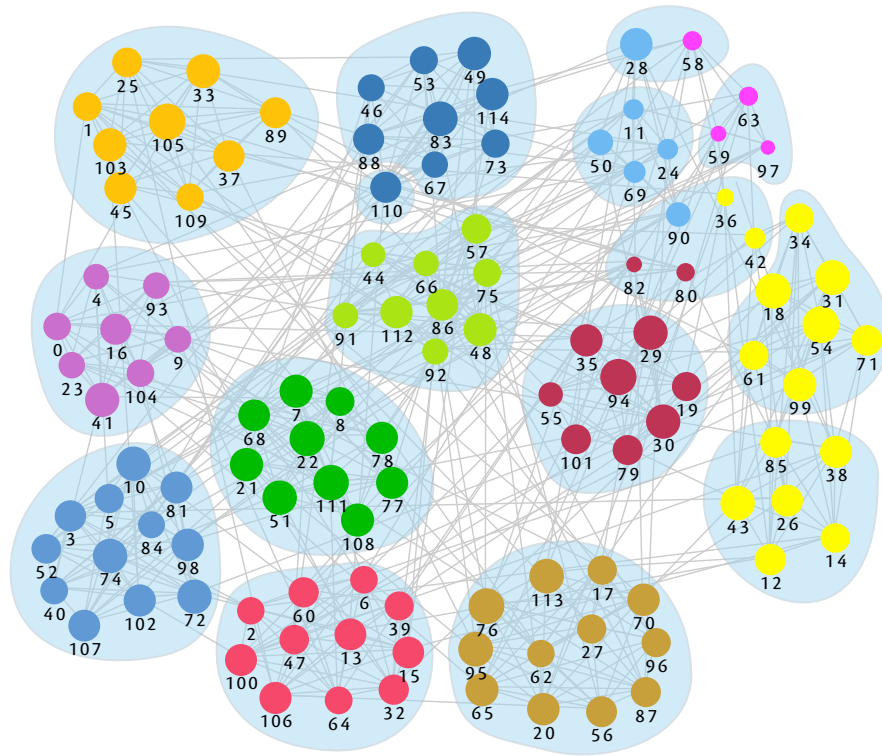


Figure 7.4. Clustering result for the Football network. $\alpha = 0.01$ and $\beta = 0.025$. Colored areas show the borders of the conferences (that is, the correct clustering), while the node colors represent the cluster assignments obtained using the MO algorithm. The size of the nodes illustrates the certainty of the cluster assignments.

7.2.2 Convergence of the small networks

Figure 7.5 shows the convergence of some of the small test networks in terms of both the network likelihood given the model and modularity. For the small networks, these measures tend to jump from one phase to another in one quick step (Football and Jazz networks). The likelihood and modularity values have been calculated for individual iteration samples only and give thus just point estimates of the real likelihood. For the smallest networks, this makes it hard to assess the exact convergence.

For the most part, the modularity of the network increases monotonically with the iterations. Only in the PGP and Jazz networks there is some later decrease in the modularity. In all the networks, a quite good clustering in terms of modularity and likelihood is found quite quickly (less than 1000 iterations). However, for many of the networks, the likelihood improves slowly for a number of iterations after the initial quick convergence. For example, for the Physicists network the likelihood reaches a plateau after 25 000 iterations.

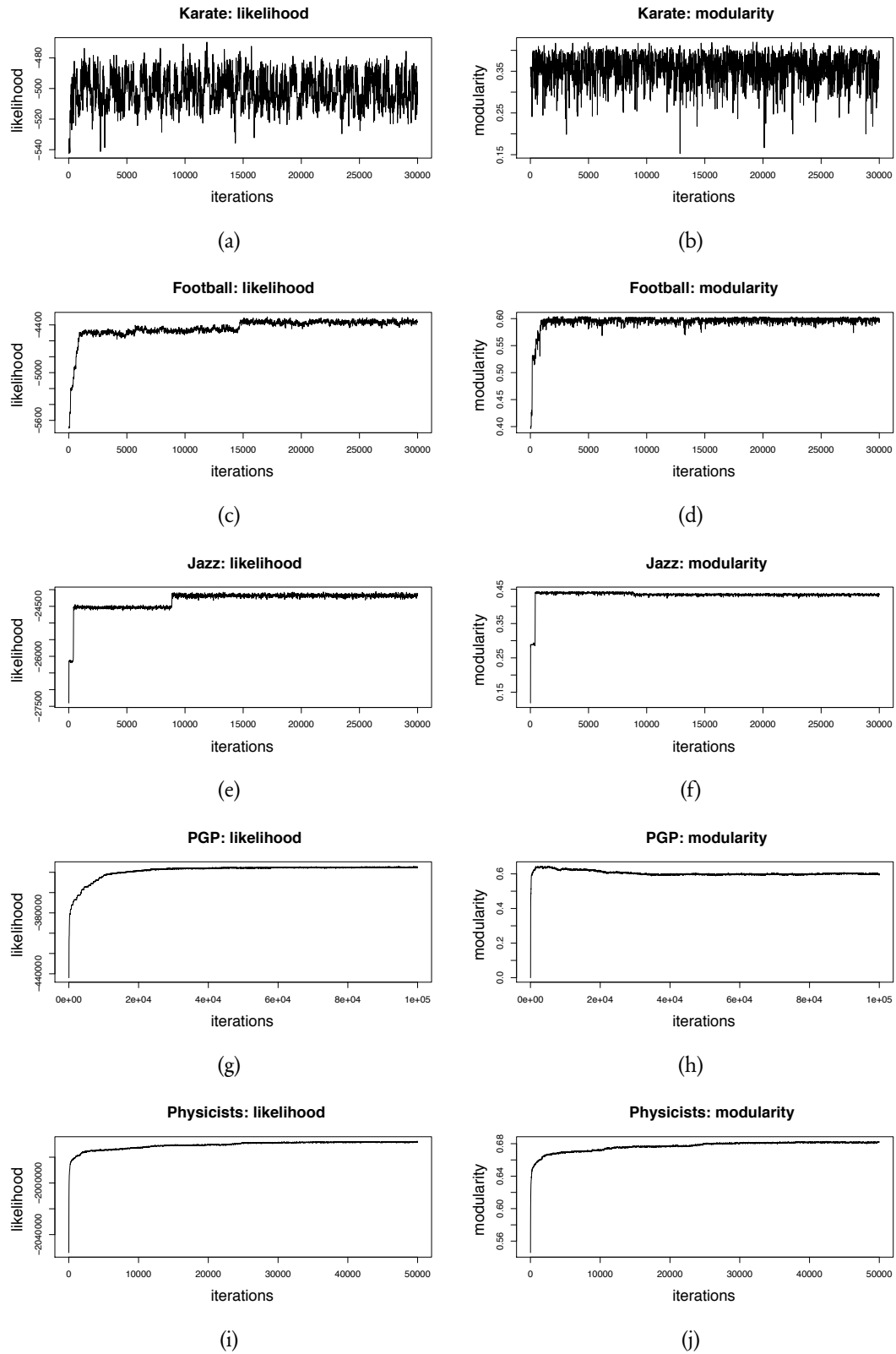


Figure 7.5. Convergence of the small networks. In the figures, on the left side is plotted the sample likelihood for the network for every iteration and on the right side the sample modularity. Because the measures were calculated without averaging over multiple samples, the values for especially the smaller networks fluctuate considerably.

7.3 Clustering Last.fm friendship network

The full Last.fm network and the network of the US users were clustered using the M0 algorithm. In this section, the clustering results for both of the networks are compared to the music tags of the users. In addition, the Danish users of Last.fm were clustered for the sake of visualizing how the actual connections between the users look like.

