HELSINKI UNIVERSITY OF TECHNOLOGY
Faculty of Electronics, Communications and Automation

Juuso Parkkinen

GENERATIVE PROBABILISTIC MODELS OF BIOLOGICAL AND SOCIAL NETWORK DATA

Thesis submitted for examination for the degree of Master of Science in Technology

Espoo 3.9.2008

Thesis supervisor:

Professor Samuel Kaski

Thesis instructor:

Janne Sinkkonen, Ph.D.

TEKNILLINEN KORKEAKOULU

DIPLOMITYÖN TIIVISTELMÄ

Tekijä: Juuso Parkkinen

Työn nimi: Generatiivisia todennäköisyysmalleja biologiselle ja sosiaaliselle verkkodatalle

Päivämäärä: 3.9.2008 Kieli: Englanti Sivumäärä: 9+61

Tiedekunta: Elektroniikan, tietoliikenteen ja automaation tiedekunta

Professuuri: Informaatiotekniikka Koodi: T-61

Valvoja: Professori Samuel Kaski

Ohjaaja: FT Janne Sinkkonen

Useat monimutkaiset systeemit voidaan esittää verkkona, jossa kaaret yhdistävät solmuja. Soluissa molekyylien, kuten proteiinien, vuorovaikutukset muodostavat verkon, ja sosiaalinen systeemi voi koostua yksittäisten toimijoiden suhteista. Verkkojen analysointi on kehittynyt pienen ihmisjoukon välisten suhteiden tutkimisesta valtavien monimutkaisten verkkojen, kuten Facebookin ja MySpacen tapaisten kommunikaatioverkkojen tai solun laajuisten molekyyliverkkojen, analysointiin. Sen lisäksi, että käytännön verkot ovat erittäin suuria, ne ovat tyypillisesti harvoja ja epätäydellistä. Tällaisten verkkojen menestyksekäs analysointi vaatii kehittyneiden laskennallisten menetelmien käyttöä.

Tämän diplomityön aiheena on uusi generatiivinen todennäköisyysmalliperhe, vuorovaikutuskomponenttimallit. Se on suunniteltu tiheästi kytkettyjen aliverkkojen löytämiseen kohinaisesta verkkodatasta. Tällaisilla aliverkoilla on monia tulkintoja käytännön sovelluksissa, kuten toiminnalliset geenimoduulit proteiinien vuorovaikutusverkoissa tai yhteisöt sosiaalisissa verkoissa. Malliperhe on suunniteltu mahdollisimman yksinkertaiseksi, jotta se olisi ymmärrettävä ja laskennallisesti toteutettavissa.

Tässä työssä mallia sovelletaan uuteen ongelmaan, proteiinien vuorovaikutusverkkoihin, ja tavoitteena on löytää biologisesti järkeviä toiminnallisia moduuleita. Vaihtoehtoja mallin laajentamiseksi ymmärtämään myös verkkoja rikkaampaa dataa, kuten solmujen ominaisuuksia, esitellään ja kokeillaan. Tehdyissä kokeissa mallit löytävät tulkittavia klusterirakenteita verkoista useilla sovellusalueilla. Ehdotetut muutokset parantavat mallin suorituskykyä.

Avainsanat: Bayesilainen päättely, geenilekspressio, proteiinien vuorovaikutus, relationaalinen data, toiminnallinen moduuli, verkkodata, vuorovaikutuskomponenttimalli

Author: Juuso Parkkinen

Title: Generative Probabilistic Models of Biological and Social Network Data

Date: 3.9.2008 Language: English Number of pages: 9+61

Faculty: Faculty of Electronics, Communications and Automation

Professorship: Computer and Information Science Code: T-61

Supervisor: Professor Samuel Kaski

Instructor: Janne Sinkkonen, Ph.D.

Many complex systems can be represented as networks in which nodes are connected with edges. In cells, interactions between molecules, such as proteins, form a network, and social systems can consist of relationships between individual actors. Network analysis has developed from early studies of relationships between a small group of people to the analysis of huge complex networks, such as communication networks like Facebook and MySpace, or cell-wide biomolecular networks. In addition to being very large, the networks arising from real-world systems are typically sparse and contain missing and incomplete data. Successful analysis of such networks thus requires advanced computational methods.

The topic of this thesis is a new generative probabilistic modeling framework, interaction component models, which is designed to detect densely connected subnetworks from noisy network data. Such subnetworks have many interpretations in practical applications, such as functional gene modules in protein interaction networks or communities in social networks. The model family is designed to be as simple as possible, to keep it understandable and computationally feasible.

In this thesis, the model is applied to a new problem domain, namely protein interaction networks, in order to detect biologically relevant functional modules. Extensions to include additional data, such as attributes of the nodes, into the analysis are proposed and tested. Improvements to model inference are also introduced and their effect studied. In the experiments, models are able to find meaningful cluster structures from networks in several problem domains. The proposed modifications improve model performance.

Keywords: Bayesian inference, functional module, gene expression, interaction component model, network data, protein interaction, relational data

Preface

The work reported in this thesis was carried out in the Helsinki Institute for Information Technology HIIT, Department of Information and Computer Science, funded by the Adaptive Informatics Research Centre of the Helsinki University of Technology. The research was supported in part by the PASCAL 2 EU Network of Excellence, ICT 216886.

I would like to thank Prof. Samuel Kaski for the supervision and overall guidance during the research, and my instructor Ph.D. Janne Sinkkonen for his helpful comments and support. The biological part of the thesis was done in close collaboration with Prof. Kaski, and the social part with both Prof. Kaski and Ph.D. Sinkkonen. The work was designed together and my personal contribution was to derive equations for and implement the extensions and improvements presented in the thesis. I also carried out the experiments and validated the results.

I would also like to thank the whole department and especially the people in the Statistical Machine Learning and Bioinformatics group for providing such a fruitful academic atmosphere and for inspiring conversations.

Finally, I would like to thank my fiancée Sanna for her love and support.

Otaniemi, 3.9.2008

Juuso A. Parkkinen

Contents

Symbols and abbreviations

Symbols

C	Number of components
$Dir(x)$	Dirichlet distribution with a symmetric parameter x
DP	Dirichlet process
i, j	Nodes in a graph
L	Set of edges in a graph
M	Number of nodes
N	Number of links
$N(x, y)$	Normal distribution with mean x and variance y
n_z	Number of links assigned to component z
$p(x)$	Probability mass or density of x
$p(x\|y)$	Conditional probability of x given y
$p(x, y)$	Joint probability of x and y
q_{zi}	Number of co-occurences of component z and node i
V	Covariance matrix
x	Real-valued scalar or vector
$x \propto y$	x is proportional to y
z	Latent component for edge
Z	Set of latent components for edges
α	Hyperparameter for the distribution over components
β	Hyperparameter for the component-specific distribution over nodes
Γ	Gamma function
θ	Probability distribution over components
$\bar{\mu}$	Mean vector of a gene expression profile
σ	Standard deviation
ϕ,φ	Component-specific probability distribution over nodes

Abbreviations

DNAD	DNA damage
EM	Expectation Maximization
GO	Gene Ontology
HMoF	Hidden Modular Random Field
ICM	Interaction Component Model
ILP	Inductive Logic Programming
IRM	Infinite Relational Model
IHRM	Infinite Hidden Relational Model
KL	Kullback-Leibler
LDA	Latent Dirichlet Allocation
MAP	Maximum a Posteriori
MCL	Markov Clustering
MCMC	Markov Chain Monte Carlo
MF	Matrix factorization
ML	Maximum Likelihood
OSMO	Osmotic shock response
PCA	Principal component analysis
PGM	Probabilistic graphical model
PLSA	Probabilistic latent semantic analysis
LSA	Latent semantic analysis
LSI	Latent semantic indexing
PPI	Protein-protein interaction
PRM	Probabilistic relational model
SNA	Social networks analysis
SRCM	Simple Relational Component Model
SRL	Statistical relational learning
SSN-DA	Simple Social Interaction LDA
SVD	Singular value decomposition
MMSB	Mixed Membership Stochastic Block model
Y2H	Yeast two-hybrid
TAP-MS	Tandem affinity purification followed by mass spectrometric analyses

1 Introduction

1.1 Problem setting

Large data collections in many fields can be presented in the form of networks or graphs. For example, proteins in a cell exhibit complicated interaction patterns, forming a cell-wide interaction network. In social context, networks can represent different types of relationships between individuals, such as friendships.

Real-world networks are analyzed to understand the structure and properties of the networks. In social networks, for instance, interesting questions for network analysis include the formation of communities, that is, strongly connected subgraphs, and the evolution of the networks over time. Analysis of network structure has also applications in network comparison, visualization, anonymization, experimental design, and optimization, to name a few examples.

Network analysis has developed from traditional interview-based analysis of very small social networks to computational models of huge communication networks. One common target for network analysis in many scientific fields is the detection of clusters or groups of nodes that are similar in some sense. The similarity may reflect the network topology or some other attributes, such as the participation of genes to the same biological processes.

In mathematics and computational science, networks have been studied as graphs that consist of a set of nodes and edges that connect nodes to each other. Graphs have been studied for decades and there exists an established genre of analysis methods. Many methods have been successfully applied to analysis of real-world networks represented as graphs.

Although a graph as an abstraction of real-world network data is in many cases flexible and useful, it has its drawbacks. Representing data as simple binary relations between entities may cause loss of essential information, such as node information and relation types. An alternative is to use a richer relational model. Choosing a suitable abstraction for data is always a compromise between representative power and computational capacity. As part of the thesis, solutions to this problem are sought and discussed.

Networks are a central representation for data in bioinformatics. Many cellular systems can be presented in the form of networks, where molecules interact with each other to perform biological processes. Typical examples of biological networks are metabolic pathways, regulatory networks and protein interaction networks. Better understanding of the structure and function of these systems may result in new biological knowledge and medical treatments, for instance.

Analyzing large networks is an example of complex systems research. Complex systems are composed of interconnected parts that as a whole exhibit properties that are not obvious from the properties of the individual parts. Complex systems can be found everywhere, for example in human economies, social structures, and

cells. Complex systems are studied by many schools of natural science, mathematics, and social science.

Computational methods of network data have to cope with several challenges: The data is typically sparse, noisy and incomplete. Many computational approaches have been presented for solving these problems. In this thesis, the focus is on generative probabilistic models, a genre of machine learning methods where the assumptions of the data are encoded into the model structure, and statistical inference is then used to learn the parameters from the data. The parameters can then tell about the captured structure of the data, for instance clustering of the nodes in a network.

In the thesis, a generative model framework for network data is applied to clustering tasks in biological and social networks. The suitability of the model for different problem domains is interesting in general, as the research questions can be very similar despite the heterogeneity of the different real-world networks. They have also been shown to share structural and functional properties. In addition to studying the suitability of the model for clustering tasks in different problem domains, the integration of multiple data sources into the analysis of simple network data is studied.

The ultimate goal guiding this work is to develop efficient, yet easy-to-use and interpretable computational methods for analyzing real-world network data. They could then be used to formulate testable hypotheses of the studied systems, such as protein interaction networks.

1.2 Contributions of the thesis

In this thesis applications and extensions are presented to a generative network model framework called Interaction Component Model (ICM). It was originally developed and introduced in the research group by Janne Sinkkonen, Janne Aukia and Samuel Kaski [53] for detecting communities from large social networks.

The method framework is here applied to a new problem domain, namely biological networks. In particular, the developed models are applied to a protein interaction network in order to seek functional modules, and also to study possibilities to incorporate functional gene data into the analysis. The integration of node attributes to the model is also studied and compared in a relational data setting. The model inference is developed with a hyperparameter estimation procedure and a new convergence estimator. The effects of these improvements are studied with experiments on citation networks.

1.3 Structure of the thesis

This thesis is organized as follows: Sections 2–4 give necessary background for understanding the main content of the thesis. In Section 2, background information is given about networks and relational data. Section 3 takes a closer look at biological

and social networks and their characteristic properties, with a brief comparison. In Section 4, background information is given about Bayesian inference and probabilistic graphical models, which form the basis for the computational tools used in the thesis. This section also includes a brief survey on related work.

In Section 5, the general idea of the ICM framework is first presented with technical details. Second, several extensions and improvements to the model framework are introduced, which are the main contributions of the thesis. The experimental part is divided into biological (Section 6) and social (Section 7) sections. These sections contain the experimental setup, including the used data sets and necessary details of each conducted experiment, as well as the results and conclusions. Finally, the thesis is wrapped up with discussion in Section 8.

2 Networks as data

Real-world systems that take the form of networks are abundant in various scientific fields. Examples include many biological networks, social networks of relationships between individuals, the Internet, networks of citations between documents, and many others.

Network data arising from different origins may seem very heterogeneous, indicating that their analysis should always be tailored for the domain-specific needs. Interestingly, it has been shown that there exist many fundamental commonalities in the properties of different types of real-world networks, forming the basis for interdisciplinary network analysis. On the other hand, the question remains whether these commonalities arise due to the common representation or are a sign of deeper relationships.

The simplest and most common representation for network data is a graph of nodes with edges connecting nodes to each other. Using a graph representation has many benefits, as there is a wide established genre of analysis methods for graph data that can be utilized whenever a real-world system is represented as a graph. In common language, networks and graphs are often used to describe the same objects, although there is an important distinction between them. A graph is a rather simple mathematical object, whereas a network can include much more information than the simple graph-like structure. For example, a social friendship network could consist of not only the people and their relationships (a graph), but also a variety of attributes for the people and their relationships. In this thesis the term 'network' is used for this kind of data structures.

On the other hand, networks can be thought of as *relational data*. Relations in principle correspond to the edges of a graph, but in a typical relational data setting there are multiple relations between the entities, and a graph representation is not sufficient to capturing all necessary aspects of the data. Such a multi-relational setting is close to many real-world networks. Analysis of networks on the one hand and relational data on the other have traditionally been distinct, but it has become increasingly evident that they have much in common.

The focus of this section is on the basic properties of networks as data. In addition, a brief background of relational data is given as well, to make the connection with networks clear and to help understanding the multi-relational model extensions and experiments.

2.1 Basics of network and relational data

In mathematics, a graph is an abstract representation of a set of objects where some pairs of the objects are connected by links. The objects are represented by mathematical abstractions called vertices, and the links that connect some pairs of

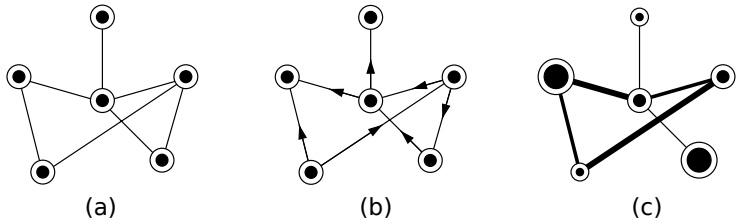

Figure 1: Examples of various types of graphs: (a) an undirected graph with only a single type of vertex and a single type of edge, (b) a directed graph in which each edge has a direction, (c) a graph with varying vertex and edge weights. Figure adapted from Newman [42].

vertices are called edges[1]. Typically, a graph is depicted in diagrammatic form as a set of dots for the vertices, joined by lines or curves for the edges, as in Figure 1.

In numerical form, graphs can be presented as a *sparse* array of connected node pairs, with possibly additional information about the connection. Another analogous representation is the *adjacency matrix* with rows and columns corresponding to the nodes. In its simplest form, each cell of the matrix has a binary value $\{0, 1\}$, indicating the (non-)existence of a link between the corresponding nodes. This connection between networks and matrices holds on a general level, in other words a graph can always be represented as a matrix. This enables the use of matrix operations for graph data, and many matrix-based methods can be applied to networks as well, such as spectral methods.

Figure 1a illustrates a simple binary graph, a common form for presenting networks. Network data can, though, contain much more than simple binary links between nodes. First, links may be *directed*, making a distinction between *sending* and *receiving* nodes (Figure 1b). Another commonly appearing feature of links is *weight*, which describes the strength or probability of a link. Nodes can also have weights (Figure 1c).

2.1.1 Graph theory

A traditional school of network analysis is called *graph theory*, the ideas of which can be tracked back to the 18th century when Leonhard Euler presented his solutions to the Königsberg bridges problem [21]. Graph theory is a purely mathematical approach for studying graphs, with numerous definitions and formulations on graph properties.

An important part of graph theory is the development of algorithms as solutions to practical graph-theoretic problems, such as the minimum-connector problem or the shortest and longest-path problems. A famous example is Dijkstra's algorithm for

[1]Different words for these are used in different contexts; in this thesis the words vertex and node are used inter-exchangeably, as well as edge, link and interaction.

ACTOR	
name	gender
fred	male
ginger	female
bing	male

MOVIE	
name	genre
m1	drama
m2	comedy

ROLE			
role	movie	actor	role-type
r1	m1	fred	hero
r2	m1	ginger	heroine
r3	m1	bing	villain
r4	m2	bing	hero
r5	m2	ginger	love-interest

Figure 2: Example of a relational schema for a movie domain. There are two types of entities (actors and movies), and each have different attributes (name, gender, genre). There are also relations between actors and movies (roles), with various types (hero, heroine, villain, love-interest). Figure adapted from Getoor et al. [27].

searching for the shortest path from a single node to all other nodes in the graph [17]. The algorithm is widely applied in routing tasks, such as finding the fastest bus connections within a city. Interestingly, many problems of practical interest can be represented by graphs and solved with graph-theoretic approaches and algorithms.

2.1.2 Relational data

Relational data and *relational model* as concepts have their origins in database management. The concept of relational database was originally defined by Edgar Codd [15]. The fundamental assumption of the relational model is that all data are represented as mathematical *n-ary* relations, an n-ary relation being a subset of the Cartesian product of n domains. For example a relationship between two entities is a binary relation.