7.3.1 Full Last.fm network

The clustering results for large networks cannot be visualized directly in form of a network, as is the case with small networks. Thus, to get an understanding on the clustering results, some type of proxy information has to be used. The aim is to compare, how well the clusters correspond to other traits of the Last.fm users. Thus, one solution is to create a traits \times clusters matrix. The elements of the matrix can be sorted with hierarchical clustering methods so that traits occurring often in the same clusters are presented close to each other. When colors are used to represent the counts in each cell of the matrix, this type of visualization is called a *heat map*.

For each user, the tag that corresponds best to his music taste has been observed. In the case of hard clustering, for each user, one could simply increment the value of the cell $\{tag, cluster\}$ by one. However, in fuzzy clustering, the user is a member of multiple clusters, and for each of the clusters a user belongs to, the cell $\{tag, cluster_i\}$ has to be incremented proportionally to the degree the user belongs to the cluster. For example, if a user listens only to music tagged with *japanese*, and belongs to cluster A with a degree of 0.5 and to cluster B with a degree of 0.5, then for that user, the values in both the cell $\{japanese, cluster A\}$ and $\{japanese, cluster B\}$ are incremented with 0.5.

Figure 7.6 shows the clustering result of the main component of the Last.fm network data as a heat map. Burn-in period for the full Last.fm network was 19000 iterations to ensure the convergence of the sampling process. After the burn-in period, 20 samples were taken at intervals of 50 iterations. More samples could have been used as well. In the Figure 7.6, smallest components with less than 20000 members are not shown. The figure shows that there is a correlation between the music tags and the clusters, in which the nodes have been assigned. This can be seen as long blue rows of tags, which correspond strongly to certain components. No tags seem to be particularly likely in the last cluster.

Tags corresponding to nationalities are common in some of the clusters. The first cluster from the left (A) contains not only listeners of *alternative rock* but also those

who listen music tagged with *japanese*, *russian*, and *polish* tags. The second cluster (B) corresponds to those who listen to metal music. The third cluster (C) contains both the electronic music styles, and the tag *swedish*. Tags in the fourth cluster (D) are related to *alternative*, and *hard core* music styles. The only strong tag in the last cluster is *trance*.

Comparing the clustering with the tags of the users has its problems, because the clusters could also be explained by nationalities of the users, which in turn affect the music tastes. This effect might also explain the tags corresponding to the nationalities, observed above. The nationalities of the users can be seen in Figure 7.7, where the labels A-D used for the components are the same as in the Figure 7.6. Correlation between nationalities and clusters seems to be even stronger than between tags and clusters. An interesting detail is that seemingly remote areas are put into the same clusters, such as Finland and China or Estonia and Argentina.

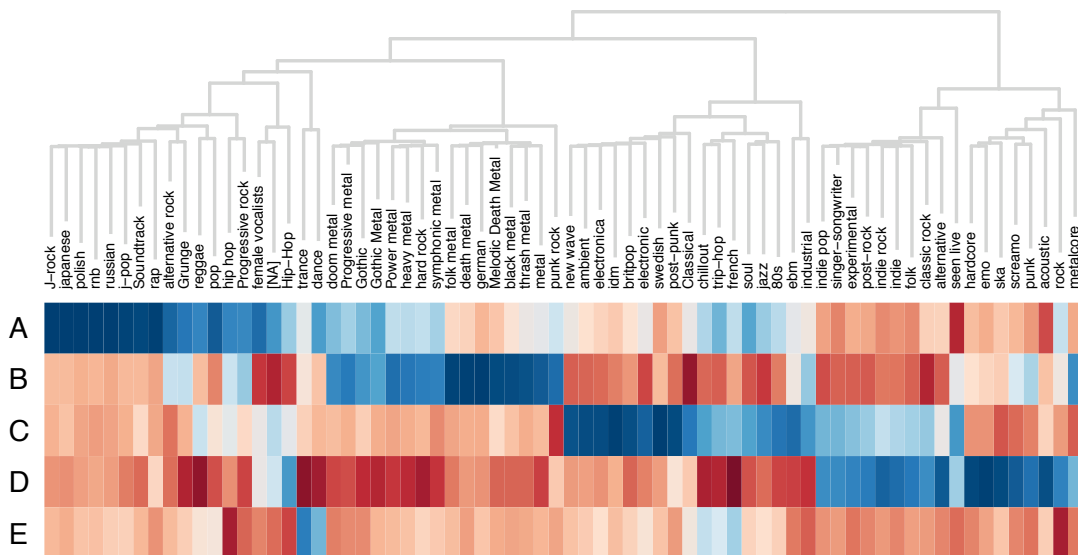


Figure 7.6. *The whole Last.fm main component clustered into five clusters and plotted together with the tag counts in each group. In the figure, each column corresponds to a music tag given by the users of Last.fm. Only the 80 most popular tags are shown. The color of a cell in the grid depicts, how likely it is for a user who listens to that tag (column) to belong to the particular cluster (row). Blue means that more than expected users belong to that particular cluster, while cells that are red correspond to clusters which are unpopular among the users who listen to that particular tag. $\alpha = 0.3$ and $\beta = 0.3$.*

7.3.2 Last.fm USA network

Because geographical location would seem to affect strongly the Last.fm social network structure, the effects of homophily and proximity are impossible to analyze

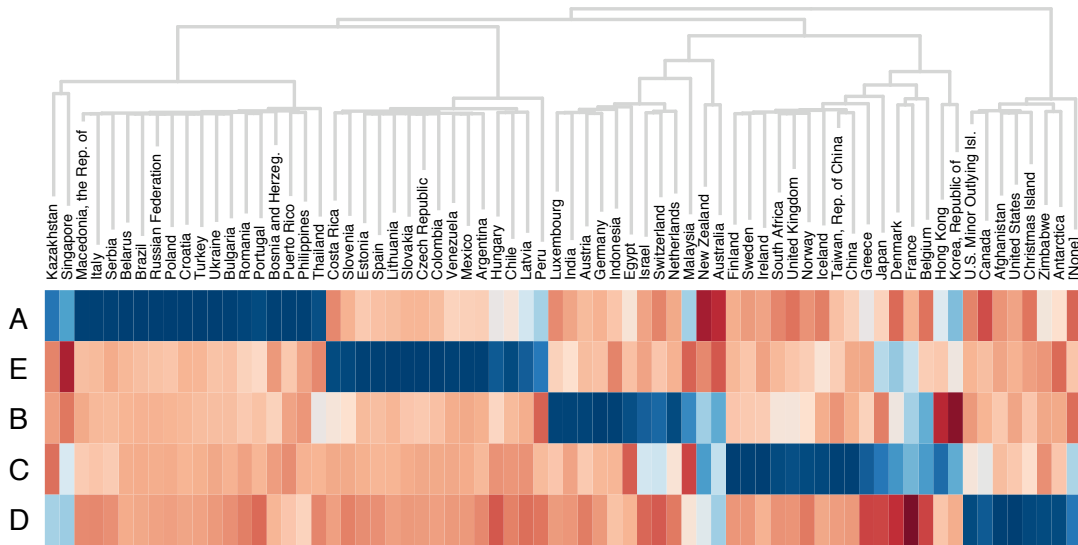


Figure 7.7. The whole Last.fm main component clustered into five groups (rows) with the countries of the group members (columns). $\alpha = 0.3$ and $\beta = 0.3$.

independently for the network clustering. One alternative solution to diminish the effect of geography and nationality is to concentrate on a smaller population of the network, which does not divide into the self-evident clusters based on countries.

The challenges caused by geography are one motivation to study the network formed by Last.fm users from the USA separately. However, even this does not eliminate the problem of having the geography affect the clustering, since even in the USA, there are many strong, local areas, and musical styles are often more popular in some parts of the country than in others.

In a similar fashion to the full network, Figure 7.8 displays the clustering of the Last.fm users from United States as a heat map. The burn-in period was 49000 iterations after which 20 samples were taken with 50 iterations interval.

New hyperparameter values were chosen for the Last.fm USA network, because the values depend on the size of the network. Slightly smaller hyperparameter values were used ($\alpha = 0.2$ and $\beta = 0.2$) than with the full Last.fm network, which as a side effect led to finding more clusters (8).

The users seem to partition well into different clusters based on their music taste. Cluster A contains those who listen to pop and country music, as well as listeners of britpop. Cluster B consists of listeners of electronic music, jazz and British alternative rock music from the 80's (*new wave* and *80s*). Cluster C consists of experimental rock music, with tags such as *indie pop*, *folk* and *experimental*. Many of the tags are however shared with the users from cluster B. Cluster D is strongly described by the listening of

Japanese and metal music. Cluster E has a strong community of listeners of Christian music. Cluster F contains music styles related to hip hop and hardcore punk while the only tags that rise above the average in cluster G are punk and ska. The cluster H cannot be easily explained using any of the tags.

In the Figure 7.8 some clusters are quite clearly bounded, such as cluster D with the listeners of Japanese and metal music, while the music tastes of others overlap, such as those belonging to clusters B (electronica, jazz) and C (indie pop, folk). Even though the cluster H does not strongly explain any music tastes, the users belonging to it seem to favor more metal and hardcore music than rock or indie pop.

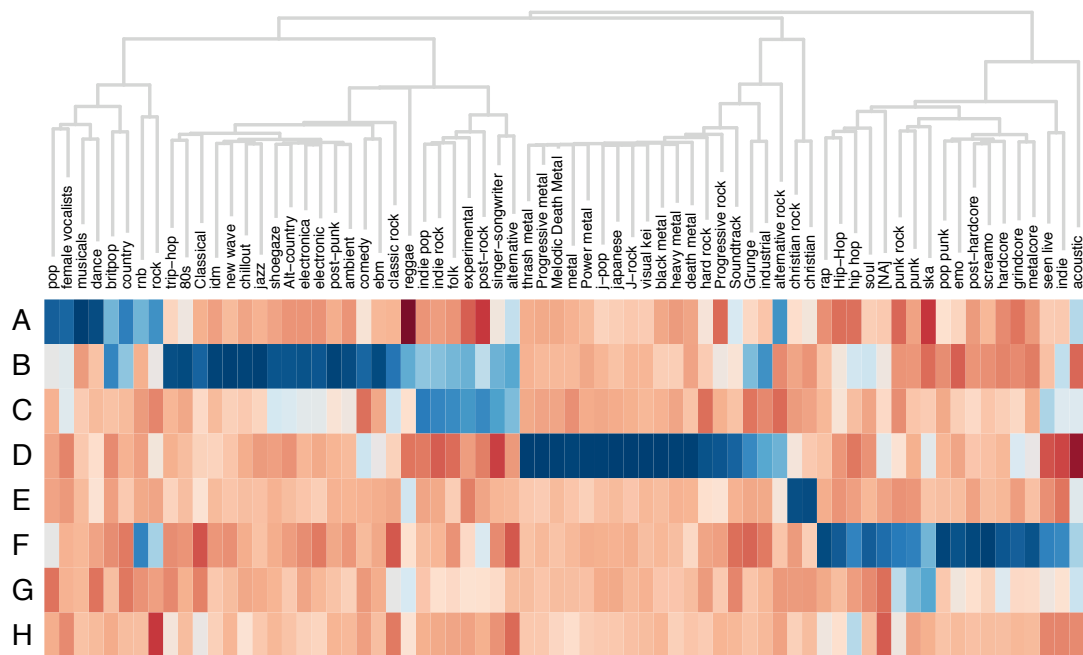


Figure 7.8. *The Last.fm users from United States belonging to the main component clustered into eight groups with their tags. $\alpha = 0.2$ and $\beta = 0.2$.*

7.3.3 Likely and unlikely tags

Table 7.3 shows the tags that correspond best and worst to the components (i.e. clusters) found in the Last.fm US network. It displays in more detail what tags are more likely and unlikely in each of the components than on average.

The tables have been obtained by calculating the occurrence counts of each of the tags in every component. Then, for every tag, it has been calculated how much more or less likely the tag is in each component than it would be based on marginal probabilities. Finally, a binomial test was performed with $p = 0.05$ to remove from the tables the tags which do not differ notably from the expected probabilities. The numbers in the

(a)		(b)	
Cluster A		Cluster B	
juggalo	1.36	shoegaze	1.35
pop	1.34	Alt-country	1.24
musicals	1.32	post-punk	1.22
Canadian	1.09	idm	1.15
female vocalists	0.85	new wave	1.15
shoegaze	-1.76	post-hardcore	-1.62
Sludge	-1.89	screamo	-1.79
black metal	-1.98	pop punk	-3.16

(c)		(d)		(e)	
Cluster C		Cluster D		Cluster E	
indie	0.46	j-pop	1.69	christian	1.53
post-rock	0.30	visual kei	1.68	podcast	1.01
folk	0.22	black metal	1.56	trance	0.87
Stoner Rock	-1.84	death metal	1.53	new age	0.70
visual kei	-1.86	japanese	1.53	Grunge	0.43
j-pop	-2.08	indie rock	-1.19	visual kei	-1.46
		post-punk	-1.21	shoegaze	-1.54
		psychedelic	-1.41	Sludge	-1.68

(f)		(g)		(h)	
Cluster F		Cluster G		Cluster H	
rnb	1.33	Jam	1.35	latin	1.13
screamo	1.21	ska	0.89	chinese	1.05
pop punk	1.15	hardcore	0.47	psytrance	0.70
post-hardcore	1.07	punk	0.40	Korean	0.68
hardcore	1.05	indie	0.11	trance	0.51
post-punk	-1.66	rnb	-1.36	Alt-country	-0.58
Korean	-2.25	visual kei	-1.43	synthpop	-1.54
psytrance	-2.48	j-pop	-2.28	juggalo	-1.62

Table 7.3. *The most likely and unlikely tags for each of the components in the Last.fm United States network.*

table are the logarithms of the ratio between the observed and expected probabilities of the tags.

The five most likely and three least likely tags are displayed for each of the clusters. For cluster C, only the three most likely tags are displayed, because the rest of the tags which occurred often in the cluster were eliminated in the binomial test.

Interest of Japanese music, represented with the tags *j-pop*, *visual kei* and *japanese* is strong in cluster D. However, in other clusters, such as C, G and E the japanese music styles seem to be quite unlikely. Another tag with similar concentration into one cluster is *shoegaze*. It is popular cluster B, but not much used in clusters A and E. Even other opposition pairs exist: *juggalo* between cluster A and H, *black metal* between A and D and *rnb* between clusters F and G.

Although the different clusters seem to differ strongly in terms of tags used, some tags are likely in more than one components. These are *trance*, which is found in both the clusters H and E, and *hardcore* that is listened by members of cluster F and G.

To conclude, by looking at the tags that differ from the average use of tags by a wide margin, we note that quite different types of music listening habits are present in the clusters, and some clusters, like C and D, seem to be opposites in terms of music interests. This would seem to indicate that the algorithm is able to group users based on their interests. Many of the separations are quite strong: in every cluster there are some tags that are 3-5 times more likely in that cluster than on average.

7.3.4 Convergence of the clustering for Last.fm networks

Figure 7.9 shows plots of the modularity and likelihood of samples for the Last.fm and Last.fm United States networks. Surprisingly, the sampling for larger network converges clearly faster than that for the smaller network of users from the United States only. This might be due to the different hyperparameters and the larger number of components found for the US network.

7.3.5 Close-up view of the Last.fm Denmark network

In Figure 7.10 a close-up view of the network with 2374 Danish Last.fm users is shown. Each node represents one person. The color of a node represents the most probable cluster for the person while the label of a node is the tag that is used most commonly to describe the bands listened by the user. Node size is determined by the certainty of the clustering result. The larger the node, the larger the certainty there is about its group.