Figure 2 shows a simple example of a relational schema for a movie domain. Another way to present the same data would be as a network, where the relations would be presented by a graph of role-edges between actors and movies, and then each entity and relation would have additional attributes. This illustrates the connection between networks and relational data.

Relational representation becomes handy for network analysis in cases where the network data is enriched with additional attributes for either links or nodes or both, as in the actor-movie-example (Figure 2), where we have different types of roles relating actors to movies. This kind of rich data is often referred to as *multi-relational* data, stressing the fact that there are now multiple types of relations between the entities.

2.1.3 Network data representations

A question arises about the differences in using a simple graph representation of data compared to a richer relational model. The question generalizes to that of choosing a suitable level of abstraction in representing data. In the case of relational data, this in practice means choosing what type of data we think is important for the analysis. For example, are the possibly different types, directions and weights for links essential, or is a simple binary link representation enough? As a part of this thesis, ways to incorporate node-wise data into simple network analysis are developed and analyzed and their benefits and drawbacks are discussed.

John Young stated in 1986 that drawbacks of a relational database include the heavy use of computer resources and the implications for data integrity. Flexibility may be obtained at the expense of performance [66]. In many cases the simplest network approach may indeed prove effective enough, as numerous successful studies have shown.

The problem of choosing a suitable representation becomes apparent as well when analyzing large networks or relational datasets with computational tools: there is always a trade-off to be made between trying to model everything and what is computationally possible. Traditional statistical learning methods force people to convert their data into a form that loses much of the relational structure. However, some recent developments in probabilistic modeling have made possible the use of much richer dependencies in data [27]. This will be discussed more in Section 2.3.3.

2.2 Data analysis with machine learning

Data analysis is the process of gathering, modeling, and transforming data with the goal of highlighting useful information, suggesting conclusions, and supporting decision making. Computational tools have been developed for analyzing large amounts of data which would be in practice impossible manually.

Many method genres have emerged that could be called computational data analysis, such as data mining, neural networks, machine learning and pattern recognition. Although these have developed from different backgrounds, they are actually largely overlapping. For instance, machine learning has its roots in computer science and statistics, whereas pattern recognition has developed from engineering, but nowadays these can be viewed as two facets of the same field.

Machine learning is a relatively recently developed branch of computer science, in which algorithms are developed that allow computers to learn based on data. A major focus here is to automatically learn to discover and recognize complex patterns and make intelligent decisions and predictions based on data. Machine learning techniques are often used for *exploratory data analysis*, where the aim is to formulate new *hypotheses* about the data, as opposed to *confirmatory* data analysis, where predefined hypotheses are being tested by statistical means [61].

A distinction is commonly made between two types of machine learning methods:

supervised and *unsupervised learning* methods. In supervised methods the data comprises examples of the input vectors (features of data elements) along with their corresponding target vectors (known classes of the elements, or *labels*). The task can then be *classification* of data, where the aim is to assign each input vector to one of a finite number of predefined discrete categories. If the desired output consists of one or more continuous variables, the task is called *regression*. [9]

In other problems the training data consists of a set of input vectors without any corresponding target values. The goal in this unsupervised setting may be to discover groups of similar examples within the data, in which case it is called *clustering*, or to determine the distribution of data within the input space, known as *density estimation*. Another common task is *visualization*, where the data is projected from a high-dimensional space down to two or three dimensions and then visualized. [9]

This division is however not clear-cut, for example, recently the concept of *semi-supervised* methods has been introduced [14]. In a semi-supervised setting some data is labeled, but also a large amount of unlabeled data, and the task is to combine them into an efficient learning method for classification purposes.

2.3 Analysis of networks and relational data

Computational methods for data analysis are in principle applicable to a large variety of application fields; for example a simple clustering method called *k-means* and its advanced versions have been applied in computer vision for image segmentation [38] and in bioinformatics for clustering gene expression profiles [57]. This follows from the fact that many kinds of real world data can be presented in a similar form, making the same methods directly applicable for data from different sources.

However, there is a fundamental difference between the typical statistical data analysis case, where the data consists of *independent* observations, and relational data, where there observations are related and hence *dependent*. Key concept here is the *independent and identically distributed* (i.i.d.) property of data, an assumption that often simplifies the mathematics of statistical methods. This data representation is sometimes called *flat*. Dependency of data points causes more complexity to the data, which has to be accounted for in the models, and this is why many commonly used algorithms cannot be directly applied on network data.

However, the problem described above can often be overcome with modifications to the original models, for example previously mentioned k-means was very recently applied on graph clustering [47]. In subsequent sections, several approaches for analysis of network and relational data are covered more closely, many of which have their origins in elsewhere.

2.3.1 Complex networks analysis

An active branch of network analysis is focused on *complex networks*, studying complex graphs with non-trivial topological features — features that do not occur in simple networks. Complex network analysis is focused on the empirical study of similarities and differences of real-world networks, motivated by questions like "How are the networks generated?" or "How do they evolve over time?"

Study of complex networks was pioneered by the *random graph* model of Erdös and Rènyi [20], which is among the simplest useful graph-generating models [42]. In the model each possible edge is independently present with some probability p. Very important concept in complex networks is the *small-world effect* [64], originating from the famous experiment by Stanley Milgram in 1960s [59]. The small-world concept states that most pairs of nodes in real-life networks seem to be connected by a short path through the network.

A lot of work has been carried out to study the degree[2] distributions of networks. A common feature of complex networks has been found to be that their degree distribution follows the *power law*: the fraction p_k of nodes in the graph having degree k follows $p_k \sim k^{-\alpha}$ for some constant α. Put in other words, the degree distribution of a network does not have any specific scale, hence the name *scale-free* is also often used in the context of complex networks [42].

Quite recently, Barabàsi and Albert presented [8] an improved version of the random graph model, called *preferential attachment* model. In this model, new links are added more likely to nodes that already have many links. This kind of network generation process leads to a scale-free network. In a more recent paper the same authors review the statistical mechanisms and dynamics of complex networks [4].

Observations of power-law nature in the connectivity of complex networks, such as biological an social networks, were inferred as a "universal architecture" of complex systems. Closer examination, however, challenged the assumptions that such distributions are special and signify a common architecture, independent of the system's specifics [34]. Also the recent work of Jure Leskovec et al. [36] has shaken the conventional views of how real-world-like networks should be generated, proposing an alternative to the preferential attachment model.

2.3.2 Graph clustering

One of the most popular goals in network data analysis is clustering. As described earlier, clustering of data aims at grouping similar elements together. Within graphs, the similarity is based on the topology of the nodes; typically clusters are thought of as sets of nodes that have a lot of connections between them and less connections to nodes in other clusters. In social networks such clusters are called *communities*. Examples of communities are shown in Figure 3. Network clustering has been

[2]The degree of a node in a network is the number of edges incident on, that is, connected to that node.

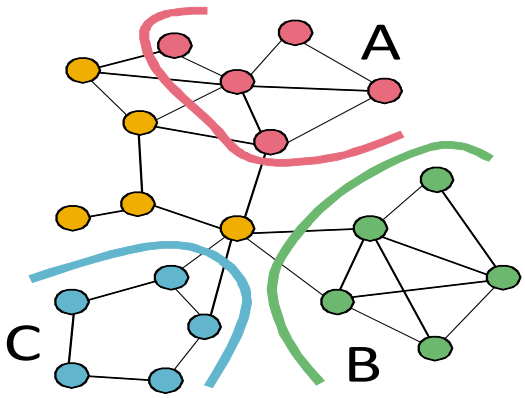

Figure 3: Example of network with community structure. Out of the five colored node sets B has the highest ratio between the number of edges inside and the number of edges outside, and is thus the most community-like set of nodes. Figure adapted from Leskovec et al. [36].

widely studied for a long time, with various approaches ranging from graph theory to statistics and machine learning.

The degree to which nodes in a graph are clustered together can be measured with the *clustering coefficient* (CC), which is defined for one node i (local clustering coefficient C_i) as the proportion of links between the nodes within its neighborhood divided by the number of links that could possibly exist between them. Formally this becomes (for undirected graphs)

$$C_i = \frac{2E_i}{k_i(k_i - 1)}, \tag{1}$$

where E_i is the number of edges connected to node i and k_i is the degree of node i. The clustering coefficient for the whole graph is then the average of those of each node in the graph. The coefficient can be used to characterize graphs, but not for finding an optimal clustering of nodes.

One approach for clustering the nodes is to define a measure of goodness for a given division of nodes into groups, and then design and algorithm to optimize this measure. An example measure that has drawn a lot of attention is *modularity* [43]. Modularity Q is high for those modules, which have dense internal connections between the nodes within modules but only sparse connections between different modules. Formally it can be defined as follows (for a particular division of network into two modules):

$$Q = \frac{1}{4m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) s_i s_j, \tag{2}$$

where m is the is the total number of edges in the graph, A_{ij} is the number of edges between nodes i and j (normally zero or one), k_i is the degree of node i and $s_i = 1$ if the node i belongs to group 1 and $s_i = -1$ if it belongs to group 2. Modularity for more than two modules is straightforward to derive based on this.

Newman presented a *spectral algorithm* for optimizing modularity and hence for clustering a graph. The algorithm is based on a special characteristic matrix of the network called modularity matrix [43]. Spectral clustering for graphs has become popular and improved versions have been developed since then, for example by Jiang et al. [33].

The spectral clustering methods presented above are called *deterministic* methods, meaning that the outcome of the algorithm is always the same. Another category of methods used in clustering are *probabilistic* methods, which in contrast to deterministic methods incorporate randomness in their functioning.

Recent and widely applied example of a probabilistic machine learning method for network clustering is called Markov Clustering (MCL) [18]. MCL is an unsupervised cluster algorithm for graphs based on simulation of stochastic flow in graphs, and it has been successfully applied on biological networks [11], among others.

In contrast to the methods described above, clustering of networks can also be performed by grouping such nodes together that link to same other nodes. A recent machine learning approach utilizing this assumption is the Simple Social Network Latent Dirichlet Allocation (SSN-LDA), presented in more detail in Section 4.3.2. Closely related family of network models are *stochastic block models*, where nodes are grouped based on their interaction patterns with other groups. A recent example of a block model is the Mixed-Membership Stochastic Block model (MMSB) by Airoldi et al. [2]

Network clustering algorithms can be categorized into *hard* and *soft* clustering, which makes a distinction between assigning nodes strictly into one cluster (hard) and allowing them to belong to many clusters, typically with some probability (soft). Choice between these depends on the application, for example proteins tend to participate in multiple cellular functions, indicating that soft clustering should be preferred. On the other hand, in a simple case of clustering people to different nationalities based on their friends can be performed as hard clustering.

Clustering methods vary also in practical properties, such as complexity, scalability and ease of use, affecting the choice of methods for any particular task. Many methods that have shown to perform well are very difficult to apply in practice, if they have a lot of parameters to set, a badly written documentation or uncommented implementations. Model complexity and scalability on the other hand set limits to the size of data sets the models can be applied to; for example a social network with millions of nodes and edges can only be efficiently analyzed with a very limited number of methods.

These practical things might well be the factors that lead to a decision in the end for example when a biologist wants to analyze her network data with some clustering algorithm, no matter how good the alternative methods have shown to be in theory. For example Markov Clustering is widely recognized as a fast, scalable and easy-to-use method that has been brought to practical applications by a large group of biologists.

2.3.3 Analysis of relational data

Analysis of relational data has drawn attention among researchers from different backgrounds, including machine learning, statistics, inductive logic programming, and databases, and there are different, partly overlapping schools of research active in the field. *Relational data mining* studies methods for knowledge discovery in databases having information about several types of objects. Relational data mining has its roots in Inductive Logic Programming (ILP), an area in the intersection of machine learning and programming languages. [19]

Statistical relational learning (SRL) is a category of methods developed for analyzing relational data with statistical approaches. SRL builds on ideas from probability theory and statistics to address uncertainty while incorporating tools from logic, databases and programming languages to represent structure. Probabilistic relational models (PRMs) are a class of SRLs that can represent rich dependency structures, involving multiple entities and the relations between them, instead of the traditional flat representations [24, 27].

Figure 4 shows an example of a PRM structure. In PRMs the properties of an object are allowed to depend probabilistically both on other properties of that object and on other properties of related objects. The basic goal is to model the uncertainty about the values of the probabilistic attributes of the objects in the given domain. An example application is a recommendation system: based on the attributes of two entities (e.g., user and movie), one wants to predict relational attributes like the preference (e.g., rating of the user for the movie). [24]

Recent examples of PRMs include the Infinite Relational Model (IRM, [35]) and Infinite Hidden Relational Model (IHRM, [65]). The underlying principle in these methods is to infer a stochastic block model of the graph structure, which is in fact very closely related to the block models mentioned in the previous section.

The connection between PRMs and probabilistic models for network data is evident, and it would thus be interesting to see an empirical comparison of their relative performance in similar tasks. As a part of the thesis the multi-relational extension of the ICM model framework is compared to the IHRM on a recommendation task.

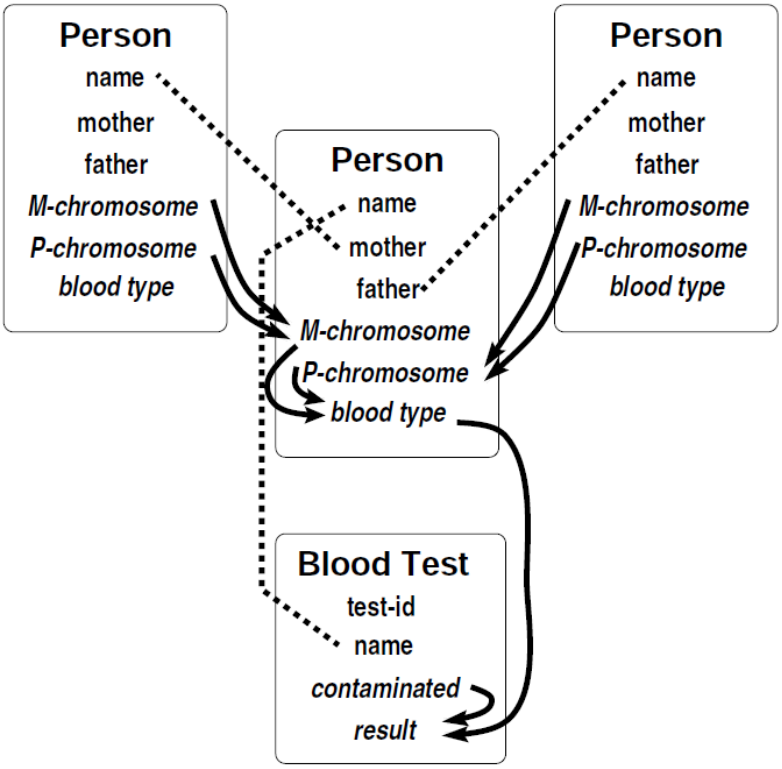

Figure 4: The probabilistic relational model structure for a simple genetics domain. Fixed attributes are shown in regular font and probabilistic attributes are shown in italics. Dotted lines indicate relations between entities and solid arrows indicate probabilistic dependencies. Figure adapted from Friedman et al. [24].

3 Real-world networks

Network analysis has been motivated by the study of many real-world systems that can be presented as networks. Special attention has recently been paid on the comparative study of networks from different origins, with emphasis on their common properties and development of mathematical models that capture these properties. Following [42], the four most prominent categories of real-world networks are social networks, information networks, technological networks, and biological networks. In this thesis biological and social networks are studied. In this section background information about these specific types of real-world networks are given.

Although biological and social systems seem to be different, representing them as networks brings them closer to each other. This allows the study of the structural and functional similarities of the networks. However, as discussed in the end of Section 2.3.1, there seems to be no universal structure across real-world networks. For instance, *geometric random graphs* have been proposed as an alternative to scale-free models for protein interaction networks [46].

It is anyway interesting to see how methods developed for one problem domain can be applied to another, when the data is represented similarly. It seems that, for example, in the case of network clustering, many methods can be successfully applied to several problem domains.

3.1 Biological networks

Biological networks cover a wide set of different types of networks, from food webs of ecosystems to neural networks and biomolecular systems. In bioinformatics the focus is on the cellular level. Examples include metabolic pathways, gene regulatory networks or protein interaction networks. Aims of these analyzes is in general the derivation of new biological knowledge in the form of testable hypotheses of the way cells work.

Basic elements in biological systems are organic molecules, such as proteins and nucleic acids. They interact with each other in order to perform biological processes, and thus form a complex network of interactions. Characteristic to these biomolecular networks is that they are dynamic, making their analysis a challenging task. Other obstacles for research are the physical size of the molecules and the speed of some interactions, making the observation of these systems hard.

From the many types of molecular networks the one involving protein interactions is especially tempting as an application for the generative model framework used in the thesis, because the data is known to be noisy and the detection of connected subgraphs has natural interpretations as biologically relevant modules, as described in the following sections. The other networks, such as gene regulatory networks, are also highly interesting, but they require models based on quite different assumptions.

3.1.1 Protein interaction networks

In the biological part of this thesis protein interaction networks are analyzed. They are widely studied molecular networks of *protein-protein interactions* (PPIs). These can be direct-contact association of protein molecules, but also longer-range interactions through the solution surrounding neighbor proteins. PPIs are important in numerous biological functions, such as signal transduction.

Interactions between proteins have been measure with various techniques, each with their own specific characteristics. In yeast two-hybrid (Y2H) methods protein pairs are tested for possible direct binary interactions, whereas tandem affinity purification followed by mass spectrometric analyses (TAP-MS) captures stable protein complexes. PPI sets obtained with these two methods are very different from each other due to the types of interactions they detect. They also contain a lot of measurement noise — even sets measured with the same method can have relatively low overlap [16].

Recently, Tarassov et al. [56] applied a new technique called protein-fragment complementary assay on a yeast interactome, giving an *in vivo* view of PPIs. This technique is supposed to give a more reliable set of interactions, as it measures them as they really exist in the cell, in contrast to Y2H and TAP-MS methods. Another source of PPIs is the biomedical literature, for example the PubMed publication database (http://www.ncbi.nlm.nih.gov/sites/entrez?db=PubMed) has millions of papers with published and curated biomedical knowledge, such as information about interacting proteins. A bunch of literature mining methods have been applied to the task of extracting relations between proteins, for example by Zhou et al. [68].

Given the differences between measurement techniques and the substantial amount of measurement noise within any technique, it is evident that the resulting networks are incomplete. Many approaches have been presentend for reducing the level of noise and producing more confident networks, for example by Collins et al. [16]. Despite the notable incompleteness of the PPI networks, they have been successfully analyzed with various applications and methods, for example in a protein evolution study [23] and a study of cancer mechanisms [32].

3.1.2 Functional gene modules and protein complexes

An important goal for PPI network analysis is the detection of *functional modules*, that is, sets of genes that are correlated across a set of biological properties and participate in the same biological process. Biological properties can represent any source of information on genes and their products, including gene expression, phenotype and protein interactions. [55]

A closely related term to functional module is *protein complex*, a group of two or more proteins formed through PPIs that acts as a functional module. A hypothetical protein interaction network with densely connected subgraphs is shown in Figure 5. The detection of functional modules and protein complexes is to a large degree an

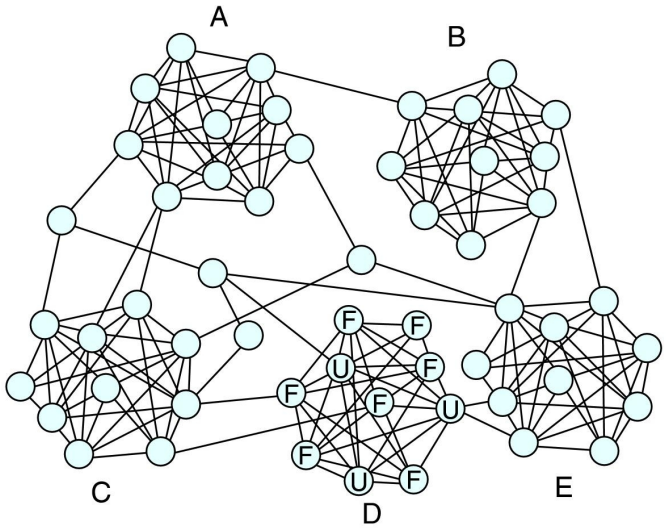

Figure 5: A schematic representation of a hypothetical protein-protein interaction network. Each sphere represents a protein and the connecting lines represent protein-protein interactions. Within an interaction network, smaller local interaction networks or 'clusters' may form (A-E). Proteins in clusters generally have similar functions, allowing prediction of the cellular function of uncharacterized proteins (U in cluster D) from the function of characterized proteins within the cluster (F). Figure adapted from Giorgini et al. [28].

overlapping task in the context of PPI networks. Detected modules and complexes can be used to predict functions for unknown proteins, based on known functions of other proteins in the same module.

Functional modules and protein complexes have been sought from protein interaction data with various clustering-type methods. The methods should be able to handle the extensive noise in the networks. Markov Clustering, presented in Section 2.3.2, has been recognized as a very effective method for this task [11].

In addition to the noisy data, non-stationarity of the modules makes the detection even harder. For example, one protein complex may consist of several subcomplexes, and the combination of subcomplexes may vary when the main complex participates in different cellular processes. The context (e.g., location, state of cell cycle, external conditions) thus acts as an additional dimension in the analysis. In many cases such data is available, and methods that can effectively incorporate this data into the analysis could prove very valuable to systems biology.

3.1.3 Fusion of multiple data sources

Protein interactions are definitely not the only data source that can be studied in order to detect functional modules. Examples of other data sources include gene

expression, protein localization, and protein motif information. Gene expression[3] has been widely studied to detect gene groups that exhibit coherent expression profiles over various conditions.

Although the protein interactions and gene expression are very different as data sources, many attempts to integrate them into a single modeling framework have been proposed recently, for example by Eran Segal [50]. Another approach by Nariai et al. combined even more data [41]. In computational data analysis, merging of heterogeneous data sources is called *data fusion*, and it is becoming very popular especially in bioinformatics.

An assumption that is frequently made when combining PPI data with gene expression data is the one-to-one mapping of genes to proteins, although it is known that one gene may have multiple different products, that is, proteins. A more sophisticated model could take this into account in addition to the context mentioned in the previous section.

3.2 Social networks

A social network is a set of people or groups of people with some pattern of contacts or interactions between them [42]. Examples include friendship networks, scientific networks where documents are linked through citations, and collaboration networks. Analysis of such networks is called *Social networks analysis* (SNA), and it typically aims at modeling the structure of the network and studying how this structure affects the functioning of individuals or groups in the network [63].

Traditional social network data was collected through interviews or questionnaires. This was very labour-intensive, resulting in rather small data sets. More reliable network data has been collected from collaboration information, such as actors acting in the same movies or scientists co-authoring a publication. Another widely used data source has been different kinds of communications records, such as phone calls or emails. [42]

Development of information technology has led to the emergence of very large, even planet-wide social networks, such as Facebook or Myspace, and thus created network data sets that were totally impossible to collect with traditional methods. Such huge data sets are a challenge for computational analysis methods. In addition to the static structure of large social networks, a lot of research activity is on the generation and evolution of such networks.

[3]Gene expression tells the activity level of a gene, as measured by the amount of mRNA molecules found in a cell at a given time. Expression profile is obtained by repeating the measurement under different conditions.

3.2.1 Communities

In social networks analysis, a lot of work is focused on the analysis and detection of *communities*, which is also an important part of this work. Definitions for a community are ambiguous, but in general a community is understood as a group of nodes with a lot of interactions within but less to nodes outside the community [22]. Community structures were illustrated in Figure 3. Sometimes communities are allowed to occur hierarchically or overlap.

Again, many different kinds of approaches have been presented for community detection. *Graph partitioning* methods try to divide the network into connected subgroups by maximixing a given formalization of communities, such as modularity discussed in Section 2.3.2. In general, numerous network clustering methods are applicable to both biological and social networks.

3.2.2 Rich networks

Although social network data are typically represented and studied as binary networks, social systems are rarely as simple. Instead, they exhibit heterogeneous types of relationships between the actors in the network, in other words they are multi-relational in nature. This inherent nature has been used in community detection only recently [13]. Also node attributes, such as word content of scientific documents, have been integrated into modeling [37].

4 Bayesian modeling and probabilistic graphical models

Large collections of noisy data set a challenge for the computational methods that are used. The general modeling framework used here is called *Bayesian* modeling. It offers an explicit way to use probabilities for quantifying uncertainty in inferences based on statistical data analysis. This is an especially tempting property as all the data used in the experiments are known to contain a lot of noise. In this section basics of the Bayesian modeling framework are given, along with a brief introduction to probabilistic graphical models.

4.1 Basics of Bayesian inference

Bayesian inference is the process of fitting a probability model to a set of data and summarizing the result by a probability distribution on the parameters of the model and on unobserved quantities such as predictions for new observations [26].

In Bayesian statistics, probability is used as the fundamental measure of uncertainty. Bayesian methods enable statements to be made about the partial knowledge of a system using probability as yardstick. The guiding principle is that the state of the knowledge about anything unknown is described by a probability distribution. As a classical example, the probability of 'heads' in a coin toss is widely agreed to be 0.5 [26].

4.1.1 Bayes' rule

Bayesian inference begins by setting up a model, providing a *joint probability distribution* $p(\theta, y)$ for the model *parameters* θ and the observed data y. This can be written as a the product of the *prior distribution* $p(\theta)$ and the *data distribution* or *likelihood function* $p(y|\theta)$ respectively: $p(\theta, y) = p(\theta)p(y|\theta)$. Conditioning on the known data y and using the basic property of conditional probability known as the *Bayes' rule*, yields the *posterior* distribution

$$p(\theta|y) = \frac{p(\theta, y)}{p(y)} = \frac{p(\theta)p(y|\theta)}{p(y)} \ , \tag{3}$$

where $p(y) = \int p(\theta)p(y|\theta)d\theta$. An equivalent form of (3) yields the *unnormalized posterior distribution*:

$$p(\theta|y) \propto p(\theta)p(y|\theta) \ . \tag{4}$$

These simple expressions encapsulate the technical core of Bayesian inference: develop the model $p(\theta, y)$ and perform the necessary computations, known as *inference*, to summarize $p(\theta|y)$ in appropriate ways [26].

The simplest way to summarize the posterior $p(\theta|y)$ is the *Maximum a posteriori* (MAP) estimate, a value of θ that maximizes the posterior probability. The MAP

estimate is close to the Maximum likelihood (ML) estimate, the difference being in the incorporation of the prior into the model in the posterior. In Bayesian modeling the prior is always involved, bringing subjective information into the modeling. The effect of the subjective information can sometimes be minimized or eliminated by choosing a *non-informative* prior.

Many statistical applications involve multiple parameters that can be regarded as related or connected in some way by the structure of the problem. It is natural to model such a problem *hierarchically*, with observable outcomes modeled conditionally on certain parameters, which themselves are given a probabilistic specification in terms of further parameters, known as *hyperparameters*. A hierarchical model is even more useful when the data are organized in multiple levels. The model framework used in the thesis is an example of a hierarchical model.

4.1.2 Marginalization

Models for complex data involve a large number of unknown parameters, and it is in dealing with such problems that the Bayesian framework reveals its principal advantages over other inference methods. In practice, one is typically interested in only a part of the model parameters at a time, and thus aims at obtaining the *marginal* posterior distribution of these particular parameters of interest. This is achieved by *marginalizing* over the unwanted *nuisance* parameters as follows [26]:

$$p(\theta_1|y) = \int p(\theta_1, \theta_2|y) d\theta_2 \, . \tag{5}$$

where θ_1 denotes the parameters of interest and θ_2 are the nuisance parameters.

4.1.3 Model selection

Model selection is an important concept in Bayesian modeling. Suppose that the analyzed data arise form one of a set of possible models, $M_1, \ldots, M_k$. Model selection refers to the problem of using data to select one of the possible models M_i. A fully Bayesian way would be to integrate over all $M, \ldots, M_k$, but this is impractical due to computational reasons and the large number of possible models. Instead, the set of possible models is usually restricted and the problem is then to choose a suitable model from the resulting subset.

One way to choose the best model would be to compute the conditional probability $p(D|M)$ (D denotes data), called *evidence*, which is often laborious in practice. Other possibilities for model selection include many approaches used widely in computational data analysis, such as cross validation and different information criteria.

4.2 Parameter inference

The process of finding (marginal) posteriors of model parameters is usually referred to as *inference*, other commonly used notion is parameter optimization. In Bayesian inference, the most conventional parameter inference procedure is random draws from the posterior distribution of the model parameters [26].

In simple cases the posterior $p(\theta|y)$ of the parameters of interest can be computed in analytic form, and draws from the posterior can thus be obtained directly. In practical applications, however, the exact computation of complex models is intractable and the posterior needs to be estimated with approximate methods. Simple possibilities that can be used include evaluating the posterior at a grid of parameters and rejection sampling, see [26] for details. Nevertheless, there is often need for more sophisticated approximate methods. In the following sections some of these are described in more detail.

4.2.1 Expectation-maximization algorithm

The *expectation-maximization* (EM) algorithm is a general technique for finding ML or MAP solutions for probabilistic models having latent variables or assignments. EM is an iterative method which alternates between performing an expectation (E) step, which computes an expectation of the log likelihood with respect to the current estimate of the distribution for the latent variables, and a maximization (M) step, which computes the parameters which maximize the expected log likelihood found on the E step. These parameters are then used to determine the distribution of the latent variables in the next E step. The E and M steps are then repeated until convergence. [9]

4.2.2 Markov chain Monte Carlo methods

Markov chain simulation, also called *Markov chain Monte Carlo* (MCMC), is based on drawing values of θ from approximate distributions, and then correcting those draws to better approximate the target posterior distribution, $p(\theta|y)$. The samples are drawn sequentially, with distribution of the sampled draws depending on the last value drawn. The key to the method's success is that the approximate distributions are improved at each step in the simulation, and they converge to the target distribution. [26]

MCMC simulations are used when it is not possible (or computationally efficient) to sample θ directly from $p(\theta|y)$; instead we sample iteratively in such a way that at each step of the process we expect to draw from a distribution that becomes closer and closer to $p(\theta|y)$. For a wide class of problems this approach appears to be the easiest way to get reliable results, at least when used carefully. [26]

Most commonly used MCMC methods are Metropolis and Metropolis Hastings algorithms, as well as Gibbs sampling. Gibbs sampler, which is in fact a special case

of Metropolis Hastings sampler, is particularly useful in multidimensional problems. Each iteration of the Gibbs sampler cycles through subvectors of the original parameter vector θ, drawing each subset conditional on the value of all other. Formally, at each iteration t, each θ_j is sampled from the conditional distribution given all the other components of θ,

$$p(\theta_j|\theta_{-j}^{t-1}, y) \ . \tag{6}$$

Here θ_{-j}^{t-1} denotes all the components of θ, except for θ_j, at their current values:

$$\theta_{-j}^{t-1} = (\theta_1^t, \ldots, \theta_{j-1}^t, \theta_{j+1}^{t-1}, \ldots, \theta_d^{t-1}) \ , \tag{7}$$

where d is the number of subvectors the parameter vector θ is divided into. This can equal the number of parameters, in which case each single parameter value is updated separately. Some model parameters can be marginalized or integrated away (see Section 4.1.2) before the sampling process.

Convergence. Iterative simulation, such as Gibbs sampling, adds two difficulties to inference using simulation. First, if the iterations have not proceeded long enough, the simulations may be grossly unrepresentative of the target distribution and hence produce bad results. The second problem is the within-sequence correlation of the simulation draws; inference from correlated draws is less precise than from independent draws [26].

Basic solutions to the problems described above are discarding a *burn-in* period from the beginning of each simulation (to assure that the simulation is converged, that is, it is close enough to the target distribution), and then taking samples from the draws with a certain interval (to reduce the correlation between draws). However, deciding a suitable length for the burn-in period and sampling interval is not straightforward. Many estimators have been proposed for monitoring the chain convergence [26].

4.2.3 Variational methods

Variational Bayesian methods are an alternative to sampling methods for making use of a posterior distribution that is computationally too intensive to sample from directly. They can be used to lower bound the marginal likelihood (i.e., "evidence") of models with a view to performing model selection, and often provide an analytical approximation to the parameter posterior probability which is useful for prediction.

In practice, the posterior distribution $p(\theta|y)$ is approximated by a variational distribution $q(\theta)$: $p(\theta|y) \approx q(\theta)$, where q is restricted to a family of distributions simpler than the original posterior. The goal is then to make q very similar with p, as measured with some distance $D(q, p)$ between the two distributions. A typically used distance measure is the Kullback-Leibler (KL) divergence. q is also usually chosen such that it can be *factorized* with respect to disjoint groups θ_i of the parameter vector θ:

$$q(\theta) = \prod_i q_i(\theta_i) \ . \tag{8}$$

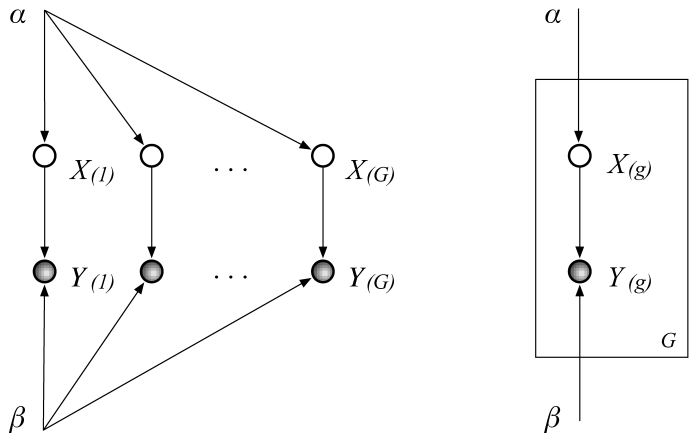

Figure 6: An example of a probabilistic graphical model, with two equivalent representations: **Left:** full model, **right:** a *plate diagram* representation of the model, which will be used later in the thesis. Nodes denote random variables; observed variables are shaded, edges denote dependencies. The box in the plate diagram denotes replicates of random variables that are independent and identically distributed. Figure adapted from Airoldi et al. [3].

4.3 Probabilistic graphical models and generative models

Computational models, such as Bayesian models, are typically represented by a set of mathematical equations which for complicated models tend to be hard to digest even for experienced scientists. A solution that provides a simple way to visualize the structure of a probabilistic model are the *probabilistic graphical models* (PGMs) [9]. In addition to visualizing the model structure, the PGM representation allows the use of graphical methods in the model inference.

PGMs are diagrammatic representations of probability distributions, where each node represents a random variable (or a group of these), and the links express dependency relationships between these variables. The graph then captures the way in which the joint distribution over all of the random variables can be decomposed into a product of factors each depending only on a subset of the variables [9].

An example of a PGM is shown in figure 6, corresponding to the probability model

$$P(Y|X,\alpha,\beta) = \prod_{g}^{G} P(X_g|\alpha) \cdot P(Y_g|X_g,\beta) \ , \tag{9}$$

where Y are observed data, X are model parameters, and α and β are prior parameters.