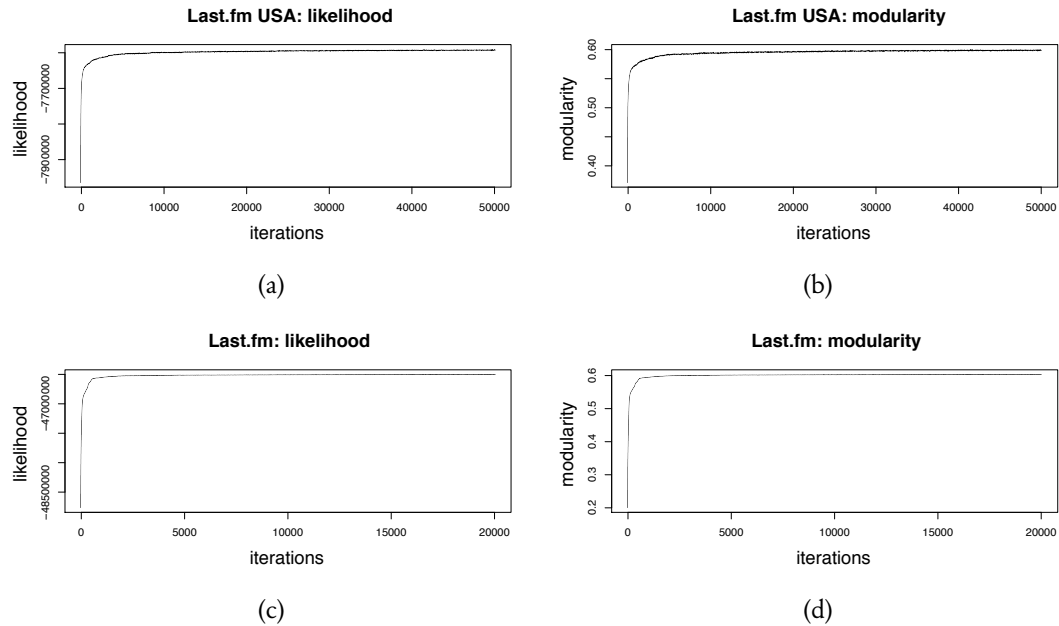


Figure 7.9. *Convergence of the full Last.fm network (top) and of the United States subset of Last.fm network (bottom). Note that the scale of the horizontal axis is different in the figures because of the differing amount of iterations.*

The first thing that can be seen from Figure 7.10 is that making a good visualization of even a quite small network like this is difficult because of the small-world structure, which connects remote nodes to each other. However, even in this type of visualization, some structures can be seen. One is the cluster in the bottom right corner of the figure, which consists of girls listening to gothic music (cluster with pink color) and boys that are friends with these girls (cluster of five nodes with dark purple color). The algorithm has effectively separated these two groups into separate components.

Another property revealed by the visualization is that with small hyperparameter values the components found seem to describe quite local structures. This would mean that with these parameters, the algorithm works essentially in a similar way as community algorithms.

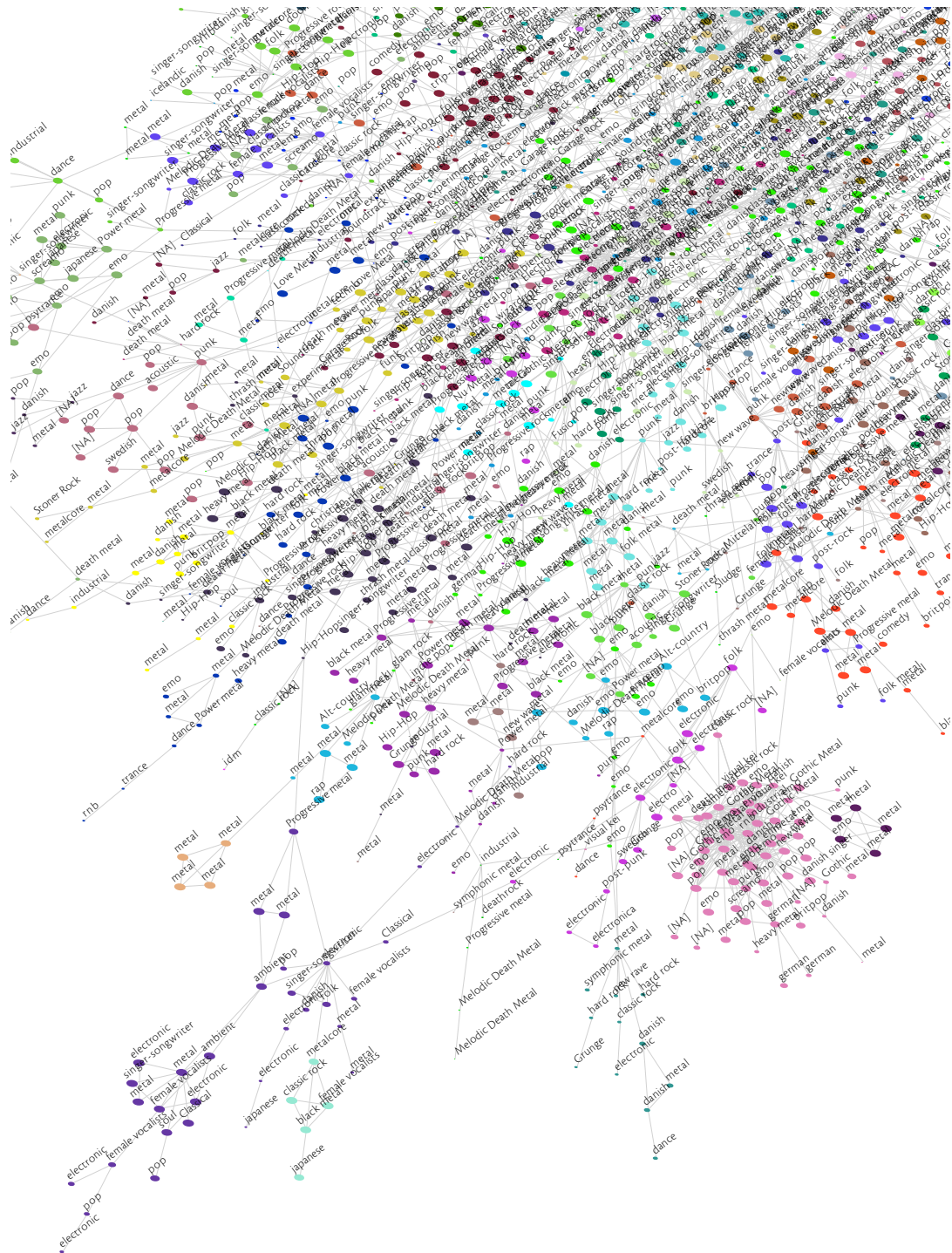


Figure 7.10. A close-up view of the results from clustering the Danish Last.fm users. In the clustering, $\alpha = 0.001$ and $\beta = 0.01$.

Chapter 8

Conclusions and discussion

This chapter begins by evaluating the results of the work based on the research problem and questions presented in Chapter 1. Then, ideas for further research are presented related to further validation of the algorithm, ways to improve the model, and approaches to enhancing the performance of the implementation.

8.1 Evaluation

In this thesis, clustering of nodes in networks has been studied. A clustering algorithm based on the generative M0 model has been compared to other algorithms based on the modularity measure, and results of the clustering have been evaluated with both small networks and a large friendship network collected from the Last.fm online service. In addition, the algorithm convergence and the choice of optimal hyperparameter values have been discussed.

The aim of this thesis has been to address the research problem:

By using only friendship network topology, can individuals be clustered into groups that differ in traits such as interest, language and gender?