Probabilistic graphical models can be interpreted as expressing the process by which the observed data arose, in other words, PGMs capture the causal process by which the data was generated. PGMs are thus *generative models* [9]. In a broader sense, the ability to generate synthetic observations from a generative model applies to all Bayesian models. Well-known examples of PGMs include Bayesian networks

and Markov random fields. PGMs have become a popular tool for computational analysis of for example biological data in a variety of domains [3].

4.3.1 Topic models

A well known example of a probabilistic graphical model is the Latent Dirichlet Allocation (LDA) [10] that allows sets of tuples of co-occurring nominal observations to be explained by unobserved groups, or *latent components*, which explain why some parts of the data are similar. LDA is often referred to as the *topic model*, because it was originally applied on document data. Topic model posits that each document is a mixture of a small number of topics and that each word in the document is generated by one of the document's topics.

Important earlier methods here are *latent semantic analysis* (LSA) and *probabilistic latent semantic analysis* (PLSA) [31]. LSA, also known as *latent semantic indexing* (LSI), is a technique used in natural language processing to analyze relationships between a set of documents and the terms they contain. LSA applies matrix operations widely used in linear algebra; it is basically a matrix factorization method.

LDA is a generative version of PLSA, as the original PLSA model did not include proper priors. LDA can thus be seen as the latest, generative probabilistic version of traditional matrix factorization. From another point of view, LDA can be viewed as probabilistic *principal component analysis* (PCA) of discrete data, and hence the name discrete PCA is also used [12].

LDA framework has also been widely applied outside the original document domain, the most interesting version regarding this thesis being the Simple Social Network LDA (SSN-LDA) [67], a modification of LDA for community detection from social networks. The model structure of SSN-LDA is presented in more detail in the next section.

4.3.2 Topic model for networks: SSN-LDA

The assumption behind the Simple Social Network LDA [67] model is that communities are modeled as latent variables in the graphical model and defined as distributions over the social actor space. In practice, the algorithm assigns nodes with similar linking distributions into same clusters.

Due to the modeling assumption SSN-LDA can find both *assortative* and *disassortative structures* from network data. A network is assortative with respect to a property if the property tends to co-occur in connected nodes more often than expected by chance [42]. The opposite, negative correlation in adjacent nodes, is called disassortativity.

Figure 7 shows the SSN-LDA model structure as a plate diagram. A more detailed way of presenting a generative model, such as SSN-LDA, is presenting its generative process step by step. In this way also the symbols in the plate diagram get

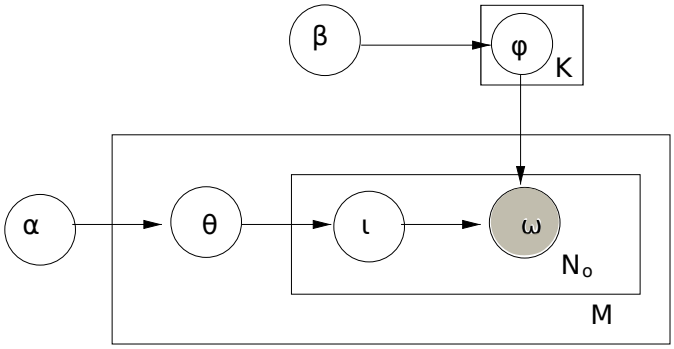

Figure 7: Plate diagram of SSN-LDA. Figure adapted from Zhang et al. [67].

introduced. The generative process behind SSN-LDA is as follows:

1. Initialization
   (a) Generate M multinomial distributions θ_i $i = 1, \ldots, M$, over *latent components* ι, $\iota = 1, \ldots, K$, from a K-dimensional Dirichlet distribution $Dir(\alpha)$.
   (b) Assign a multinomial distribution φ_ι over the nodes i to each component ι by sampling from the Dirichlet distribution $Dir(\beta)$.
2. Link generation (repeat for each link $\omega = 1, \ldots, L$, with sending nodes $i = 1, \ldots, M$)
   (a) Draw a latent component ι from the multinomial θ_i .
   (b) Choose the link endpoint j with probabilities ϕ_ι; set up a directional link ω between nodes i and j .

5 New generative model for network data

This section presents the methods and models developed and used in this thesis, starting from the basic Interaction Component Model framework and proceeding to the extensions and improvements. For clarity, some technical details are presented in a separate Appendix.

The general motivation for developing a new network model is related to the SSN-LDA method described in the previous section. SSN-LDA does not discriminate between the assortative and disassortative property of networks. However, community is clearly an assortative property. If the specific goal of network analysis is to detect communities, then a model designed explicitly for that kind of structure in the data could perform better than a more general model due to the lower number of parameters to be estimated. The new model is thus designed as a modification of SSN-LDA that is specialized to model assortative community structure.

5.1 Generative model for interactions

The methods in this thesis are based on a generative probabilistic model for graphs called Interaction Component Model (ICM; [53]). The model assumes a latent component structure and assigns each edge on the graph to one of these components. Based on the edge assignments, one can then infer the component membership probabilities of the nodes. Depending on the application, the components can be interpreted as communities or protein complexes, or in general any densely connected subgraphs. The simplest model variant for binary interactions is denoted *ICMc*, c standing for communities.

5.1.1 Model framework

A plate diagram of ICMc is shown in Figure 8A. The generative process behind the ICMc is as follows:

1. Initialization
   (a) Generate a multinomial distribution θ for the latent components z from a Dirichlet distribution $\mathrm{Dir}(\alpha)$.
   (b) For each component z, generate a multinomial distribution ϕ_z over the nodes i from a Dirichlet distribution $\mathrm{Dir}(\beta)$.
2. Link generation
   (a) Draw a component z from θ .
   (b) Generate a link by drawing its end nodes, i and j, independently from each other, from ϕ_z .

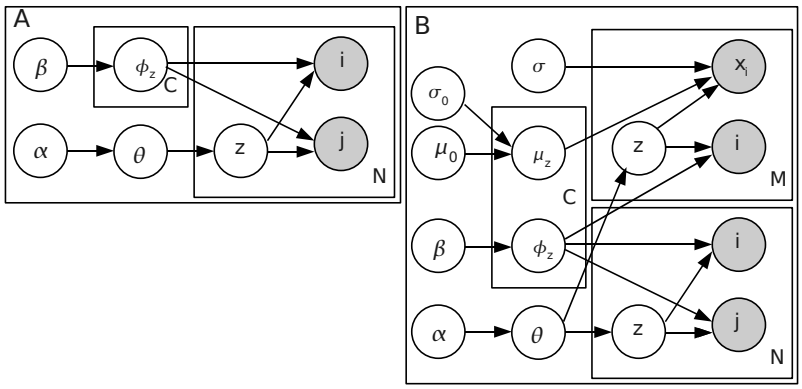

Figure 8: **(A)** Plate diagram of the basic ICMc model. **(B)** Plate diagram of the ICMg2 model.

5.1.2 Equations and inference with collapsed Gibbs sampling

The likelihood of network data can now be formulated as the product of the probabilities of single, independently generated links l as follows.

$$p(L, Z|\phi, \theta) = \prod_{l}^{N} \theta_{z(l)} \phi_{z(l)i(l)} \phi_{z(l)j(l)} = \prod_{z}^{C} \theta_z^{n_z} \prod_{iz}^{MC} \phi_{zi}^{q_{zi}} \ , \tag{10}$$

where N, C and M are the number of links, components and nodes, respectively, L denotes link data, Z denotes all components assignments of the links, and in the latter expression n_z is a count of links assigned to each component, and q_{zi} is a count of component-node co-occurrences.

For parameter optimization, a variant of Gibbs sampling is used, known as *collapsed Gibbs*, in which some model parameters are marginalized out. In particular, the multinomial distribution parameters θ and ϕ_z are integrated out following Griffiths and Steyvers [29], leaving only the component assignments of each link, which we are interested in. The details of the marginalization are presented in Appendix A.1.

Finally, the marginalized probability is separated into link-wise factors, and the probability of one left-out link l_0 to be assigned to component z_0, given the assignments of all other links, becomes

$$p(l_0, z_0|L', Z', \alpha, \beta) = \frac{n'_{z_0} + \alpha}{N' + C\alpha} \cdot \frac{(q'_{z_0 i_0} + \beta)(q'_{z_0 j_0} + \beta)}{(2n'_{z_0} + 1 + M\beta)(2n'_{z_0} + M\beta)} \ , \tag{11}$$

where counts n', q' and N' denote the counts as they were if the link l_0 was nonexistent.

The probabilities (11) can then be used to sample a new component z for a left-out link, with the probabilities $p(z|l_0, L', Z', \alpha, \beta) = u_z/u.$, the denominator using the dot notation for the sum. A Gibbs iteration follows by leaving one link out at a time, and sampling a new latent component for it as above.

5.1.3 Inferring the results

For clustering applications, the aim is to infer the component memberhips of nodes, that is, probabilities $p(z|i)$. From Bayes rule we obtain

$$p(z|i) = \frac{\theta_z \phi_{zi}}{\sum_{z'} \theta_{z'} \phi_{z'i}} \,. \tag{12}$$

This is, however, somewhat laborious to compute, and can in practice be approximated well with

$$p(z|i) \approx \frac{q_{zi}}{\sum_{z'} q_{z'i}} \,. \tag{13}$$

For prediction applications, one can reconstruct the original parameters θ and ϕ from the expected values of the marginalized parameters as

$$\hat{\theta} = \frac{n_z + \alpha}{\sum_{z'} n_{z'} + C\alpha} \tag{14}$$

and

$$\hat{\phi}_z = \frac{q_{zi} + \beta}{\sum_{i'} qzi' + M\beta} \,. \tag{15}$$

5.1.4 Infinite ICMc

In the basic ICMc the number of components has to be predefined. In many practical applications it would, however, be useful to infer the number based on data. Standard model selection methods, such as different information criteria or cross validation could in principle be used, but a more suitable method that fits the ICM framework easily is the Dirichlet Process (DP) prior [58].

The DP prior has an infinite number of components in principle, but only a finite number is realized in practice during the inference. Because the DP model is not used in the experiments in the thesis, the full derivation of the corresponding Gibbs sampler is omitted here. The resulting collapsed sampling equation, replacing the finite equation (11), is

$$p(l_0, z_0|L', Z', \alpha, \beta) = \frac{C(n'_{z_0} + \alpha)}{N' + C\alpha} \cdot \frac{(q'_{z_0 i_0} + \beta)(q'_{z_0 j_0} + \beta)}{(2n'_{z_0} + 1 + M\beta)(2n'_{z_0} + M\beta)} \,, \tag{16}$$

where the function $C(n_z + \alpha) \equiv n_z$ if $n_z \neq 0$ and $C(0, \alpha) = \alpha$. In the latter case, a new component is generated in the process.

5.1.5 ICM and related models

The ICM framework has its roots in topic models, and it is therefore not surprising that the generative process behind ICMc is very close to that of SSN-LDA. There is, though, a notable difference in the modeling assumptions: SSN-LDA groups nodes

that share similar link distributions, and can thus detect more complex structures than ICMc, which is only designed to detect community-like structures.

From the other network models mentioned in Section 2.3.2 the stochastic block model is also closely related to ICMc. In fact, the ICM framework was extended to model block structures in a very recent work [45]. This extensions is not so relevant for this thesis and is thus not covered here.

5.2 Generative model for protein interactions

In the biological part of this thesis the basic ICMc is applied to protein interaction networks in order to seek functional modules of protein complexes. Although the model was originally designed to detect communities in social networks, the modeling assumptions are very similar to the common interpretation of functional modules as densely connected subgraphs. It is thus interesting to see how well ICM performs in the biological task.

5.3 Incorporating gene expression data into the analysis

In addition to the network clustering task described above, ways to include functional data about the nodes into the model are introduced in the thesis. The idea is that functional data for the genes can improve the detection of the modules. In particular, genes with functional similarity should be included in the same modules. In subsequent sections, two ways to combine protein interaction data with gene expression data are presented.

5.3.1 Transforming expression profiles into relations

In the first model variant including gene expression, denoted as *ICMg1*, gene expression data is transformed into relations that describe the functional similarity of the genes. These relations are then added to the original PPI network. In practice, the Pearson correlation of expression for each pair of genes is computed, and all pairs where the correlation exceeds 0.85 are treated as additional links[4].

The motivation is that both the existence of protein-protein interactions and potential co-regulation inferred from the correlation links give evidence of functional relatedness of the genes. This approach is similar to the one used by Ulitsky and Shamir [62], apart from the fact that we do not make any difference between the two types of links.

[4]The same cutoff value as in [41].

5.3.2 Generative process including gene expression profiles

In the next model variant, denoted as *ICMg2*, the same idea is taken further by including the gene expression data into the generative model. In practice this is achieved by generating for each component a specific expression profile from a Gaussian distribution. Node-specific expression profiles are then generated from Gaussian distributions with component-specific means. The underlying assumption is that genes in a component should share similar expression profiles in addition to being strongly connected.

Plate diagram of this model variant is shown in Figure 8B. The generative process goes as follows:

1. Initialization
   (a) Generate a multinomial distribution θ for components z from a Dirichlet distribution $\mathrm{Dir}(\alpha)$.
   (b) For each component z generate a multinomial distribution ϕ_z over nodes i from a Dirichlet distribution $\mathrm{Dir}(\beta)$.
   (c) For each component z draw a mean vector of expression profiles $\bar{\mu_z}$ from a prior multivariate Normal distribution $N(\bar{\mu}_0, V_0)$ with zero mean $\bar{\mu}_0 = 0$ and diagonal covariance matrix $V_0 = \sigma_0^2 I$.
2. Link generation
   (a) Draw a component z from θ .
   (b) Generate a link by drawing its end nodes, i and j, from ϕ_z .
3. Node generation
   (a) Draw a component z from θ .
   (b) Generate a node k from ϕ_z .
   (c) Generate data vector $\bar{x}_k$ from a multivariate normal distribution $N(\bar{x}_k|\bar{\mu}_z, V)$ with component-specific mean $\bar{\mu}_z$ and covariance matrix $V = \sigma^2 I$.

Note that each node could be generated multiple times in step 3. This is allowed in the generative process to simplify computations, and it is not supposed to have any practical effect here , as each gene has exactly one expression profile in the data. In other application domains it could be an advantage.

5.3.3 Equations and inference with collapsed Gibbs sampling

Joint probability of ICMg2 is a product of the link-specific probabilities that are the same as with the basic ICM, and the normally distributed expression profile

probabilities with priors for the component-specific means. This becomes

$$p(L, X, Z, \phi, \mu, \theta) = D_1(\alpha, \beta, \sigma^2) \prod_z^C \theta_z^{n_z + m_z + \alpha - 1} \prod_{iz}^{MC} \phi_{zi}^{q_{zi} + \beta - 1} \times \prod_z^C \left[N(\bar{\mu}_z | \bar{\mu}_0, V_0) \prod_{z(k)=z}^{m_z} N(\bar{x}_k | \bar{\mu}_z, V) \right], \tag{17}$$

where X denotes the node data, m_z is the number of nodes assigned to component z, $\bar{\mu}_0$ and $\bar{\mu}_z$ are the prior and component-specific means, respectively, V_0 and V are the prior and data covariance matrices, respectively.

The collapsed Gibbs sampler is derived analogously to the basic ICM, the marginalization now including the component-specific means $\bar{\mu}_z$. Details are presented in Appendix A.2. The sampler now involves the separation of the marginalized probability into link-wise and node-wise factors. The probability of the left-out link l_0 to be assigned to component z_0 becomes

$$p(l_0, z_0 | L', Z', \alpha, \beta) = \frac{n'_{z_0} + m_{z_0} + \alpha}{N' + M + C\alpha} \cdot \frac{(q'_{z_0 i_0} + \beta)(q'_{z_0 j_0} + \beta)}{(2n'_{z_0} + m_{z_0} + 1 + M\beta)(2n'_{z_0} + m_{z_0} + M\beta)}. \tag{18}$$

Likewise, the probability of the left-out node-data $\bar{x}_0$ to be assigned to component z_0 is

$$p(\bar{x}_0, z_0 | X', Z', \alpha, \beta) = \frac{(n_{z_0} + m'_{z_0} + \alpha)(q'_{z_0 k_0} + \beta)}{2n_{z_0} + m'_{z_0} + M'\beta} \cdot \left[\frac{|S|}{|S'|} \right]^{1/2} \times \exp\left[-\frac{1}{2} \bar{x}_0^T V^{-1} \bar{x}_0 + \frac{1}{2} A^T S^{-1} A - A'^T S'^{-1} A' \right]. \tag{19}$$

For notation details, see Appendix A.2. Gibbs sampler iterates by sampling component assignments in turn for links and expression profiles, separating one data point at a time and using equations 18 and 19 as with the basic ICM. Inference of the results is analogous to ICMc, see Section 5.1.3.

5.4 ICM for multi-relational data

The idea of extending the ICM model framework to handle multi-relational data was originally presented by Sinkkonen et al. [54]. The approach is basically similar to the ICMg2, which is actually already a multi-relational model, as similarity of expression profiles is seen as an additional relation between the genes.

Multi-relational ICM assumes that also the node-wise data is multinomial count data, instead of the Gaussian data as in ICMg2. The model framework is easily applicable to any problem setting, which satisfies the modeling assumptions of data representable as counts. In the model, the global component structure is similar to that of the basic ICM, and then component-specific multinomial distributions are

assumed over each data type. Corresponding collapsed Gibbs samplers are derived separately for each data type, and the overall iteration proceeds through all of these in turn. In a case study presented in the paper, [54], the model was applied on a simple case of citation data with additional word content data for each document. The resulting combined model of citation and content data outperformed approaches that use either data source alone.