In general, it seems, that friendship topology does separate users into distinct groups and the interests of the users in different groups differ in a number of ways. However, it is difficult to assess, whether the clusters found represent some globally optimal clustering, or whether it is just one possible division of the network among many equally good explanations.

In addition to the research problem, also five subquestions were posed. Below, the subquestions are discussed in detail. Both previous research and the empirical findings are used to answer the questions.

1. *Based on previous research, is it plausible that friendships can be used to predict traits of the individuals?*

Research in the field of sociology clearly supports the theory that similar people tend to interact with each other. However, also other factors guide friendship formation, such as people introducing friends to each other and geographical location.

2. *Can the M0 method be used in finding clusters from a friendship network?*

Tests with both small and large networks show that the M0 approach is effective at finding meaningful clusters in many kinds of networks. Yet, the usefulness of an algorithm depends not only on the result it gives but also on how easy it is to use, including the amount of manual tuning it requires. The selection of hyperparameter values for the algorithm is often difficult, which reduces its usefulness. Thus, ways to make the selection easier will clearly need some further research.

3. *Based on the clustering, is it possible to make some conclusions about the traits affecting the friendships of the individuals?*

In the Last.fm network, there was a strong correlation between the clustering results and the musical interests. However, it is possible that the shared musical interests depend on the geographical locations of the users, since even music taste can be local. Structures that depict geographical information are typically not as interesting as other traits, since they can be estimated directly from information on users, without the need to use friendship networks.

Thus, it is difficult to evaluate, whether it is possible to extract information about traits, such as values or interests, from a friendship network, or if the network just illustrates the geographical locations of its members. Yet, the results from the Last.fm USA network give a suggestion that friendships could predict also other traits than the geographical location.

4. *Does the algorithm find local clusters in a network (communities) or more diffuse latent component structures (traits)?*

It seems that the algorithm can find both small, local structures and more diffuse components. By changing the hyperparameter values, the algorithm can be used flexibly to find both clusters of different sizes, and clusters which have different characteristics. The number of clusters scales from a few up to thousands. When the number is large, the clusters correspond to the local neighborhoods of the nodes.

However, the effects of the various hyperparameter combinations are not fully clear. A question remains, whether the more diffuse components are just fuzzy

clusters or whether they can be made to reflect some real global properties of the nodes.

5. *How well does the method compare with other approaches in terms of clustering results and algorithm speed?*

Comparison of the results with community algorithms that optimize modularity are promising. When good hyperparameter values are used and in a favorable setting, the algorithm is an effective community algorithm. For two of the test networks, it outperformed all the hierarchical clustering algorithms, with which it was compared.

The M0 algorithm is not optimal for all clustering tasks, since it is slower than the fastest agglomerative methods. However, the strength of the method is the fuzzy assignment of nodes to clusters, where a node can belong to a number of clusters simultaneously. This allows assessing the confidence of the cluster assignments and detection of different roles in the network, such as the nodes which are central in the clusters and those that are members of multiple clusters.

8.2 Applicability of the results

Although the research on the M0 algorithm is still in its early phases, the algorithm and this study may be of interest to those working with networks, such as sociologists, physicists and computer scientists. The algorithm is easy to implement, and can be used to study various types of structural data. The model can be incorporated into larger network analysis systems as a method for reducing the dimensionality of the data. The same model can also be used for other purposes than simple clustering of nodes, for example in the prediction of future links or in the estimation of missing labels in the data. Moreover, by extending the model, many possibilities emerge for analyzing various types of sparse relational data sets, including web forums and citation networks.

For the company, Xtract Ltd, this study provides valuable insight on the properties of the M0 algorithm and its applicability for analyzing huge networks. The optimized implementation and tests performed on networks of different sizes should make it relatively easy to get the algorithm into production use. In addition, during this study, information was gained on the sociological basis of network analysis, which is applicable both in understanding customer problems and in devising new methods for analyzing social networks.

8.3 Future work

Some promising results obtained using the M0 model have been presented in this work. However, the research on the M0 model and on latent component models for networks in general is still in its first stages. As can be seen from the answers in the previous section, open questions remain related to the use and functioning of the algorithm. There is also a wide range of directions for further research. These can be divided into four categories: *validation*, *improvement*, *performance optimization*, and *further possibilities*.

8.3.1 Validation

The algorithm performance could be further validated by making more extensive comparison of the results with other algorithms. Furthermore, although the algorithm was shown to find clusters that differ in the traits of their members, a more detailed analysis could reveal how well the traits can be predicted based on the clustering results.

The understanding of the properties of the model are still limited. Further tests need to be performed to determine if label switching occurs when running the model for long times and with dense networks. Also, the effect of increasing the number of samples that are taken during the sampling process needs to be tested.

Moreover, the validity of the generative model could be evaluated by generating networks using the M0 model with parameters inferred from real networks. The characteristics of these networks, such as degree distribution and clustering coefficient, could be compared to those of the original network, to assess how well the model can capture the patterns in the network.

8.3.2 Improvement

The generative model could be improved to incorporate more information about network structure, such as weights on edges, directional edges and node attributes. These might improve the quality of the clustering results and the predictive power of the algorithm.

The selection of hyperparameters is currently difficult. Some effort should be put into making the use of the hyperparameters easier, such as adding an uninformative prior to the hyperparameters. For this, ideas could possibly be borrowed from other mixture models with Dirichlet priors.

The Dirichlet distributions used in the algorithm provide a flexible, non-informative prior for the model. Nevertheless, some properties, such as correlation between the

components, cannot be easily explained using Dirichlet distributions. One approach to overcome this problem would be to try to replace the Dirichlet distribution with a logistic normal distribution in a similar fashion to Blei and Lafferty (2005) and Ahmed and Xing (2007).

8.3.3 Performance optimization

It should be possible to improve the performance of the algorithm by distributing the Gibbs sampling process on multiple CPUs (central processing units). Moreover, it might be possible to improve the algorithm speed with other adjustments, such as implementing some of the core procedures with a lower level language, for example C. Maybe even larger performance improvements could be achieved by further evaluation of EM and Variational approaches to parameter estimation.

8.3.4 Further possibilities

All in all, in addition to the testing and optimizing of the M0 algorithm, a lot of interesting topics related to studying networks have come up during this work. Because the field of study is constantly evolving, behind just about every corner there is an opportunity for exciting research.

There are many unexplored possibilities for using methods from the network sciences in information systems. In one way or another, intelligent search engines and social networking services will most certainly use networks for recommending products based on the behavior of other users, and for visualizing to users the relationships between people or pieces of information.

Network algorithms can be incorporated into the analysis of all types of interaction, such as marketing research, fraud detection and analysis of biological structures. In article databases, networks made from citations can be used to join redundant items, and to find articles that are important in a field. Moreover, fast algorithms devised for studying networks can be used even in such surprising fields as labeling structures in computer vision, semantic web research, and even game theory.

Nevertheless, more knowledge is still needed on how networks evolve, what networks from various sources have in common, and what tools are useful in the specific domains. Understanding the actual problems, and the subjects that are studied, serves as a natural starting point for developing better methods and algorithms for network analysis.