In this thesis results are reported from an experiment, where the multi-relational ICM is applied on the MovieLens data [49]. The data consists of two types of entities, users and movies, and several attributes for both of these. There is also a relation between the entities, namely rating given by a user to a movie, with two possible values, positive and negative. A relational schema for this is presented later in Section 7.1.2. The model details are omitted for brevity.

5.5 Improved inference

In addition to extending and applying the ICM framework to new kinds of data the thesis includes improvements made into the inference procedure. First, the hyperparameters α and β are found to have a notable effect on the clustering results, and should thus be chosen well [7]. Recently, Asuncion et al. highlighted the importance of proper estimation of hyperparameters in topic models in general [6].

The hyperparameters can of course be optimized manually if the results can be validated somehow, for example based on some ground truth classes of the nodes. But in general, a more sophisticated approach would be to estimate the hyperparameters from the data, for which one solution is introduced here. Additional improvements are made on convergence monitoring and on assessing different Gibbs sampler chains.

5.5.1 Hyperparameter estimation by sampling

Many approaches have been presented for estimating the hyperparameters of LDA-type models, for example by Thomas Minka [40]. The sampling scheme used in this thesis for the hyperparameters α and β goes as follows:

1. Find the maximum of the posterior distribution with Newton's method. It is important to stay in a log-concave region, so at each step we have to check whether the 2nd derivative is negative, otherwise the current parameter value is halved.

2. Metropolis sampling. Use a normal approximation of the posterior evaluated at the MAP estimate, obtained with Newton, as a proposal distribution. A 2.5-fold standard deviation for the normal distribution is used to achieve better performance. Metropolis is currently iterated 15 times, rejecting the move if the parameter value is not positive.

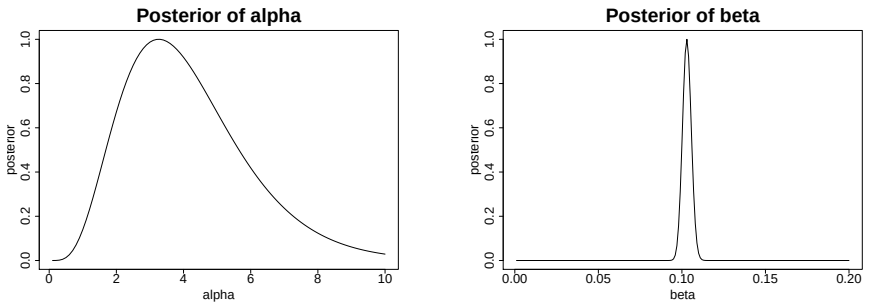

Figure 9: Posteriors of the hyperparameters α (left) and β (right) for ICMc from one run on the Cora dataset.

The same sampling procedure is used for alpha and beta. It requires evaluating the 1st and 2nd derivatives of the log-posterior at each Newton step and then the log-posterior itself at each Metropolis step.

The posterior of α (with a Dirichlet prior) is

$$p(\alpha) \propto p(\alpha)p(L|\alpha) = p(\alpha) \cdot \frac{\Gamma(C\alpha)}{\Gamma(\alpha)^C} \frac{\prod_z \Gamma(n_z + \alpha)}{\Gamma(N + C\alpha)} \tag{20}$$

and β's posterior

$$p(\beta) \propto p(\beta)p(L|\beta) = p(\beta) \cdot \prod_z \frac{\Gamma(M\beta)}{\Gamma(\beta)^M} \frac{\prod_i \Gamma(q_{zi} + \beta)}{\Gamma(2n_z + M\beta)} \,. \tag{21}$$

The logarithmic posteriors and their derivatives, which are needed for the computations, are presented in Appendix A.3. The corresponding equations are easy to derive for alpha in the case of the Dirichlet Process prior, but are omitted here.

A simpler approach would be to use the MAP estimate directly for the hyperparameters. The performance would then depend on the shape of the posterior of the parameter. Figure 9 shows the posteriors of the hyperparameters, as computed based on a point-estimate of the assignment of links to the components in the end of a sampling procedure. So in fact these are not true posteriors, but conditional posteriors given the component assignments z. Conditional posterior of alpha seems to be quite wide, whereas for beta the posterior is extremely narrow. In the latter case the MAP estimate would probably perform well.

5.5.2 Estimating convergence

Monitoring the convergence of MCMC chains is an important part of the inference, as noted in Section 4.2.2. One possible measure is the marginal likelihood, but it is hard and laborious to estimate from the MCMC samples.

Instead a log-probability of the left-out link, $\log p(z|l_0, L', Z', \alpha, \beta)$ is used. In addition to monitoring convergence this is proposed and tested as a measure of the

goodness of MCMC chains, indicating that choosing a chain with the highest average log-probability would give better results. Although the quantity itself is non-standard, note its interpretation as a leave-one-out estimate for the entropy of the predictive distribution for new links. It requires little additional computing, as the probability is already computed during the sampling.

6 Experiments with biological data

In the first experimental part of the thesis the ICM framework is applied to the study of biological network data. This section contains the research questions, used methods, data sets and evaluation criteria, followed by results and brief conclusions.

6.1 Detecting functional gene modules from biological networks

In this experiment the task is to detect functional modules from protein interaction networks and gene expression data. There are two research questions:

- Can the basic ICM find biologically relevant gene modules from protein interaction data?
- How does the integration of gene expression data into the analysis affect the detection of functional modules?

6.1.1 Methods

From the ICM variants the basic ICMc using PPI data and both ICMg1 and ICMg2 that utilize gene expression data are used in the experiment. In order to assess their module detection performance, the proposed methods are compared to two recently published methods, the Hidden Modular Random Field (HMoF; [52]) and Matisse [62]. Both of these utilize protein interaction and gene expression data, though with quite different ways.

In the HMoF method, the network is modeled with a modularity-optimization algorithm and gene expression with k-means clustering, and a specific parameter ω is used to control the weighting between these two data types. In Matisse, the gene expression is transformed into similarity values between genes. An algorithm is then devised to detect node groups that are strongly connected and highly similar. Common to both of these methods is that they do not directly take the noise in the PPI data into account.

Matisse differs from the other methods in the sense that it leaves some genes out from the clustering and also infers the number of clusters automatically. Due to the probabilistic nature of all the models, the number of clusters could be set automatically in several well-justifiable ways, such as cross-validation and different types of information criteria (see, e.g., Bishop [9] for standard model selection methods). For ICM variants a natural option would be to use a Dirichlet Process prior for the component distribution. Dirichlet Process is a common non-parametric prior for estimating the number of components based on the data [58].

However, since implementation of comparable model complexity control methods would be laborious in practice for some of the methods, the number of clusters of

the other methods is fixed to the median of 20 Matisse runs to bias the results in favor of Matisse, to make sure that the result is not due to the additional degrees of freedom ICM has in choosing the cluster sizes. Each method was run 20 times to obtain confidence intervals, resulting in a different set of clustered genes for each Matisse run.

The models have some tunable parameters which affect their performance. All these parameter values were chosen a priori and not optimized. Our ICM models have two hyperparameters controlling the component distribution and node distributions within components, respectively. Based on earlier studies we set the hyperparameter values to $\alpha = 10$ and $\beta = 0.01$. This study was conducted before the development of the hyperparameter estimation procedure and it is hence not used here. The model variant ICMg2 has three additional hyperparameters for generating the expression data, which we set to $\mu_0 = 0$, $\sigma_0^2 = 1$ and $\sigma^2 = 0.1$ to describe small variations around the base value of zero.

The number of clusters for all other methods than Matisse was set to the median of 20 Matisse runs on both datasets, resulting in 24 and 25 clusters in the osmotic shock response and DNA damage data sets, respectively. HMoF has a weight parameter ω defining the relative weighting between the expression and network data in the model. This was fixed to $\omega = 0.2$ as in the original paper [52]. Matisse was run with the default parameters given in its implementation.

6.1.2 Data sets and evaluation

The PPI data set is obtained by pooling two yeast *Saccharomyces cerevisiae* data sets, [62], [41], which are originally obtained from various public databases. The first gene expression data set is the Osmotic shock response (OSR) set [44] and the other one is a DNA damage (DNAD) set [25]. Since the implementations of all methods do not support missing samples in the sense that either expression or PPI links would be completely missing from some genes, subsets without such missing data are analyzed here.

Two combined data sets are obtained, one with 1711 genes, 10 250 interactions and 133 observations of gene expression (OSR), and another with 1823 genes, 12 382 interactions and 52 gene expression observations (DNAD). Pooling the expression links with the original PPI's for the ICMg1 results in 14 256 (OSR) and 15 547 (DNAD) links in total. Missing values in the expression data were interpolated using the 10-nearest neighbor method [60].

For evaluating detected functional modules, two measures are used: Gene Ontology (GO) [5] enrichment analysis and protein complex overlap. Gene Ontology contains manual annotations of genes to known biological process classes. These classes can be used as a reference set for validating obtained gene modules. In GO enrichment analysis a hypergeometric p-value is computed for each pair of found module and GO class [48]. Lower p-value means that the modules contain more of the same gene class than would be probable if they were generated randomly. A common

approach is then to treat all pairs under a certain cutoff-value as enriched, and a higher number of enriched modules and GO classes is then considered as a better clustering. In this thesis the Fisher exact test [48] is used and the number of enriched modules and GO classes are computed on a range of p-values ($p = \{10^{-1}, ..., 10^{-10}\}$).

As mentioned earlier, functional modules are closely related to protein complexes. They can thus be used as an additional validation of the modules, by computing how well the modules overlap with known complexes. For the analysis, a set of known complexes were obtained from the Comprehensive Yeast Genome Database at MIPS [30]. The total number of complexes in the used MIPS collection is 267. The number of protein complexes existing in our datasets with at least 2 proteins was 95 and 143 for OSMO and DNAD, respectively. Out of these, 33 and 46 contained at least five proteins.

6.2 Results

The task in the biological experiment was to detect relevant functional modules from combinations of protein interaction and gene expression data. In order to evaluate the obtained modules, they are compared to two known sets of genes and proteins, Gene Ontology annotations and protein complexes. Although these are manually curated sets, they represent only part of the truth, and should thus be used with care. They are, however, widely used in validating clusterings for genes and proteins, and are supposed to give reliable results in comparative studies.

The results of the GO enrichment analysis are shown in Figure 10. It shows the number of enriched modules and GO classes as a function of the cutoff p-value for enrichments. Matisse does not perform as well as the other methods in the enrichment analysis. The other four methods perform about equally well in the Osmotic shock response data set, but in the DNA damage data set the three ICM-based methods outperform HMoF as well.

Second, the overlap of the modules with known protein complexes was measured. From the results, shown in Figure 11, it is evident that the first four methods find a significant proportion of the protein complexes with the ICM variant outperforming HMoF to some extent, whereas Matisse's performance is clearly worse.

6.3 Conclusions

As the first part of the thesis the ICM framework was applied to a completely new application domain, biomolecular networks of protein interactions. The particular task was to detect biologically relevant functional gene modules. First, the goal was to evaluate and compare the performance of the basic ICM applied on a plain PPI graph in the task. The results from two experiments on yeast data show that the model framework in general outperforms recently introduced alternatives, which do not directly model the noise in the interactions.

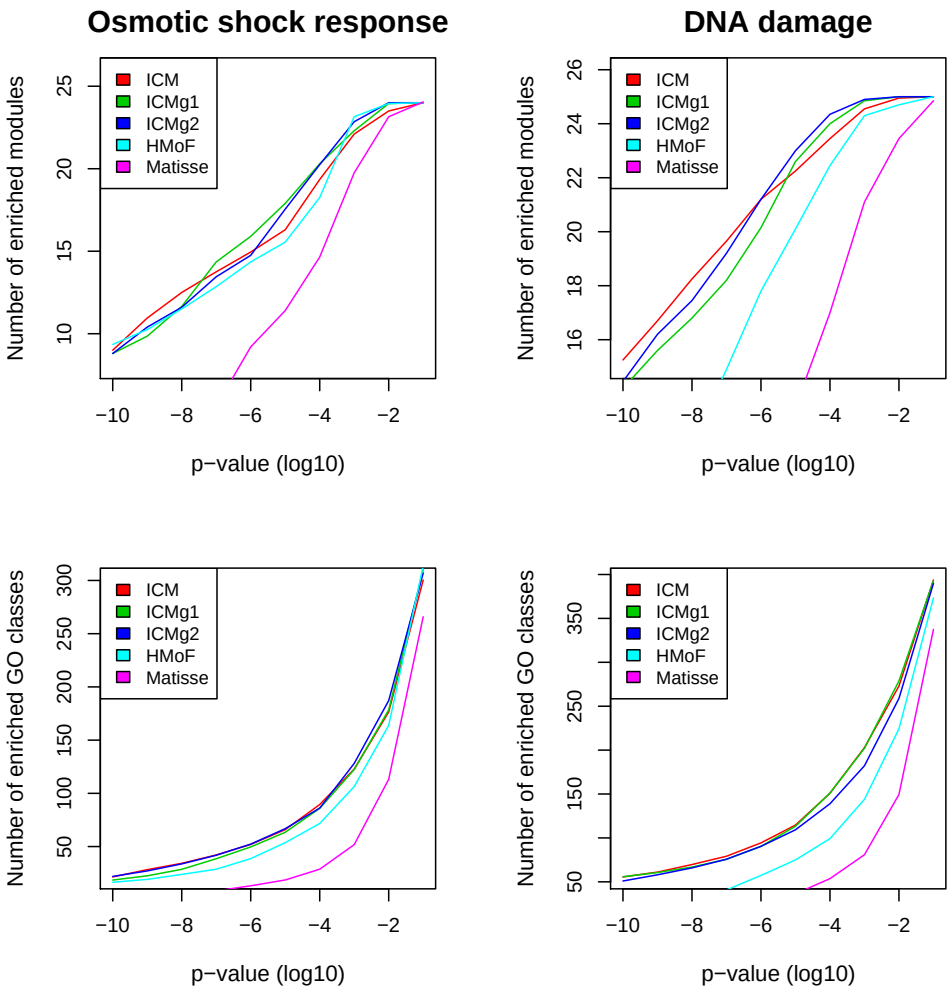

Figure 10: Gene Ontology enrichment results. The number of enriched modules and GO classes as a function of the hypergeometric p-value cutoff. Top row: The number of modules in which at least one GO class is enriched. Bottom row: The number of GO classes enriched in at least one module. Left: Osmotic shock response data. Right: DNA damage data. All values are means over the 20 runs. More enrichments is better.

In addition, two different approaches were introduced for combining gene expression data into the analysis. The combined approaches again outperform the two reference methods, but when compared to the ICM variant without expression data there are no clear differences in one way or the other. So the benefit of the proposed integrative approaches involving gene expression data compared to using plain protein interactions remains questionable.

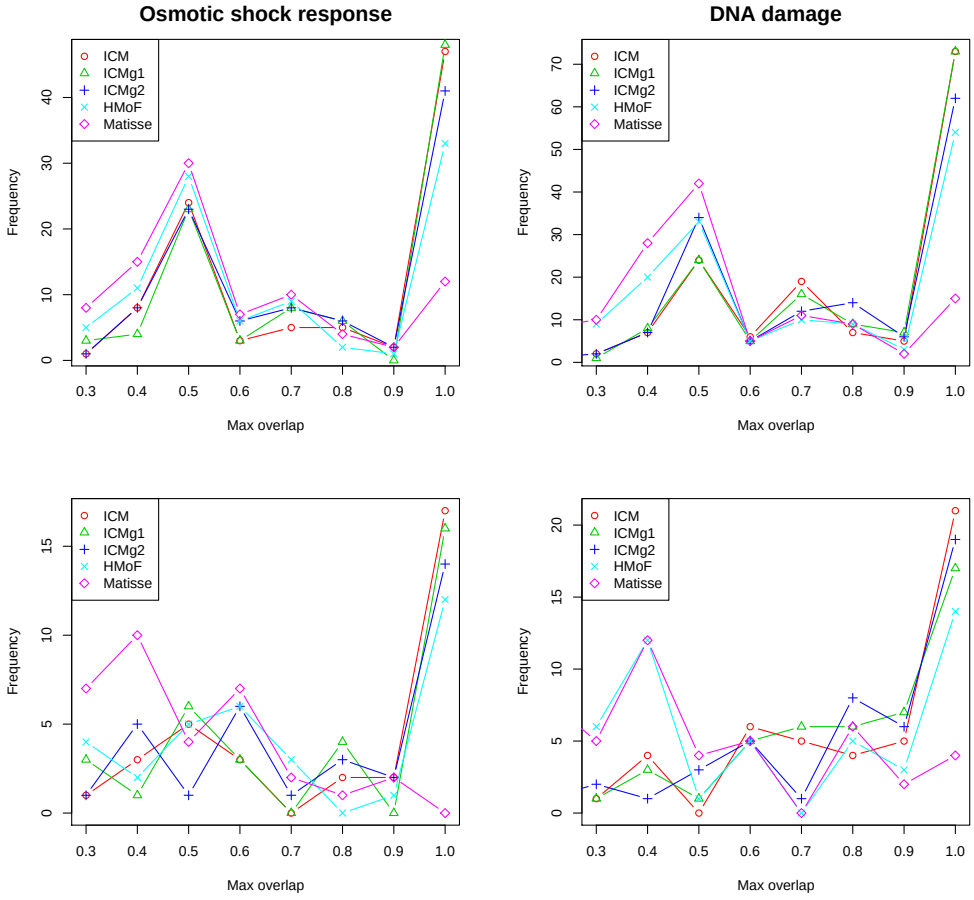

Figure 11: Protein complex overlap. The number of protein complexes (y-axis) with a specific degree of overlap (x-axis). Top: complexes with at least 2 proteins, bottom: complexes with at least 5 proteins, left: Osmotic shock response data, right: DNA damage data. The results in which the area under the curve is concentrated on the right end are better, as this indicates higher overlap with the protein complexes.

7 Experiments with social network data

7.1 Clustering medium-scale social networks

In the second experimental part the task is to detect communities from medium-scale social networks. The experiments were designed to answer the following questions:

- How does sampling of the hyperparameters α and β affect the clustering results with ICM?
- Can the introduced convergence estimator be used for evaluating MCMC chains in a clustering task?
- How can the model be applied on a recommendation task in a multi-relational setting and what is the benefit of node attributes here?

7.1.1 Methods

Three variants of the ICM framework are used here: basic ICMc, and ICMc with fixed (f) and sampled (s) hyperparameters. With multi-relational data the corresponding multi-relational model is used. For comparison a set of different methods are used in the different tasks. In basic community detection, ICM is compared with SSN-LDA and spectral clustering, described in Sections 4.3.2 and 2.3.2, respectively. Parameter inference for SSN-LDA was performed with an analogous collapsed Gibbs sampler to that of ICM. In the case of multi-relational data ICM is compared with the *Infinite hidden relational model* (IHRM; [65]).

All models except the spectral clustering, which is deterministic, were run ten times for each data set to get information about the variation of the results between runs. The number of components for the methods was set to match the known ground truth, to allow straightforward evaluation. The hyperparameter values for ICMc and SSN-LDA were chosen based on earlier studies [7] to give good results, see Table 1 for chosen parameter values.

Table 1: Modeling parameters for medium-scale networks. Network characteristics and modeling parameters for the medium-size networks. In the table, I is the number of nodes in the network, L is the number of edges, Ct is the number of ground truth classes, Cs is the number of clusters sought by the methods and α and β are the hyperparameters of the models.

					ICMc		SSN-LDA	
Network	I	L	Ct	Cs	α	β	α	β
Cora	2 485	5 068	7	7	0.143	0.02	0.143	0.025
Citeseer	2 110	3 668	6	6	0.166	0.04	0.166	0.006
Polblogs	1 222	16 714	2	2	0.5	0.003	0.5	0.4

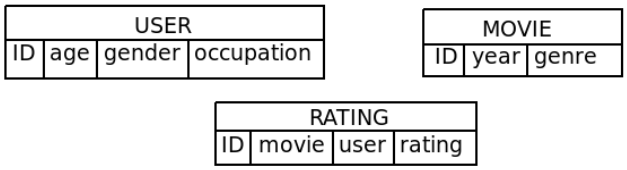

Figure 12: Relational schema of the MovieLens data.

7.1.2 Data sets and evaluation

The Cora and CiteSeer datasets [51] consist of citations between scientific publications. Nodes outside the giant component of the network were removed, and potential directional links were symmetrized. After preprocessing, the Cora dataset had 2,110 papers in seven bibliographical categories, while the CiteSeer dataset had 2,485 papers in six categories. The Polblogs dataset [1], recorded in 2005, has hyperlinks between 1,222 weblogs on US politics. Summary of the data sets is shown in Table 1.

For evaluating the clustering result perplexity is used. It is a measure of the ability of a model to recover an underlying nominal category, and commonly used, for instance, in natural language processing. Perplexity is here applied to the confusion matrix formed of the evaluation samples, that is, to the table of frequencies with standard classes of the samples as columns (c), and the model-given components or clusters as rows (m). Perplexity for the evaluation sample is then defined as

$$\text{perplexity} = 2^{-\frac{\sum_l \log \hat{p}(c_l|m_l)}{N}} , \tag{22}$$

where N is the number of evaluated data samples, indexed by l, and c_l and m_l are their class and component, respectively. The probabilities $\hat{p}(c|m)$ are empirical probabilities, computed by normalizing the rows of the confusion matrix.

Perplexity is a monotonic function of the empirical conditional information $H(C|M)$, which has been shown to be a good measure for clustering by Meilă [39]. From the two-way measure proposed by Meilă, only the other "way" is needed, because the other corresponds to the fixed ground truth.

In the multi-relational experiment the ICM is applied on the MovieLens data [49], and the task is to predict user ratings for movies, based on training data. The model variant is denoted Simple relational component model (SRCM). An illustration of the MovieLens data is shown in figure 12 as a relational schema.

In the data there were 702 users and 603 movies, with on average 112 ratings per user. Ratings were binarized into positive and negative ones, the threshold being user average. For held-out users, 156 of the 702, twenty ratings were used to predict the rest, and the overall average accuracy of these predictions is reported. In the attribute setup, year and genre of the movies, and age, gender, and occupation of the users were added to the model as independent (movie, attribute) or (user, attribute) co-occurrences.

7.2 Results

Results from the experiment with three medium-scale social networks are presented in Figure 13. The probabilistic methods show better or equal performance to spectral clustering. On the Polblogs, spectral clustering resulted in the clustering with 6 nodes in one and all other nodes in an other cluster, giving a very poor perplexity score (not shown). After removing the 6 nodes spectral performed very well, as shown in the results with a cross.

With the citation datasets ICMc with sampled hyperparameters was marginally better than SSN-LDA, but there is no significant difference for Polblogs. With fixed hyperparameters ICMc performs about equally to SSN-LDA. All the probabilistic methods show a large variation between the runs, indicating a need for good ways to evaluate different sampling chains.

From the perplexity results in Figure 13 it seems that the ICMc with hyperparameters generally outperforms the one with fixed parameters, suggesting that the used estimation procedure is effective.

Figure 14 shows a plot of the average leave-out log probability from one run on the Citeseer data, showing that although the estimator has reached a stationary level after a few thousand iterations, there is clear variation between single samples. Moreover, subsequent samples seem to be highly correlated with each other. These results indicate that the proposed measure can be used as a rough convergence estimator. One should still remember to a lot of samples with a large enough interval for reliable inference.

The values of the estimator are plotted against the perplexities for each run of the hyperparameter-sampled ICMc in the insets of Figure 13. Corresponding correlations are: Cora: -0.47, Citeseer: -0.68, and Polblogs: +0.61. Additionally, the best ICMc(s) run according to the measure is marked by the open diamond in the performance figures. The measure is able to choose a relatively good model for all datasets except the Polblogs, where the overall differences between the runs was very small.

Finally, the prediction results for the MovieLens-experiment are shown in Figure 15. IHRM is better but the less complex SRCM performs well too. Both methods have considerable variance between runs, that is, convergence to a local area of the posterior mass. This experiment does not give evidence of any predictive benefit from adding margin attributes of users and movies to either of the models, but see [65] which reports a benefit from the attributes, for the IHRM.

7.3 Conclusions

As the second part of the thesis the model was extended in order to improve community detection on social networks. The experiments on a set of medium-scale social networks show that estimating the hyperparameters improves clustering results compared to using fixed hyperparameter values. This indicates that the choice of the hyperparameters is an important factor when the model framework is used

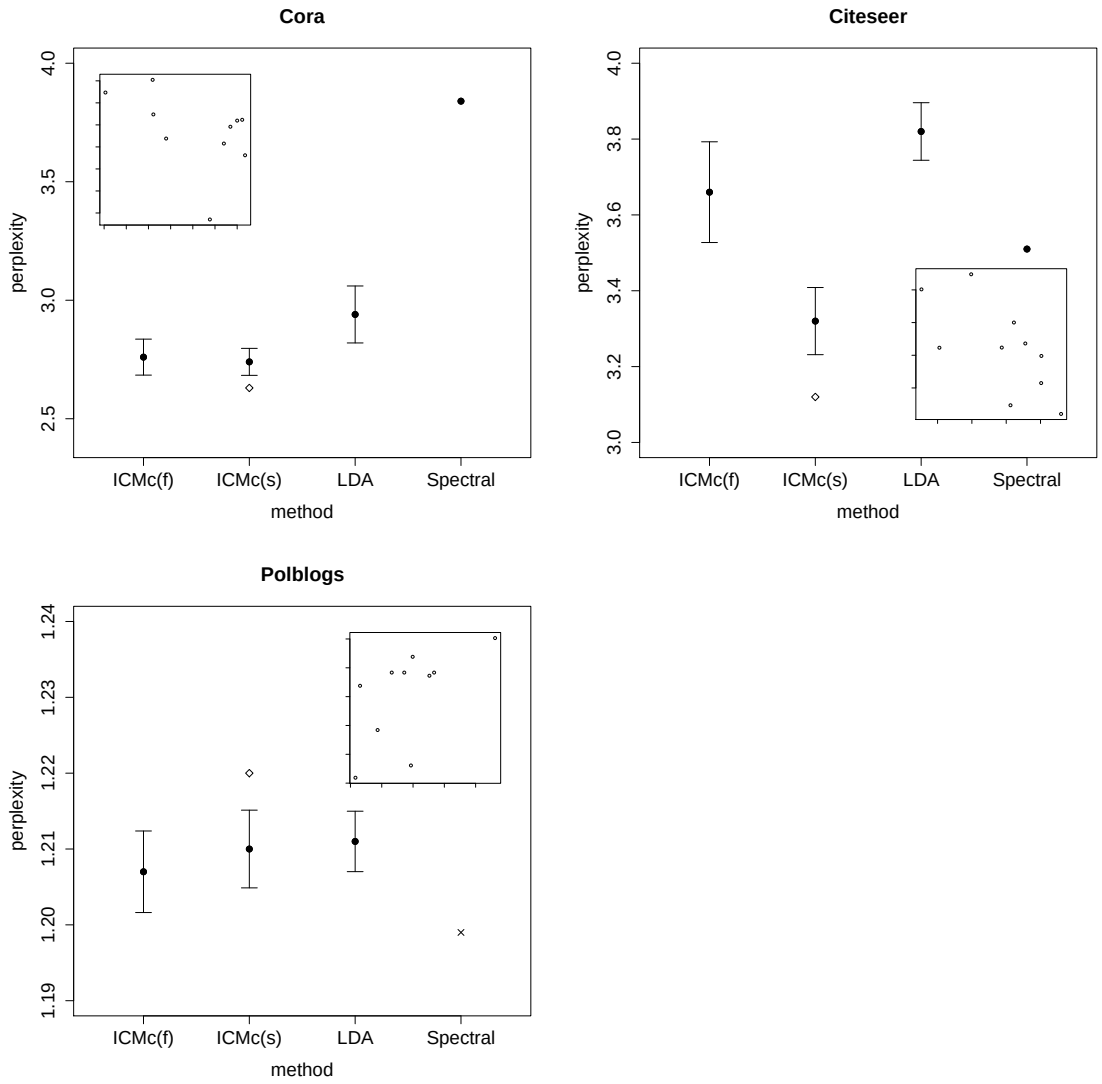

Figure 13: Comparison on medium-sized networks. Performance in finding true clusters, as measured by the perplexity of predicting ground-truth classes with the clusters. Datasets: Cora (top, left), Citeseer (top, right) and Polblogs (bottom, left). Methods: ICMc with fixed (f) and sampled (s) hyperparameters, SSN-LDA and Spectral clustering. The 2SE error bars are over 10 runs and the white diamond corresponds to the best run of ICMc(s) chosen by the convergence estimator (see see Section 5.5.2). Insets show the proposed convergence estimator plotted against the perplexity for each separate run of ICMc(s).

for clustering, and that the proposed sampling scheme is suitable for this task. With sampled hyperparameters ICMc was slightly bettern than SSN-LDA, but the difference was not significant in all the datasets. On the other hand, sampling the hyperparameter for SSN-LDA as well could probably improve its performance as well.

A common problem for both ICMc and SSN-LDA is the relatively large variation

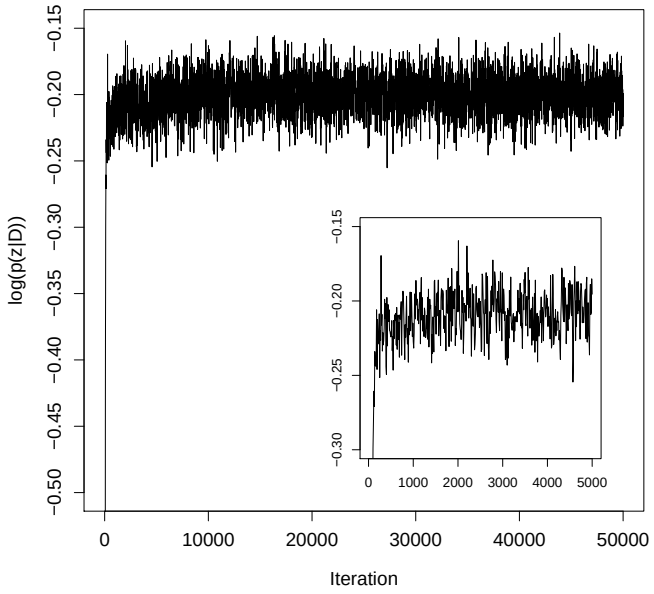

Figure 14: Convergence of the Gibbs sampler with sampled hyperparameters on the Citeseer citation set (first 5000 iterations enlarged as an inset). In the sampling, 50,000 iterations over the data were ran, but about 15,000 would have been enough for convergence, and about 3,000 for getting useful results (see the inset).

between different runs. This indicates that the sampler is not properly converged to the posterior, and is instead stuck in some local optimum. This is a general problem for many inference methods, especially with sparse data.

In addition to the hyperparameter estimation scheme a new convergence estimator was proposed, based on the probabilities of the component assignments of the left-out links. This estimator is easy to compute and based on the experiments it can be used to monitor the convergence of the sampling chain. Moreover, the experiments show that the estimator can be used to choose a good chain. This helps partially for the general convergence problem described above.

The model framework is also easily extendable and it was applied on a multi-relational problem setting. It was compared to a probabilistic relational model in a recommendation task. Although the ICM model was not originally designed for predictive purposes, it reached a performace level comparable to a more structure relational model. On the other hand, adding node-wise data into the model did not improve the performance of any of the methods significantly.

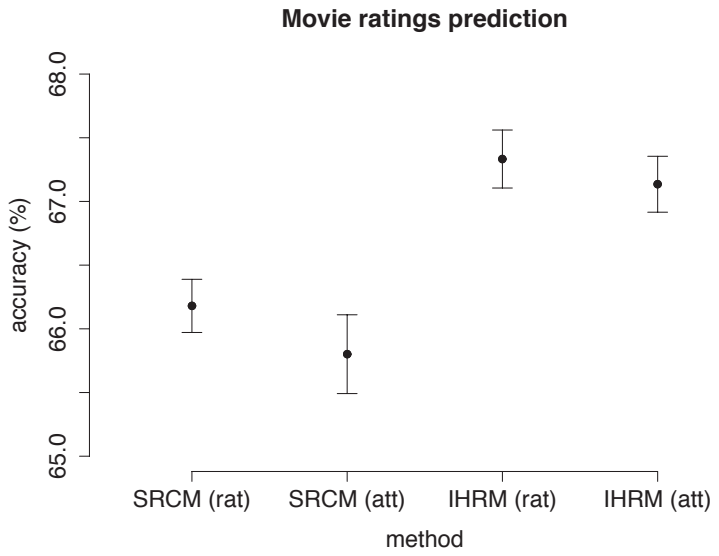

Figure 15: Rating prediction accuracies for our simple relational component model (SRCM) and IHRM of Xu et al. [65], either with data containing ratings only (rat) or with additional movie and user attributes (att). The 2SE error bars are over ten runs.

8 Discussion

In this thesis, extensions, improvements and applications of a generative probabilistic modeling framework for network data have been presented. The main goals were to study the suitability of the model on clustering-type tasks within different application domains, and additionally to study the effect of several modifications on model performance. Taken together, the obtained results indicate that the ICM framework is suitable for detecting community-like cluster structures from noisy network data within various problem domains. The model inference can additionally be improved significantly with relatively simple methods.

However, several questions remain and new ones have arisen during the work. Regarding the study of network analysis in general, a lot of work has been devoted to finding common properties from real-world networks arising from different fields, and models have been introduced that achieve good results in a variety of domains. This has been seen as an indication of some fundamental commonalities between these networks, especially in the complex networks research community. Not all scientists agree with the claim that networks from different fields share enough common properties to be of practical significance.

One important question here is the role of the chosen data representation. When representing a complex network system as a simple graph, researchers have to make strong simplifying assumptions. It is thus possible that as data are abstraced to the same simple form, they may seem to be structurally very similar. This does not however necessarily indicate that the important parts of the original systems are similar.

Another questionable aspect of networks analysis is the seemingly dominant role of clustering as a general task. Especially clustering of networks into densely connected subgraphs is a very popular task in at least biological and social domains. Reasons for this probably include that many real-world networks have been shown to exhibit such structures, and that these tend to have natural interpretations in many applications. This is a tempting feature especially for computational scientists, who may lack the deeper knowledge of the system the data is obtained from.

Many clustering methods for graphs have been proposed with various backgrounds, ranging from purely mathematical graph theory to machine learning. Reviews have been made in order to evaluate clustering algorithms on some specific problem domain. The evaluation of clusterings is however far from straightforward. Due to the inherent dependencies between data points, many standard evaluation methods cannot be applied as such to networks.

A typical approach is to compare the obtained clustering to some known classes of the nodes with statistical means, as was done in this thesis. Such results depend always on the goodness measure and theground truth classes, which are imperfect representations of the truth and can easily be misinterpreted. Studying the evaluation methods for clustering has actually become an interesting goal of research itself, and some general progress may be expected in the future.

Another possibility is to generate artificial datasets, where the truth is known well. A problem here is how to generate data that exhibits the nature of real-world networks as well as possible. This is in itself a very challenging task and in fact one popular goal in the analysis of real-world networks.