Bibliography

- Adamic, L. and Adar, E., 2003. Friends and neighbors on the web. *Social Networks*, 25(3):211–230.
- Adamic, L. A., Buyukkokten, O., and Adar, E., 2003. A social network caught on the Web. *First Monday*, 8(6). Available at: http://firstmonday.org/issues/issue8_6/adamic/index.html.
- Ahmed, A. and Xing, E. P., 2007. On tight approximate inference of logistic normal admixture model. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS 2007)*. Omnipress, Madison, USA. Proceedings on CD.
- Airodi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P., 2007. Mixed membership stochastic blockmodels. *ArXiv Physics e-prints*. 0705.4485v1.
- Albert, R. and Barabasi, A.-L., 2002. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97.
- Allen, T. J., 1977. *Managing the Flow of Technology*. MIT Press, Cambridge, USA.
- Amaral, L. A. N., Scala, A., Barthélémy, M., and Stanley, H. E., 2000. Classes of behavior of small-world networks. *Proceedings of the National Academy of Sciences USA (PNAS)*, 97(21):11149–11152.
- Antoniak, C. E., 1974. Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. *The Annals of Statistics*, 2(6):1152–1174.
- Arenas, A., 2007. Network Data Sets. Available at: <http://deim.urv.cat/~aarenas/data/welcome.htm>, accessed 23 August 2007.
- AudioScrobbler, 2007. AudioScrobbler web services. Audioscrobbler Ltd. Available at: <http://www.audioscrobbler.net/data/webservices/>, accessed 22 August 2007.
- Barabási, A.-L. and Albert, R., 1999. Emergence of scaling in random networks. *Science*, 286(5439):509–512.
- Barber, D., 2006. Machine Learning: A Probabilistic Approach. Available at: http://www.idiap.ch/~barber/mlgm_epfl_book.pdf, draft version, accessed 22 August 2007.

- Berger-Wolf, T. Y. and Saia, J., 2006. A framework for analysis of dynamic social networks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'06)*, pages 523–528. ACM Press, New York, USA.
- Bergstra, J., 2006. *Algorithms for Classifying Recorded Music by Genre*. Master's thesis, Department of Computer Science and Operations Research, University of Montreal, Montreal, Canada.
- Berkhin, P., 2002. Survey Of Clustering Data Mining Techniques. Technical report, Accrue Software, San Jose, USA.
- Berthold, M. and Hand, D. J., 1999. *Intelligent Data Analysis: An Introduction*. Springer-Verlag, Berlin, Germany.
- Bishop, C. M., 2006. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, New York, USA.
- Blackwell, D. and MacQueen, J. B., 1973. Ferguson distributions via Pólya urn schemes. *Annals of Statistics*, 1:353–355.
- Blei, D., Ng, A., and Jordan, M., 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.
- Blei, D. M. and Lafferty, J. D., 2005. Correlated Topic Models. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 18. MIT Press.
- Blue, J. L., Beichl, I., and Sullivan, F., 1995. Faster Monte Carlo simulations. *Physical Review E*, 51(2):867–868.
- Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D.-U., 2006. Complex Networks: Structure and Dynamics. *Physics Reports*, 424(4-5):175–308.
- Bolstad, W. M., 2004. *Introduction to Bayesian Statistics*. John Wiley & Sons, New York, USA.
- Boyd, D., 2006. Friends, Friendsters, and Top 8: Writing community into being on social network sites. *First Monday*, 11(12). Available at: http://firstmonday.org/issues/issue11_12/boyd/index.html.
- Brandes, U., Delling, D., Gaertler, M., Görke, R., Hoefler, M., Nikoloski, Z., and Wagner, D., 2006. Maximizing Modularity is hard. *ArXiv Physics e-prints*. physics/0608255v2.
- Breiger, R., Boorman, S., and Arabie, P., 1975. An algorithm for clustering relational data, with applications to social network analysis and comparison with multi-dimensional scaling. *Journal of Mathematical Psychology*, 12(3):328–383.
- Brown, J. J. and Reingen, P. H., 1987. Social Ties and Word-of-Mouth Referral Behavior. *The Journal of Consumer Research*, 14(3):350–362.

- Bruyn, A. D. and Lilien, G. L., 2004. A Multi-Stage Model of Word of Mouth through Electronic Referrals. *eBRC Research Paper Series (ref. 2004-02)*. Available at: http://www.debruyn.info/research/papers/debruynlilien2004_referrals.pdf, working paper.
- Buntine, W. L., 1994. Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, 2:159–225.
- Buntine, W. L., 2002. Variational Extensions to EM and Multinomial PCA. In T. Elomaa, H. Mannila, and H. Toivonen, editors, *Proceedings of the 13th European Conference on Machine Learning (ECML'02)*, volume 2430 of *Lecture Notes in Computer Science*, pages 23–34. Springer, London, UK.
- Byrne, D. E., 1971. *The Attraction Paradigm*. Academic Press, New York, USA.
- Cairncross, F., 1997. *The Death of Distance*. Harvard Business School Press, Boston, USA.
- Casella, G. and Robert, C. P., 1996. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94.
- Castellano, C., Cecconi, F., Loreto, V., Parisi, D., and Radicchi, F., 2004. Self-contained algorithms to detect communities in networks. *The European Physical Journal B*, 38:311–319.
- CDG, 2007. Network Resources - Data Sets. Collective Dynamics Group of Columbia University, New York, USA. Available at: <http://cdg.columbia.edu/cdg/datasets>, accessed 23 August 2007.
- Cern Jet, 2007. Cern Jet Library. European Organization for Nuclear Research (CERN). Available at: <http://dsd.lbl.gov/~hoschek/colt/>, accessed 22 August 2007.
- Chapelle, O., Schölkopf, B., and Zien, A., editors, 2006. *Semi-Supervised Learning*. MIT Press, Cambridge, USA.
- Clauset, A., Newman, M. E. J., and Moore, C., 2004. Finding community structure in very large networks. *Physical Review E*, 70(6):066111.
- Clauset, A., Moore, C., and Newman, M. E. J., 2006. Structural Inference of Hierarchies in Networks. *ArXiv Physics e-prints*. physics/0610051v1.
- Clauset, A., Shalizi, C. R., and Newman, M. E. J., 2007. Power-law distributions in empirical data. *ArXiv Physics e-prints*. 0706.1062v1.
- Costa, L., Rodrigues, R. A., Travieso, G., and Boas, P. R. V., 2007. Characterization of complex networks: A survey of measurements. *Advances in Physics*, 56(1):167–242.
- Creative Commons, 2007. Creative Commons: Frequently Asked Questions. Creative Commons. Available at: <http://wiki.creativecommons.org/FAQ>, accessed 22 August 2007.

- Culhane, A. C., Thioulouse, J., Perriere, G., and Higgins, D. G., 2005. MADE4: an R package for multivariate analysis of gene expression data. *Bioinformatics*, 21(11):2789–2790.
- Cyram, 2005. NetMiner v.2.6's Analysis Measures, A List of Reference. Cyram Co. Available at: <http://www.netminer.com/NetMiner/pdf/ReferencePart1.pdf>, accessed 23 August 2007.
- Danon, L., Diaz-Guilera, A., and Arenas, A., 2006. The Effect of size heterogeneity on community identification in complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 11:11010.
- Daudin, J. J., Picard, F., and Robin, S., 2007. A mixture model for random graphs: A Variational Approach. Technical Report 4, Statistics for systems biology group, INRA, Jouy-en-Josas, France.
- Dempster, A. P., Laird, N. M., and Rubin, D. B., 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38.
- Dietterich, T. G., 2003. Machine Learning. In L. Nadel, editor, *Nature Encyclopedia of Cognitive Science*. Macmillan, London, UK.
- Dorogovtsev, S. N. and Mendes, J. F. F., 2003. *Evolution of Networks: From Biological Nets to the Internet and WWW*. Oxford University Press, Oxford, UK.
- Duch, J. and Arenas, A., 2005. Community detection in complex networks using Extremal Optimization. *Physical Review E*, 72(2):027104.
- Ebel, H., Davidsen, J., and Bornholdt, S., 2003. Dynamics of Social Networks. *Complexity*, 8(2):24–27.
- Erdős, P. and Rényi, A., 1959. On random graphs I. *Publicationes Mathematicae (Debrecen)*, 6:290–297.
- Euler, L., 1736. Solutio Problematis Ad geometriam Situs Pertinentis. *Commenrarii Academiae Scientiarum Imperialis Petropolitanae*, 8:128–140. Reprint in English in N. Biggs, E. Lloyd, and R. Wilson (1976), editors, *Graph Theory 1736-1936*, Clarendon Press, Oxford, UK.
- Feld, S. L., 1981. The Focused Organization of Social Ties. *The American Journal of Sociology*, 86(5):1015–1035.
- Ferguson, T. S., 1973. A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics*, 1(2):209–230.
- Fiore, A. T. and Donath, J. S., 2005. Homophily in Online Dating: When Do You Like Someone Like Yourself? In G. C. van der Veer and C. Gale, editors, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2005)*, pages 1371–1374. ACM Press, New York, USA.