As the real-world networks have been clustered for decades now and most achievements are nothing but small improvements evaluated with questionable methods, other approaches should probably be sought instead. Recently, the work by Jure Leskovec [36] has shaken the traditional views of large social networks, especially their structure and evolution. While communities have been assumed as the basic building blocks of even very large networks, Leskovec has shown that this does not hold when the networks and communities grow large enough. This shows that new and critical approaches are needed to advance science around real network systems.

The modeling framework used in the thesis was motivated by the need for a simple generative model for community structures. Some aspects of the model and current applications should, however, be discussed.

Collapsed Gibbs sampling was used for inference, although the recent development of variational methods raises the question of whether the current method is optimal for the task. A very recent study by Asuncion et al. [6] showed that for topic models the choice of proper hyperparameter values is essential to model performance, indicating that the hyperparameter estimation procedure presented in the thesis was a critical improvement and could be studied further in the future. The same study also showed that the best sampling and variational methods show only insignificant performance differences. Thus, given the quickness and robustness of variational methods, they should be considered as an alternative for the collapsed Gibbs sampler. Their applicability for sparse data should on the other hand be studied carefully

Despite the improved inference, the difference to SSN-LDA in the experiments was relatively small. This indicates that although ICM was designed especially for community-like structures, whereas SSN-LDA can find a wider range of structures, the benefit from the specialization is not enough to distinguish the methods in this task. On the other hand, ICM may be more easily extendable, for example the Dirichlet Process prior can be included straightforwardly, while for SSN-LDA it would require a more complex and computationally heavier hierarchical prior.

Based on this study, it is hard to give general advice about choosing between the compared models. However, it can be concluded that if the data is known to be noisy, probabilistic approaches should be preferred. Spectral clustering seems to be unreliable in some cases, as it failed totally with the Polblogs data before manual edits of the data. ICM could be preferred over SSN-LDA, if it is clear that the data has a community structure and the goal is to detect it. Using the estimated hyperparameters and the proposed evaluation of different chains one can easily obtain relatively good results. In more general cases where little is known of the data, SSN-LDA may be a better choice, at least if the hyperparameters are estimated. The proposed sampling scheme should be straightforwardly applicable to SSN-LDA

as well.

An interesting aspect of the ICM model is that it operates on the level of links, whereas most of the other related algorithms operate on nodes. Unfortunately, because the real-world applications studied so far tend to deal with nodes, this aspect cannot be exploited, and can instead be a disadvantage in such applications.

A recent development in network analysis is the incorporation of additional data into the simple graph analysis. This also brings networks very close to the relational model. It is intuitively clear that if we have for example additional information about the actors in a social network, it should be included in the study and could lead to better results.

The multi-relational experiments presented in this thesis on biological and social domains were not, however, that promising, as the performance did not change significantly compared to using the graph data only. The result reflects other studies on PPI and other biological data [41, 52]. There are many possible reasons for this. In the biological case this may truly be due to protein interaction being a stronger indication of functional similarity than the overall similarity of gene expression profiles. It is also possible that the chosen evaluation criteria are biased towards bare network data.

On the basis of these rather simple experiments, it is too early to conclude that the additional data is useless. A more probable explanation is that the models have to be improved on how they treat different types of data. In the multi-relational ICM framework, for example, the interaction data has much more weight than node-wise data, resulting in an unfavorable bias.

Related to this, an important aspect mostly neglected in current approaches is the context-dependency of measurement data. Both protein interaction and gene expression vary a lot due to many factors, such as location and environmental conditions. Also the measuring techniques have notable influence on the resulting data. Researchers should be aware of such factors concerning the studied data, and moreover this knowledge should be incorporated into the models.

A Technical details

The appendix contains technical details, mainly equations, for the different Interaction Component Models.

A.1 Equations for ICMc

This section contains the detailed equations for the step-by-step derivation of the collapsed Gibbs sampler for the basic ICMc.

A.1.1 Likelihood, joint probability and marginalization

The likelihood of ICMc is

$$p(L, Z|\phi, \theta) = \prod_{l}^{N} \theta_{z(l)} \phi_{z(l)i(l)} \phi_{z(l)j(l)} = \prod_{z}^{C} \theta_z^{n_z} \prod_{iz}^{MC} \phi_{zi}^{q_{zi}}, \tag{23}$$

where in the latter expression we have counts n_z of links assigned to each component, and counts q_{zi} of component-node co-occurrences. Adding the Dirichlet priors to the likelihood we get the joint probability

$$p(L, Z, \phi, \theta|\alpha, \beta) = D_1(\alpha, \beta) \prod_{z}^{C} \theta_z^{n_z+\alpha-1} \prod_{iz}^{MC} \phi_{zi}^{q_{zi}+\beta-1}, \tag{24}$$

where $D_1(\alpha, \beta)$ is a normalizing constant from the priors:

$$D_1(\alpha, \beta) = \frac{\Gamma(C\alpha)}{\Gamma(\alpha)^C} \cdot \left[\frac{\Gamma(M\beta)}{\Gamma(\beta)^M}\right]^C. \tag{25}$$

A.1.2 Inference with collapsed Gibbs sampling

For the collapsed Gibbs sampler the multinomial distribution parameters θ and ϕ_z are integrated out from (24). This results in the marginalized joint probability

$$p(L, Z|\alpha, \beta) = D_1(\alpha, \beta) \frac{\prod_z \Gamma(n_z + \alpha)}{\Gamma(N + C\alpha)} \cdot \prod_z \frac{\prod_i \Gamma(q_{zi} + \beta)}{\Gamma(2n_z + M\beta)}. \tag{26}$$

Because links are generated independently, they can be separated from $p(L, Z|\alpha, \beta)$ into link-wise factors. Separate one arbitrary link, say l_0, associated to the latent variable z_0 and to nodes i_0 and j_0, from the product, and denote by (L', Z') the other links and their associated latent components, and by (q', n', N') the counts as they were if the link was nonexistent. For most indices, we will have $q' = q$ and

$n' = n$, and always $N' = N - 1$, but for some indices $q' = q - 1$ and $n' = n - 1$. Due to the gamma function property

$$\begin{aligned} \Gamma(x) &= (x-1)\Gamma(x-1) \\ \Gamma(x) &= (x-1)(x-2)\Gamma(x-2)\ , \end{aligned}$$

we can write the recurrence formula

$$\begin{aligned} p(L', Z', l_0, z_0|\alpha, \beta) = p(L', Z'|\alpha, \beta) \cdot u_z = \\ D_1 \frac{\prod_z \Gamma(n'_z + \alpha)}{\Gamma(N' + C\alpha)} \cdot \prod_z \frac{\prod_i \Gamma(q'_{zi} + \beta)}{\Gamma(2n'_z + M\beta)} \cdot u_z\ , \end{aligned} \tag{27}$$

where

$$u_z \equiv p(l_0, z_0|L', Z', \alpha, \beta) = \frac{n'_{z_0} + \alpha}{N' + C\alpha} \cdot \frac{(q'_{z_0 i_0} + \beta)(q'_{z_0 j_0} + \beta)}{(2n'_{z_0} + 1 + M\beta)(2n'_{z_0} + M\beta)}, \tag{28}$$

which is the same as (11) in the main text.

A.2 Equations for ICMg2

Next, detailed equations for the step-by-step derivation of the collapsed Gibbs sampler are presented for the model variant ICMg2.

A.2.1 Joint probability and marginalization

Joint probability of ICMg2 is a product of the link-specific probabilities that are the same as with the basic ICMc, and the normally distributed expression profile probabilities for the nodes, with priors for the component-specific means:

$$\begin{aligned} p(L, X, Z, \phi, \mu, \theta) = D_1(\alpha, \beta) E_1(\bar{\mu_0}, V_0, V) \prod_z^C \theta_z^{n_z + m_z + \alpha - 1} \prod_{iz}^{MC} \phi_{zi}^{q_{zi} + \beta - 1} \\ \cdot \prod_z^C \left[N(\bar{\mu}_z|\bar{\mu}_0, V_0) \prod_{z(k)=z}^{m_z} N(\bar{x}_k|\bar{\mu}_z, V) \right], \end{aligned} \tag{29}$$

where D_1 and E_1 are normalizing constants. The node data generation part can be written

$$
\begin{aligned}
p(X,\mu) &= \prod_z^C \Bigg[N(\bar{\mu}_z|\bar{\mu}_0, V_0) \prod_{z(k)=z}^{m_z} N(\bar{x}_k|\bar{\mu}_z, V) \Bigg] \\
&= \prod_z^C (2\pi)^{d/2(m_z+1)} |V_0|^{-1/2} |V|^{-m_z/2} \\
&\cdot \exp\Bigg[-\frac{1}{2} \Bigg((\bar{\mu}_z - \bar{\mu}_0)^T V_0^{-1} (\bar{\mu}_z - \bar{\mu}_0) + \sum_k^{m_z} (\bar{x}_k - \bar{\mu}_z)^T V^{-1} (\bar{x}_k - \bar{\mu}_z) \Bigg) \Bigg] \\
&= \underbrace{(2\pi)^{d/2(M+C)} |V_0|^{-C/2} |V|^{-M/2}}_{=E_2 = constant} \\
&\cdot \prod_z^C \exp\Bigg[-\frac{1}{2} \Bigg(\bar{\mu}_z^T V_0^{-1} \bar{\mu}_z - 2\bar{\mu}_z^T V_0^{-1} \bar{\mu}_0 + \bar{\mu}_0^T V_0^{-1} \bar{\mu}_0 + \cdots \\
&+ \sum_k^{m_z} (\bar{\mu}_z^T V^{-1} \bar{\mu}_z - 2\bar{\mu}_z^T V^{-1} \bar{x}_k + \bar{x}_k^T V^{-1} \bar{x}_k) \Bigg) \Bigg] \\
&= E_2 \prod_z^C \exp\Bigg[-\frac{1}{2} \Bigg(\bar{\mu}_z^T \underbrace{(V_0^{-1} + \sum_k^{m_z} V^{-1})}_{=S^{-1}} \bar{\mu}_z - 2\bar{\mu}_z^T S^{-1} \underbrace{S(V_0^{-1}\bar{\mu}_0 + V^{-1} \sum_k^{m_z} \bar{x}_k)}_{=A} + \cdots \\
&+ A^T S^{-1} A - A^T S^{-1} A + \bar{\mu}_0^T V_0^{-1} \bar{\mu}_0 + \sum_k^{m_z} (\bar{x}_k^T V^{-1} \bar{x}_k) \Bigg) \Bigg] \\
&= E_2 \prod_z^C \exp\Bigg[-\frac{1}{2} \Bigg(\bar{\mu}_z^T S^{-1} \bar{\mu}_z - 2\bar{\mu}_z^T S^{-1} A + A^T S^{-1} A \cdots \\
&- A^T S^{-1} A + \bar{\mu}_0^T V_0^{-1} \bar{\mu}_0 + \sum_k^{m_z} (\bar{x}_k^T V^{-1} \bar{x}_k) \Bigg) \Bigg].
\end{aligned}
\tag{30}
$$

By adding the normalizing constant of the Gaussian we get

$$
\begin{aligned}
p(X|\bar{\mu}) &= E_2 \prod_z^C (2\pi)^{d/2} |S|^{1/2} N(\bar{\mu}_z|A, S) \\
&\cdot \exp\Bigg[-\frac{1}{2} \Bigg(-A^T S^{-1} A + \bar{\mu}_0^T V_0^{-1} \bar{\mu}_0 + \sum_k^{m_z} (\bar{x}_k^T V^{-1} \bar{x}_k) \Bigg) \Bigg] \\
&= E_3 \exp\Bigg[-\frac{1}{2} \sum_k^M (\bar{x}_k^T V^{-1} \bar{x}_k) \Bigg] \prod_z^C N(\bar{\mu}_z|A, S) f(m_z, \bar{x}_z),
\end{aligned}
\tag{31}
$$

where where posterior covariance matrix S, posterior mean A, an auxiliary function f and a normalizing constant E_3 are as follows:

$$S = (V_0^{-1} + m_z V^{-1})^{-1} \tag{32}$$

$$A = S \cdot (V_0^{-1}\bar{\mu}_0 + V^{-1}\sum_k^{m_z} \bar{x}_k) \tag{33}$$

$$f(m_z, \bar{x}_z) = |S|^{1/2} \exp\left[\frac{1}{2} A^T S^{-1} A\right] \tag{34}$$

$$E_3 = (2\pi)^{-(Md)/2} |V_0|^{-C/2} |V|^{-M/2} \exp\left[-\frac{C}{2}\bar{\mu}_0^T V_0^{-1}\bar{\mu}_0\right]. \tag{35}$$

The whole joint probability of equation (29) can now be written

$$\begin{aligned} p(L, X, Z, \phi, \mu, \theta) = D_1 E_3 \prod_z^C \theta_z^{n_z + m_z + \alpha - 1} \prod_{iz}^{MC} \phi_{zi}^{q_{zi} + \beta - 1} \\ \times \prod_z^C \left[N(\bar{\mu}_z | A, S) f(m_z, \bar{x}_z) \right] \cdot \exp\left[-\frac{1}{2} \sum_k^M (\bar{x}_k^T V^{-1} \bar{x}_k) \right]. \end{aligned} \tag{36}$$

A.2.2 Inference with collapsed Gibbs sampling

A collapsed Gibbs sampler is derived analogously to the basic ICMc, the marginalization now including the component specific means μ_z, which are integrated out from (29). The marginalized probability is then separated into link-wise and node-wise factors, using auxiliary results derived in the next section.

Sampling equation for links is analogous to that of ICM:

$$\begin{aligned} p(L', Z', l_0, z_0) = p(L', Z') \cdot u_0 = D_2 \frac{\prod_z \Gamma(n'_z + m_z + \alpha)}{\Gamma(N' + M + C\alpha)} \\ \times \prod_z \frac{\prod_i \Gamma(q'_{zi} + \beta)}{\Gamma(2n'_z + m_z + M\beta)} \times \prod_w^C f(m_w, \bar{x}_w) \times \exp\left[-\frac{1}{2} \sum_k^M (\bar{x}_k^T V^{-1} \bar{x}_k) \right] \cdot u_0 \ , \end{aligned} \tag{37}$$

where

$$u_0 \equiv p(l_0, z_0 | L', Z') = (n'_{z_0} + m_{z_0} + \alpha) \cdot \frac{(q'_{z_0 i_0} + \beta)(q'_{z_0 j_0} + \beta)}{(2n'_{z_0} + m_{z_0} + 1 + M\beta)(2n'_{z_0} + m_{z_0} + M\beta)} \ . \tag{38}$$

For nodes we get

$$\begin{aligned} p(\bar{x}', Z', x_0, z_0) = p(\bar{x}', Z') \cdot u_0 = D_2 \frac{\prod_z \Gamma(n_z + m'_z + \alpha)}{\Gamma(N + M' + C\alpha)} \\ \times \prod_z \frac{\prod_k \Gamma(q'_{zk} + \beta)}{\Gamma(2n_z + m'_z + M'\beta)} \times \prod_w^C f(m'_w, \bar{x}'_w) \times \exp\left[-\frac{1}{2} \sum_k^{M'} (\bar{x}_k^T V^{-1} \bar{x}_k) \right] \cdot u_0 \ , \end{aligned} \tag{39}$$

where

$$u_0 \equiv p(\bar{x}_0, z_0|X', Z') = \frac{(n_{z_0} + m'_{z_0} + \alpha)(q'_{z_0 k_0} + \beta)}{2n_{z_0} + m'_{z_0} + M'\beta} \cdot \left[\frac{|S|}{|S'|}\right]^{1/2} \cdot \exp\left[-\frac{1}{2}\bar{x}_0^T V^{-1}\bar{x}_0 + \frac{1}{2}A^T S^{-1}A - A'^T S'^{-1}A'\right]. \quad (40)$$

A.2.3 Auxiliary results

For Gibbs sampling we want to separate the effect of one node from the joint probability. This is shown in counts m' and M' and in the data sum , which are denoted with , e.g., m'_{z_0}. In (36) this changes S, A and the data sum. We want therefore factorize (36) as.

$$h(m'_{z_0}, \bar{x}_0)E_3 \exp\left[-\frac{1}{2}\sum^{M'}(\bar{x}_k^T V^{-1}\bar{x}_k)\right]\prod_z^C |S'|^{1/2}\exp\left[\frac{1}{2}A'^T S'^{-1}A'\right]. \quad (41)$$

Here an arbitrary node indexed by 0 is separated from the others, with data $\bar{x}_0$, component z_0 and $m'_{z_0} = m_{z_0} - 1$. The factor h can be divided into factors h_n, when $h = \prod_n h_n$. Symbols: $m = m_{z_0} = m'_{z_0} + 1$, $m' = m'_{z_0}$ and $\sum' x_k = \sum^{m'} x_k = \sum^{m} x_k - x_0$). The factors of h can now be solved:

1. We want h_1 to fulfill

$$h_1(m') \cdot |S'|^{1/2} = |S|^{1/2} \Rightarrow h_1 = \left[\frac{|S|}{|S'|}\right]^{1/2}. \quad (42)$$

2. We want h_2 to fulfill

$$h_2(x_0) \cdot \exp\left[-\frac{1}{2}\sum^{M'}(\bar{x}_k^T V^{-1}\bar{x}_k)\right] = \exp\left[-\frac{1}{2}\sum^{M}(\bar{x}_k^T V^{-1}\bar{x}_k)\right]$$
$$\Rightarrow h_2(x_0) = \exp\left[-\frac{1}{2}\bar{x}_0^T V^{-1}\bar{x}_0\right]. \quad (43)$$

3. We want h_3 to fulfill

$$h_3(m', x_0) \cdot \exp\left[\frac{1}{2}A'^T S'^{-1}A'\right] = \exp\left[\frac{1}{2}A^T S^{-1}A\right]$$
$$\Rightarrow h_3 = \exp\left[\frac{1}{2}\left(A^T S^{-1}A - A'^T S'^{-1}A'\right)\right]. \quad (44)$$

For h we then get

$$h = \left[\frac{|S|}{|S'|}\right]^{1/2} \cdot \exp\left[-\frac{1}{2}\bar{x}_0^T V^{-1}\bar{x}_0 + \frac{1}{2}A^T S^{-1}A - A'^T S'^{-1}A'\right]. \quad (45)$$

A.3 Hyperparameter sampling

This appendix contains the equations needed for the hyperparameter sampling procedure used in the thesis.

A.3.1 Hyperpriors

Possible hyperpriors are normal and inverse gamma distributions. Currently a gamma prior with shape $a = 1.5$ and inverse scale $b = 1.0$ are used.
The density of the gamma distribution for parameter θ with hyperparameters a and b is

$$p(\theta|a,b) = \frac{b^a}{\Gamma(a)}\theta^{a-1}e^{-b\theta} \; . \tag{46}$$

Taking the logarithm produces

$$\log p(\theta|a,b) = a\log b - \log\Gamma(a) + (a-1)\log\theta - b\theta \; . \tag{47}$$

Differentiating with respect to θ gives

$$\frac{\partial \log p(\theta|a,b)}{\partial\theta} = \frac{a-1}{\theta} - b \; , \tag{48}$$

and another differentiation gives

$$\frac{\partial^2 \log p(\theta|a,b)}{\partial\theta^2} = \frac{1-a}{\theta^2} \; . \tag{49}$$

Equivalently, the inverse gamma distribution for parameter θ with hyperparameters a and b is

$$p(\theta|a,b) = \frac{b^a}{\Gamma(a)}\theta^{-a-1}e^{-b/\theta} \; . \tag{50}$$

The derivatives are now

$$\frac{\partial \log p(\theta|a,b)}{\partial\theta} = \frac{-(a+1)}{\theta} + \frac{b}{\theta^2} \tag{51}$$

and

$$\frac{\partial^2 \log p(\theta|a,b)}{\partial\theta^2} = \frac{a+1}{\theta^2} - \frac{2b}{\theta^3} \; . \tag{52}$$

A.3.2 Posteriors

Recall the notation: $C = \#components$, $N = \#links$, $M = \#nodes$.

Alpha's posterior includes the prior $p(\alpha)$, which can be either a gamma or an inverse gamma distribution (see previous section). The posterior is

$$p(L,\alpha) \propto p(\alpha)p(L|\alpha) = p(\alpha)\cdot\frac{\Gamma(C\alpha)}{\Gamma(\alpha)^C}\frac{\prod_z\Gamma(n_z+\alpha)}{\Gamma(N+C\alpha)} \; , \tag{53}$$

or in the log domain

$$\log p(L,\alpha) = \log p(\alpha) + \log\Gamma(C\alpha) - C\log\Gamma(\alpha) + \sum_z \log\Gamma(n_z+\alpha) - \log\Gamma(N+C\alpha)\ . \tag{54}$$

Differentiating with respect to α gives

$$\frac{\partial \log p(L,\alpha)}{\partial\alpha} = \frac{\partial \log p(\alpha)}{\partial\alpha} + C\Psi(C\alpha) - C\Psi(\alpha) + \sum_z \Psi(n_z+\alpha) - C\Psi(N+C\alpha)\ , \tag{55}$$

where the Digamma function $\Psi()$ is the derivative of the logarithmic Gamma function. Differentiating again produces

$$\frac{\partial^2 \log p(L,\alpha)}{\partial\alpha^2} = \frac{\partial^2 \log p(\alpha)}{\partial\alpha^2} + C^2\Psi'(C\alpha) - C\Psi'(\alpha) + \sum_z \Psi'(n_z+\alpha) - C^2\Psi'(N+C\alpha)\ , \tag{56}$$

where the Trigamma function $\Psi'()$ in turn is the derivative of the Digamma function. Posterior of β is equivalently proportional to

$$p(L,\beta) \propto p(\beta)p(L|\beta) = p(\beta)\cdot\prod_z \frac{\Gamma(M\beta)}{\Gamma(\beta)^M}\frac{\prod_i \Gamma(q_{zi}+\beta)}{\Gamma(2n_z+M\beta)}\ , \tag{57}$$

or in the log domain

$$\log p(L,\beta) = \log p(\beta) + \sum_z \left[\log\Gamma(M\beta) - M\log\Gamma(\beta) + \sum_i \log\Gamma(q_{zi}+\beta) - \log\Gamma(2n_z+M\beta)\right] ..\tag{58}$$

Differentiating the logarithmic form with respect to β gives

$$\frac{\partial \log p(L,\beta)}{\partial\beta} = \frac{\partial \log p(\beta)}{\partial\beta} + \sum_z \left[M\Psi(M\beta) - M\Psi(\beta) + \sum_i \Psi(q_{zi}+\beta) - M\Psi(2n_z+M\beta)\right]\ . \tag{59}$$

Differentiating again gives

$$\frac{\partial^2 \log p(L,\beta)}{\partial\beta^2} = \frac{\partial^2 \log p(\beta)}{\partial\beta^2} + \sum_z \left[M^2\Psi'(M\beta) - M\Psi'(\beta) + \sum_i \Psi'(q_{zi}+\beta) - M^2\Psi'(2n_z+M\beta)\right]\ . \tag{60}$$

References

[1] Adamic, L. A., and Glance, N. The political blogosphere and the 2004 U.S. election: Divided they blog, 2005.

[2] Airodi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research 9*, Sep (2008), 1981–2014.

[3] Airoldi, E. M. Getting started in probabilistic graphical models. *PLoS Compututational Biology 3*, 12 (12 2007), e252.

[4] Albert, R., and Barabàsi, A.-L. Statistical mechanics of complex networks. *Reviews of Modern Physics 74*, 1 (2002), 47–97.

[5] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., and Butler, H. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nature Genetics 25* (2000), 25–29.

[6] Asuncion, A., Welling, M., Smyth, P., and Teh, Y. W. On smoothing and inference for topic models. In *The 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009)* (2009).

[7] Aukia, J. Bayesian clustering of huge friendship networks. Master's thesis, Laboratory of Computer and Information Science, Helsinki University of Technology, Espoo, Finland, 2007.

[8] Barabàsi, A.-L., and Albert, R. Emergence of scaling in random networks. *Science 286*, 5439 (1999), 509–512.

[9] Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. New York: Springer, 2006.

[10] Blei, D., Ng, A., and Jordan, M. Latent Dirichlet allocation. *Journal of Machine Learning Research 3* (2003), 993–1022.

[11] Brohee, S., and van Helden, J. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics 7*, 1 (2006), 488.

[12] Buntine, W., and Jakulin, A. Applying discrete pca in data analysis. In *AUAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence* (Arlington, Virginia, United States, 2004), AUAI Press, pp. 59–66.

[13] Cai, D., Shao, Z., He, X., Yan, X., and Han, J. Community mining from multi-relational networks. In *Proceedings of the 2005 European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'05)* (Porto, Portugal, 2005).

[14] Chapelle, O., Schölkopf, B., and Zien, A., Eds. *Semi-Supervised Learning*. MIT Press, Cambridge, USA, 2006.

[15] Codd, E. F. A relational model of data for large shared data banks. *Communications of the ACM 13*, 6 (1970), 377–387.

[16] Collins, S. R., Kemmeren, P., Zhao, X.-C., Greenblatt, J. F., Spencer, F., Holstege, F. C. P., Weissman, J. S., and Krogan, N. J. Toward a Comprehensive Atlas of the Physical Interactome of Saccharomyces cerevisiae. *Molecular and Cellular Proteomics 6*, 3 (2007), 439–450.

[17] Dijkstra, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik 1* (1959), 269–271.

[18] Dongen, S. A cluster algorithm for graphs. Tech. rep., Amsterdam, The Netherlands, The Netherlands, 2000.

[19] Džeroski, S., Ed. *Relational Data Mining*. Springer-Verlag New York, Inc., New York, NY, USA, 2000.

[20] Erdős, P., and Rényi, A. On random graphs i. *Publicationes Mathematicae (Debrecen) 6* (1959), 290–297.

[21] Euler, L. Solutio problematis ad geometriam situs pertinentis. *Commenrarii Academiae Scientiarum Imperialis Petropolitanae 8* (1736), 128–140. Reprint in English in N. Biggs, E. Lloyd, and R. Wilson (1976), editors, *Graph Theory 1736-1936*, Clarendon Press, Oxford, UK.

[22] Fortunato, S., and Castellano, C. Community structure in graphs. *ArXiv e-prints* (2007). arXiv:0712.2716.

[23] Fraser, H. B., Hirsh, A. E., Steinmetz, L. M., Scharfe, C., and Feldman, M. W. Evolutionary Rate in the Protein Interaction Network. *Science 296*, 5568 (2002), 750–752.

[24] Friedman, N., Getoor, L., Koller, D., and Pfeffer, A. Learning probabilistic relational models. In *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence* (San Francisco, CA, USA, 1999), Morgan Kaufmann Publishers Inc., pp. 1300–1309.

[25] Gasch, A., Huang, M., Metzner, S., Botstein, D., and Elledge, S. Genomic expression responses to dna-damaging agents and the regulatory role of the yeast atr homolog mec1p. *Molecular Biology of the Cell 12* (2001), 2987–3003.

[26] Gelman, A., Carlin, J., Stern, H., and Rubin, D. *Bayesian data analysis*, second ed. Chapman & Hall, Boca Raton, USA, 2004.

[27] Getoor, L., Koller, D., Taskar, B., and Friedman, N. Learning probabilistic relational models with structural uncertainty. In *In Proceedings of the ICML-2000 Workshop on Attribute-Value and Relational Learning:Crossing the Boundaries* (2000), pp. 13–20.

[28] Giorgini, F., and Muchowski, P. Connecting the dots in huntington's disease with protein interaction networks. *Genome Biology 6*, 3 (2005), 210.

[29] Griffiths, T. L., and Steyvers, M. Finding scientific topics. *Proceedings of the National Academy of Sciences USA 101 Suppl 1* (2004), 5228–5235.

[30] Güldener, U., Münsterkötter, M., Kastenmüller, G., Strack, N., van Helden, J., Lemer, C., Richelles, J., Wodak, S. J., Garcia-Martinez, J., Perez-Ortin, J. E., Michael, H., Kaps, A., Talla, E., Dujon, B., Andre, B., Souciet, J. L., Montigny, J. D., Bon, E., Gaillardin, C., and Mewes, H. W. Cygd: the comprehensive yeast genome database. *Nucleic Acids Research* (2005), 1;33 Database issue:D364–8.

[31] Hofmann, T. Probabilistic latent semantic analysis. In *In Proc. of Uncertainty in Artificial Intelligence, UAI 1999* (1999), pp. 289–296.

[32] Huang, Y. J., Hang, D., Lu, L. J., Tong, L., Gerstein, M. B., and Montelione, G. T. Targeting the Human Cancer Pathway Protein Interaction Network by Structural Genomics. *Molecular and Cellular Proteomics 7*, 10 (2008), 2048–2060.

[33] Jiang, J. Q., Dress, A. W., and Yang, G. A spectral clustering-based framework for detecting community structures in complex networks. *Applied Mathematics Letters 22*, 9 (2009), 1479 – 1482.

[34] Keller, E. F. Revisiting "scale-free" networks. *BioEssays 27*, 10 (2005), 1060–1068.

[35] Kemp, C., Tenenbaum, J. B., Griffiths, T. L., Yamada, T., and Ueda, N. Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)* (Menlo Park, USA, 2006), Y. Gil and R. Mooney, Eds., AAAI Press.

[36] Leskovec, J., Chakrabarti, D., Kleinberg, J., Faloutsos, C., and Gharamani, Z. Kronecker graphs: an approach to modeling networks. *ArXiv e-prints* (2008). arXiv:0812.4905v2.

[37] Liu, Y., Niculescu-Mizil, A., and Gryc, W. Topic-link lda: joint models of topic and author community. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning* (New York, NY, USA, 2009), ACM, pp. 665–672.

[38] Luo, M., Ma, Y.-F., and Zhang, H.-J. A spatial constrained k-means approach to image segmentation. In *Proceedings of the 2003 Joint Conference of the Fourth International Conference onInformation, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia.* (2003), vol. 2, pp. 738–742.

[39] Meilă, M. Comparing clusterings—an information based distance. *Journal of Multivariate Analysis 98*, 5 (2007), 873–895.

[40] Minka, T. Estimating a dirichlet distribution, 2000. URL [http://research.microsoft.com/en-us/um/people/minka/papers/dirichlet/].

[41] Nariai, N., Kolaczyk, E. D., and Kasif, S. Probabilistic protein function prediction from heterogenous genome-wide data. *PLoS ONE 2(3)* (2007), e337.

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

[43] Newman, M. E. J. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences USA 103* (2006), 8577–8582.

[44] O'Rourke, S., and Herskowitz, I. Unique and redundant roles for hog mapk pathway components as revealed by whole-genome expression analysis. *Molecular Biology of the Cell 15* (2004), 532–42.

[45] Parkkinen, J., Gyenge, A., Sinkkonen, J., and Kaski, S. A block model suitable for sparse graphs. In *The 7th International Workshop on Mining and Learning with Graphs (MLG'09), Leuven, Belgium, July 2-4* (2009).

[46] Przulj, N., Corneil, D. G., and Jurisica, I. Modeling interactome: scale-free or geometric? *Bioinformatics 20*, 18 (2004), 3508–3515.

[47] Riesen, K., and Bunke, H. Kernel k-means clustering applied to vector space embeddings of graphs. In *Art. Neural Networks in Pattern Recognition, Proc. 3rd IAPR Workshop* (2008), L. Prevost, S. Marinai, and F. Schwenker, Eds., LNCS 5064, Springer, pp. 24–35.

[48] Rivals, I., Personnaz, L., Taing, L., and Potier, M.-C. C. Enrichment or depletion of a go category within a class of genes: which test? *Bioinformatics 23*, 4 (2007), 401–407.

[49] Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J. Analysis of recommendation algorithms for e-commerce. In *ACM Conference on Electronic Commerce* (2000), pp. 158–167.

[50] Segal, E., Wang, H., and Koller, D. Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics 19*, Suppl 1 (2003), i264–i272.

[51] Sen, P., and Getoor, L. Link-based classification. Tech. Rep. CS-TR-4858, University of Maryland, College Park, USA, 2007.

[52] Shiga, M., Takigawa, I., and Mamitsuka, H. Annotating gene function by combining expression data with a modular gene network. *Bioinformatics 23* (2007), i468–i478.

[53] Sinkkonen, J., Aukia, J., and Kaski, S. Inferring vertex properties from topology in large networks. In *Working Notes of the 5th International Workshop on Mining and Learning with Graphs (MLG'07)* (Florence, Italy, 2007), Universita degli Studi di Firenze.

[54] Sinkkonen, J., Aukia, J., and Kaski, S. Infinite mixtures for multi-relational categorical data. In *The 6th International Workshop on Mining and Learning with Graphs (MLG'08), Helsinki, Finland, July 4-5* (2008).

[55] Tanay, A., Steinfeld, I., Kupiec, M., and Shamir, R. Integrative analysis of genome-wide experiments in the context of a large high-throughput data compendium. *Molecular Systems Biology 1*, 2 (2005).

[56] Tarassov, K., Messier, V., Landry, C. R., Radinovic, S., Molina, M. M. S., Shames, I., Malitskaya, Y., Vogel, J., Bussey, H., and Michnick, S. W. An in Vivo Map of the Yeast Protein Interactome. *Science 320*, 5882 (2008), 1465–1470.

[57] Tavazoie, S., Hughes, J. D., Campbell, M. J., Cho, R. J., and Church, G. M. Systematic determination of genetic network architecture. *Nature genetics 22*, 3 (1999), 281–285.

[58] Teh, Y. W. Dirichlet processes. Submitted to Encyclopedia of Machine Learning, 2007.

[59] Travers, J., and Milgram, S. An experimental study of the small world problem. *Sociometry 32*, 4 (1969), 425–443.

[60] Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. B. Missing value estimation methods for dna microarrays. *Bioinformatics 17* (2001), 520–525.

[61] Tukey, J. W. *Exploratory data analysis.* Addison-Wesley, Reading, USA, 1977.

[62] Ulitsky, I., and Shamir, R. Identification of functional modules using network topology and high-throughput data. *BMC Systems Biology 1* (2007), 8.

[63] Wasserman, S., and Faust, K. *Social Network Analysis: Methods and Applications.* Cambridge University Press, Cambridge, UK, 1994.

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

[65] Xu, Z., Tresp, V., Yu, K., and Kriegel, H.-P. Infinite hidden relational models. In *Proceedings of the 22nd International Conference on Uncertainty in Artificial Intelligence (UAI 2006)* (2006).

[66] Young, J. Relational databases – benefits and drawbacks. *Data Processing 28*, 6 (1986), 312 – 313.

[67] Zhang, H., Qiu, B., Giles, C. L., Foley, H. C., and Yen, J. An LDA-based community structure discovery approach for large-scale social networks. In *Intelligence and Security Informatics (ISI) 2007*. IEEE, 2007, pp. 200–207.

[68] Zhou, D., He, Y., and Kwoh, C. K. Extracting protein-protein interactions from medline using the hidden vector state model. *International Journal of Bioinformatics Results and Applications 4*, 1 (2008), 64–80.