- Fortunato, S. and Barthelemy, M., 2007. Resolution limit in community detection. *Proceedings of the National Academy of Sciences USA (PNAS)*, 104(1):36–41.
- Freedman, D. A., 1963. On the asymptotic behavior of Bayes estimates in the discrete case. *The Annals of Mathematical Statistics*, 34:1386–1403.
- Fruchterman, T. M. J. and Reingold, E. M., 1991. Graph Drawing by Force-Directed Placement. *Software – Practice and Experience*, 21(11):1124–1164.
- Gelman, A., 1996. Inference and Monitoring Convergence. In W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors, *Markov chain Monte Carlo in practice*, pages 131–143. Chapman & Hall, London, UK.
- Gelman, A., Carlin, J., Stern, H., and Rubin, D., 2003. *Bayesian data analysis*. Chapman & Hall, Boca Raton, USA, second edition.
- Geman, S. and Geman, D., 1984. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 6(6):721–741.
- Getoor, L. and Diehl, C. P., 2005. Link Mining: A Survey. *SIGKDD Explorations Special Issue on Link Mining*, 7(2):3–12.
- Geweke, J., 1992. Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. In *Bayesian Statistics*, volume 4, pages 169–193. Oxford University Press, Oxford, UK.
- Geweke, J., 2007. Interpretation and inference in mixture models: Simple MCMC works. *Computational Statistics & Data Analysis*, 51(7):3529–3550.
- Ghahramani, Z., 2004. Unsupervised Learning. In *Advanced Lectures on Machine Learning (LNAI)*, volume 3176 of *Lecture Notes in Computer Science*, pages 72–112. Springer-Verlag, Berlin, Germany.
- Girvan, M. and Newman, M. E. J., 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences USA (PNAS)*, 99(12):7821–7826.
- Gleiser, P. and Danon, L., 2003. Community Structure in Jazz. *Advances in Complex Systems*, 6(4):565–573.
- Griffiths, T. L. and Steyvers, M., 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences USA (PNAS)*, 101 Suppl 1:5228–5235.
- Griffiths, T. L. and Yuille, A. L., 2006. Technical Introduction: A Primer on Probabilistic Inference. Technical report, University of California Digital Repositories, Department of Statistics, University of California (UCLA), Los Angeles, USA. Available at: <http://repositories.cdlib.org/uclstat/papers/2006010103>.

- Guardiola, X., Guimera, R., Arenas, A., Diaz-Guilera, A., Streib, D., and Amaral, L. A. N., 2002. Macro- and micro-structure of trust networks. *ArXiv Physics e-prints*. cond-mat/0206240v1.
- Guimera, R., Danon, L., Diaz-Guilera, A., Giralt, F., and Arenas, A., 2003. Self-similar community structure in a network of human interactions. *Physical Review E*, 68(6):065103.
- Guimera, R., Sales-Pardo, M., and Amaral, L. A. N., 2004. Modularity from Fluctuations in Random Graphs and Complex Networks. *Physical Review E*, 70(2):025101.
- Gustafsson, M., Lombardi, A., and Hornquist, M., 2006. Comparison and validation of community structures in complex networks. *Physica A*, 367:559–576.
- Handcock, M. S., Raftery, A. E., and Tantrum, J. M., 2007. Model-based clustering for social networks. *Journal Of The Royal Statistical Society Series A*, 170(2):301–354.
- Haythornthwaite, C. and Wellman, B., 1998. Work, friendship and media use for information exchange in a networked organization. *Journal of the American Society for Information Science*, 49(12):1101–1114.
- Heckerman, D., Meek, C., and Koller, D., 2004. Probabilistic Entity-Relationship Models, PRMs, and Plate Models. In T. Dietterich, L. Getoor, and K. Murphy, editors, *Working Notes of the ICML-2004 Workshop on Statistical Relational Learning and Connections to Other Fields (SRL-2004)*, pages 55–60. ICML, Banff, Canada.
- Herring, S. C., 2002. Computer-mediated communication on the Internet. *Annual Review of Information Science and Technology*, 36:109–168.
- Hoff, P. D., Raftery, A. E., and Handcock, M. S., 2002. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97:1090–1098.
- Hofmann, T., 1999. Probabilistic Latent Semantic Analysis. In K. B. Laskey and H. Prade, editors, *Proceedings of Uncertainty in Artificial Intelligence (UAI'99)*, pages 50–57. Morgan Kaufmann Publishers, Stockholm, Sweden.
- Holme, P., 1994. *Form and function of complex network*. Ph.D. thesis, Department of Physics, Umeå University, Umeå, Sweden.
- Huisman, M. and van Duihin, M. A. J., 2005. Software for Social Network Analysis. In P. Carrington, J. Scott, and S. Wasserman, editors, *Models and Methods in Social Network Analysis*, pages 270–316. Cambridge University Press, New York, USA.
- Ibarra, H., 1992. Homophily and differential returns: Sex differences in network structure and access in an advertising firm. *Administrative Science Quarterly*, 37(3):422–447.
- Jain, A. K., Murty, M. N., and Flynn, P. J., 1999. Data Clustering: A Review. *ACM Computing Surveys*, 31(3):264–323.
- Java, 2007. Java Platform, Standard Edition 5.0. Sun Microsystems, Inc. Available at: <http://java.sun.com/j2se/1.5.0/>, accessed 22 August 2007.

- Kadushin, C., 2004. Introduction to Social Network Theory. Available at: <http://home.earthlink.net/~ckadushin/Texts/Basic%20Network%20Concepts.pdf>, draft version of chapter 2, accessed 23 August 2007.
- Keller, M., 2004. A Cross-Cultural Perspective on Friendship Research. *International Society for the Study of Behavioural Development Newsletter*, 46(2):10–11.
- Kemp, C., Griffiths, T. L., and Tenenbaum, J. B., 2004. Discovering latent classes in relational data. Technical report, Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, USA.
- Kemp, C., Tenenbaum, J. B., Griffiths, T. L., Yamada, T., and Ueda, N., 2006. Learning Systems of Concepts with an Infinite Relational Model. In Y. Gil and R. Mooney, editors, *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*. AAAI Press, Menlo Park, USA.
- Kempe, D., Kleinberg, J., and Kumar, A., 2000. Connectivity and inference problems for temporal networks. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing (STOC'00)*, pages 504–513. ACM Press, New York, USA.
- Kleinberg, J., 2003. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373–397.
- Kumpula, J. M., Saramaki, J., Kaski, K., and Kertesz, J., 2007. Limited resolution in complex network community detection with Potts model approach. *The European Physical Journal B*, 56(1):41–45.
- Lake, C., 2006. Interview with Martin Stiksel of Last.fm. *E-consultancy*. Available at: <http://www.e-consultancy.com/news-blog/362081/interview-with-martin-stiksel-of-last-fm.html>, posted online 8 Nov 2006. Accessed 23 August 2007.
- Lazarsfeld, P. and Merton, R. K., 1954. Friendship as a Social Process: A Substantive and Methodological Analysis. In *Freedom and Control in Modern Society*, pages 18–66. Van Nostrand, New York, USA.
- Leskovec, J., Kleinberg, J., and Faloutsos, C., 2005. Graphs over time: densification laws, shrinking diameters and possible explanations. In R. L. Grossman, R. Bayardo, K. Bennett, and J. Vaidya, editors, *Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining (KDD'05)*, pages 177–187. ACM Press, New York, USA.
- Liben-Nowell, D., Novak, J., Kumar, R., Raghavan, P., and Tomkins, A., 2005. Geographic Routing in Social Networks. *Proceedings of the National Academy of Sciences USA (PNAS)*, 102(33):11623–11628.
- Lieken, A., 2007. Data mining musical profiles. Available at: <http://anthony.lieken.net/index.php/Computers/DataMining>, accessed 21 August 2007.

- Macskassy, S. A. and Provost, F., 2004. Simple Models and Classification in Networked Data. In *CeDER Working Paper 03-04*. Stern School of Business, New York University, New York, USA. Working paper.
- MADE4, 2007. MADE4: Multivariate analysis of microarray data using ADE4. Available at: <http://bioinf.ucd.ie/people/aedin/R/>, accessed 22 August 2007.
- Maimon, O. and Rokach, L., 2005. *The Data Mining and Knowledge Discovery Handbook*. Springer-Verlag, New York, USA.
- McLachlan, G. J. and Peel, D. A., 2000. *Finite Mixture Models*. John Wiley & Sons, New York, USA.
- McPherson, M., Smith-Lovin, L., and Cook, J., 2001. Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology*, 27:415–444.
- Michaelson, A. and Contractor, N. S., 1992. Structural Position and Perceived Similarity. *Social Psychology Quarterly*, 55(3):300–310.
- Minka, T., 2003. Bayesian inference, entropy, and the multinomial distribution. Microsoft Research. Available at: <http://research.microsoft.com/~minka/papers/multinomial.html>, online tutorial.
- Muff, S., Rao, F., and Caflisch, A., 2005. Local modularity measure for network clusterizations. *Physical Review E*, 72(5 Pt 2):056107.
- Navarro, D. J., Griffiths, T. L., Steyvers, M., and Lee, M. D., 2006. Modeling individual differences using Dirichlet processes. *Journal of Mathematical Psychology*, 50(2):101–122.
- Neal, R. M., 2000. Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics*, 9(2):249–265.
- Neville, J. and Jensen, D., 2005. Leveraging relational autocorrelation with latent group models. In S. Džeroski and H. Blockeel, editors, *Proceedings of the 4th international workshop on Multi-relational mining (MRDM-05)*, pages 49–55. ACM Press, New York, USA.
- Newman, M., Barabasi, A.-L., and Watts, D. J., 2006. *The Structure and Dynamics of Networks*. Princeton University Press, Princeton, USA.
- Newman, M. E. J., 2001. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences USA (PNAS)*, 98(2):404–409.
- Newman, M. E. J., 2003a. The structure and function of complex networks. *SIAM Review*, 45(2):167–256.
- Newman, M. E. J., 2003b. Mixing patterns in networks. *Physical Review E*, 67(2 Pt 2):026126.

- Newman, M. E. J., 2004. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6):066133.
- Newman, M. E. J., 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences USA (PNAS)*, 103:8577–8582.
- Newman, M. E. J. and Girvan, M., 2004. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113.
- Newman, M. E. J. and Leicht, E. A., 2006. Mixture models and exploratory data analysis in networks. *ArXiv Physics e-prints*. physics/0611158.
- Nowicki, K. and Snijders, T. A. B., 2001. Estimation and Prediction for Stochastic Blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087.
- Paccagnella, L., 1998. Language, Network Centrality, and Response to Crisis in On-line Life: A Case Study in the Italian Cyberpunk Computer Conference. *The Information Society*, 14(2):117–135.
- Palla, G., Derenyi, I., Farkas, I., and Vicsek, T., 2005. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818.
- Park, H. W., 2003. Hyperlink Network Analysis: A New Method for the Study of Social Structure on the Web. *Connections*, 25(1):49–61.
- Prefuse, 2007. Prefuse: Interactive information visualization toolkit. Available at: <http://prefuse.org/>, accessed 22 August 2007.
- Pujol, J. M., Béjar, J., and Delgado, J., 2006. Clustering algorithm for determining community structure in large networks. *Physical Review E*, 74(1):016107.
- Qamra, A., Tseng, B., and Chang, E. Y., 2006. Mining blog stories using community-based and temporal clustering. In *Proceedings of the 15th ACM international conference on Information and knowledge management (CIKM '06)*, pages 58–67. ACM Press, New York, USA.
- Rapoport, A., 1957. Contribution to the theory of random and biased nets. *Bulletin of Mathematical Biophysics*, 19:257–277.
- Ritter, C. and Tanner, M. A., 1992. Facilitating the Gibbs Sampler: The Gibbs Stopper and the Griddy-Gibbs Sampler. *Journal of the American Statistical Society*, 87(419):861–868.
- Sethuraman, J., 1994. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650.
- Sinkkonen, J., Aukia, J., and Kaski, S., 2007. Inferring vertex properties from topology in large networks. In *Working Notes of the 5th International Workshop on Mining and Learning with Graphs (MLG'07)*. Università degli Studi di Firenze, Florence, Italy. Extended Abstract.

- Streeter, C. L. and Gillespie, D. F., 1992. Social Network Analysis. *Journal of Social Service Research*, 16(1/2):201–222.
- Tavare, S. and Ewens, W. J., 1997. The Ewens sampling formula. In *Multivariate discrete distributions*. John Wiley & Sons, New York, USA.
- Travers, J. and Milgram, S., 1969. An Experimental Study of the Small World Problem. *Sociometry*, 32(4):425–443.
- Trove, 2007. GNU Trove: High performance collections for Java. Available at: <http://trove4j.sourceforge.net/>, accessed 22 August 2007.
- Tukey, J. W., 1977. *Exploratory data analysis*. Addison-Wesley, Reading, USA.
- Turner, J. C., Hogg, M. A., Oakes, P. J., Reicher, S. D., and Wetherell, M. S., 1987. *Rediscovering the Social Group: A Self-Categorization Theory*. Basil Blackwell, Oxford, UK.
- Van De Bunt, G. G., Van Duijn, M. A. J., and Snijders, T. A. B., 1999. Friendship Networks Through Time: An Actor-Oriented Dynamic Statistical Network Model. *Computational and Mathematical Organization Theory*, 5(2):167–192.
- Wakita, K. and Tsurumi, T., 2007. Finding community structure in mega-scale social networks. *ArXiv Physics e-prints*. cs/0702048v1.
- Wasserman, S. and Faust, K., 1994. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, UK.
- Watts, D. J. and Strogatz, S. H., 1998. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442.
- Weiss, M. A., 1998. *Data Structures and Algorithm Analysis in Java*. Addison-Wesley, Reading, USA.
- Wong, C. K. and Easton, M. C., 1980. An Efficient Method for Weighted Sampling without Replacement. *SIAM Journal on Computing*, 9(1):111–113.
- Yuan, Y. C. and Gay, G., 2006. Homophily of Network Ties and Bonding and Bridging Social Capital in Computer-Mediated Distributed Teams. *Journal of Computer-Mediated Communication*, 11(4):1062–1084.
- Zachary, W. W., 1977. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473.
- Zahn, G. L., 1991. Face-to-Face Communication in an Office Setting: The Effects of Position, Proximity, and Exposure. *Communication Research*, 18(6):737–754.
- Zhang, S., Wang, R.-S., and Zhang, X.-S., 2007. Identification of overlapping community structure in complex networks using fuzzy c-means clustering. *Physica A*, 374(1):483–490.

- Zheng, A. X. and Goldenberg, A., 2006. Exploratory Study of a New Model for Evolving Networks. In *Proceedings of the Workshop on Statistical Network Analysis: Models, Issues, and New Directions at ICML-06*, Lecture Notes in Computer Science. Springer-Verlag, Berlin, Germany. To appear.
- Zhu, X., 2005. Semi-Supervised Learning Literature Survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, Madison, USA. Available at: http://www.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf.