Dynamic Relational Models of Complex Network

by

Zheng Yu

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Software Engineering and Intelligent Systems

Department of Electrical and Computer Engineering

University of Alberta

Abstract

Analysis of complex networks is one of the most important topics in the Machine Learning field. At the same time, classical probabilistic graphical relational models are one of the most popular methods used to perform such tasks. However, there are several limitations associated with a process of constructing probabilistic relational models. Some of them are: inability to cope with fully masked data; assumptions of data independence; insufficient interpretability and precision of models; and inadequate modelling of network's dynamics in continuous time. All this leads to construction of simplified models, as well as lack of full utilization of the available data.

In this thesis, we proposed a number of methods developed based on different types of Machine Learning techniques, such as Deep Learning and Bayesian nonparametric and stochastic processes, to address these limitations. More specifically, we propose some modifications of the mixed membership stochastic blockmodel, i.e., we focus on modeling: 1) coupling relations within/across groups/communities of nodes using the multilayer network with static settings; 2) coupling relations between communities using a matrix factorization method; and 3) coupling relations between nodes across groups/communities using a long short term memory.

In addition, we also improve the ability of relational models from the perspective of accuracy (model performance) and interpretability. In this case, we enable clustering of both nodes and edges simultaneously. We use discrete fragmentation coagulation process to cluster nodes of a network, and mixed membership stochastic blockmodel to cluster its edges. Furthermore, we focus on modelling changes in relational data occurring over continuous time. Specifically, in order to prevent an information loss we use the continuous fragmentation coagulation process to model the community evolution, as well as Hawkes process to model the reciprocating relation among nodes. We validate our model using synthetic and real datasets.

Preface

The research in this thesis was done by Zheng Yu under the supervision of Professor Marek Reformat. Zheng Yu was partially supported by the China Scholarship Council.

Chapter 3 includes materials published as Z. Yu, M. Pietrasik, M. Reformat, ”Deep Dynamic Mixed Membership Stochastic Blockmodel.” 2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI). IEEE, 2019. I was responsible for data collection, experiment design, model design and manuscript writing. The second author, M. Pietrasik, contributed to the code and writing of the manuscript. M. Reformat was the supervisory author and was involved with manuscript composition.

Chapter 4 includes materials published as Z. Yu, X. Fan, M. Pietrasik, M. Reformat, ”Fragmentation Coagulation Based Mixed Membership Stochastic Blockmodel.” AAAI. 2020. I was responsible for the data collection, data analysis, model design, experiment and manuscript writing. The second author, Xuhui Fan, contributed to the manuscript writing. Dr. Marek Reformat was the supervisory author and was involved with manuscript writing.

Chapter 5 includes materials to be submitted as Z. Yu, X. Fan, R. Zhang, M. Reformat, ” Reciprocating Interactions Simulation in Continuous Time.” 2020 (to be submitted). I was responsible for the data collection, data cleaning, model design, result analysis, and manuscripts writing. Xuhui Fan contributed to the manuscript writing and model design. Dr. Marek Reformat was the supervisory author and was involved with manuscript writing.

Acknowledgements

My sincere thanks to my supervisor, Professor Marek Reformat. Without his consistent encouragement, motivation, inspiration and guidance, I could not finish this research.

I would like to give my sincere thanks to all my labmates. I would also thank to Xuhui Fan. I get inspired by the useful discussion with him. His strong academic support helps me during my study.

My thanks then go to all my friends. Their support and accompany makes my life more interesting and happier.

Finally, my thanks go to my parents for their unconditional education, support, understanding and love during my whole life. They made me what I am. Without them, I am not be able to go through all the things, especially the difficulty I met.

Contents

List of Tables

List of Figures

Glossary

assortative mixed membership stochastic blockmodel (a-MMSB)

Bayesian Poisson tucker decomposition (BPTD)

Bayesian Poisson tensor factorization (BPTF)

Bayesian community detection (BCD)

Chinese restaurant process (CRP)

convolutional neural network (CNN)

copula mixed membership stochastic blockmodel (cMMSB)

graph convolutional network (GCN)

hierarchical Dirichlet process (HDP)

Indian buffet process (IBP)

infinite latent attribute model (ILA)

infinite multiple relational model (IMRM)

infinite relational model (IRM)

latent feature relational model (LFRM)

latent feature model (LFM)

mixed membership stochastic blockmodel (MMSB)

multiplex network embedding model (MNE)

multiscale community blockmodel (MCBM)

nested Chinese restaurant process (nCRP)

probabilistic matrix factorization (PMF)

singular value decomposition (SVD)

stochastic gradient Markov chain Monte Carlo (SG-MCMC)

stochastic variational inference (SVI)

stochastic blockmodel (SBM)

Chapter 1

Introduction

1.1 Problem Statement

Complex networks represent all kinds of systems that contain, and are based on, rational data. Their analysis and processing allow us to gain insight into structures they represent – we learn more about nodes themselves and even more about relations existing between them. For example, when we analyze e-commerce related networks we can find out more about preferences of individuals or relations between them and could provide so-called 'targeted advertising' of different products or services. In the case of a task of clustering nodes of knowledge graphs, nodes with similar characteristics can be 'put' together to form groups[1], Figure 1.1. In all these situations, the key is to discover relations between the nodes of a network, would they represent items or individuals.

Analysis of the existing work in such areas as artificial intelligence, machine learning and data mining leads to a conclusion that some aspects of data-based analysis of complex networks are considered to a limited degree. Besides, a few assumptions made during construction of Bayesian models limit their abilities to represent complexity of networks and to analyze them thoroughly. In particular, some of the most significant issues are: (1) latent class models, as typical Bayesian models used for network analysis, lack the ability to deal with scenarios where all relations within/across specific groups are masked; (2) existing modelling techniques that assume data independence could be replaced

[1] www.allthingsdistributed.com

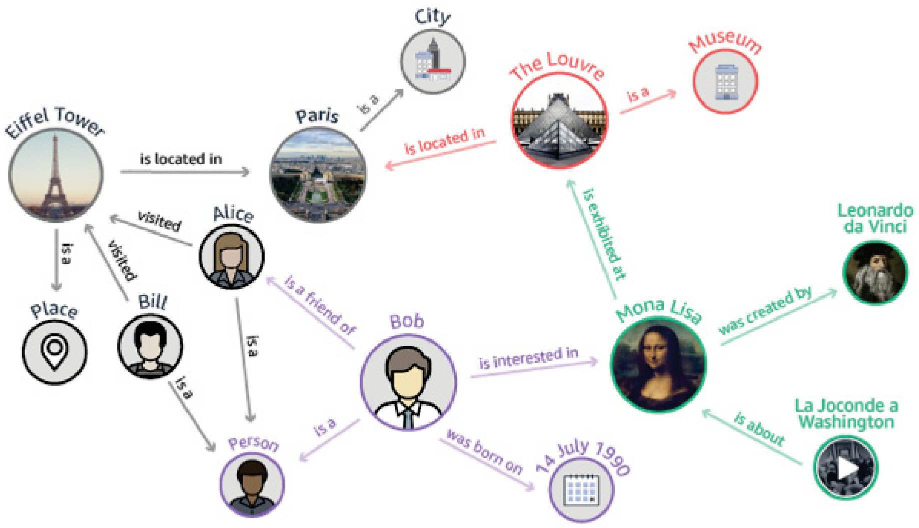


Figure 1.1: Knowledge Graph

by deep learning methods that promise more tailored for specific needs analysis of real data; (3) entity-based and linkage-based clustering methods used for analysis of network behaviour separately – one at a time – may lead problems related to interpretability and model precision; and finally (4) discretization of continuous data could cause information loss when complex networks are created and analysed. Therefore, we can postulate that models constructed under such cirumanstances are significantly simplified and although they are capable of achieving a state-of-art performance, such assumptions lead to simplified operations that could be inadequate at different scenarios. Below, we elaborate and illustrate consequences of these cases and assumptions.

1.1.1 Data Masking

Latent class models are one of classical methods used for modelling relations. Such methods focus on learning: (1) latent group for each entity; as well as (2) relations within/across groups. These models are able to predict/recover partially unobserved data. They allow to summarize the behaviour of the entities within/across group of the data. Yet, they are not very effective in the cases when observations of relations between entities within/across specific groups are fully masked. At the same time, masked data is quite common because of the security and privacy issues.

1.1.2 Data Independence

Nowadays, deep learning methods are quite successful in solving natural language and computer vision problems. Such technology shall be exploited in tasks related to analysis of complex network. One can envision an investigation whether the block of classical Bayesian model may be approximated using deep learning techniques. Such methods would address assumptions/limitations related to current Bayesian models. In particular, relational models usually involve two elements: group vectors representing the node latent information; and similarity matrix representing degrees of similarity between groups. From the perspective of a Bayesian model, the node's feature vector is often expressed as a multinomial distribution, while each entry of the similarity matrix is modelled as a Bernoulli distribution. Parameterizing the nodes information and/or similarity matrix in such a way implies data independence. However, such a strong assumption may limit the capability of a method to model dependencies between latent groups. In the real world, to analyze the relation between entities, it is more reasonable to consider the group-pair for each two entities dependently. In a more formal way, the conditional independence should be ruled out:

$$\mathrm{Pr}(z_A, z_B|\theta_A, \theta_B) \neq \mathrm{Pr}(z_A|\theta_A)\mathrm{Pr}(z_B|\theta_B)$$

1.1.3 Model Simplification

In the existing Bayesian relational models, there is an implicit assumption of considering either the latent classes (communities) of entities or groups of linkages singly but not simultaneously. For example, an entity-based clustering model focuses only on community clustering of entities ignoring potential grouping of linkages.

However, this will lead to at least two drawbacks. One of them is lack of interpretabiliy: an entity that belongs to one community can play different

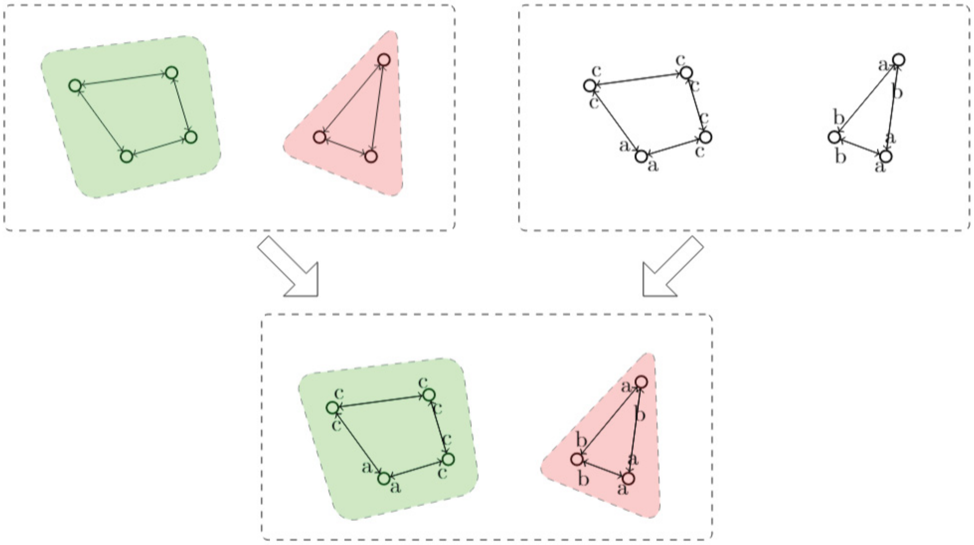


Figure 1.2: Example of Bayesian method for network analysis

roles in contact with other entities within/across communities. Additionally, these roles of the entity to others are also influenced by the other communities. Therefore, combing/considering entities and linkages is a demanding task, see an example in Figure 1.2. Another drawback is related to the model precision as the inappropriate size of community or group number may lead to an underfitting or overfitting problem. This becomes even more crucial in the case of a dynamic setting as the community evolution may cause the number of communities to vary with time. Here, difficulties arise in finding a method suitable to infer a number of parameters (communities, groups) during construction of a model.

1.1.4 Time Discretization

A process of constructing a model based on temporal data introduces another type of challenge – handling continuous time. In such cases, time discretization is often applied. However, it is not easy to determine a size of discretization interval in a way of all single events are captures, uneven distribution of events makes such a process almost impossible to succeed, see Figure 1.3. Whatever the technique of preprocessing is used, e.g., binarizing the data after time discretization, it leads to a loss of information and difficulties in extract realistic

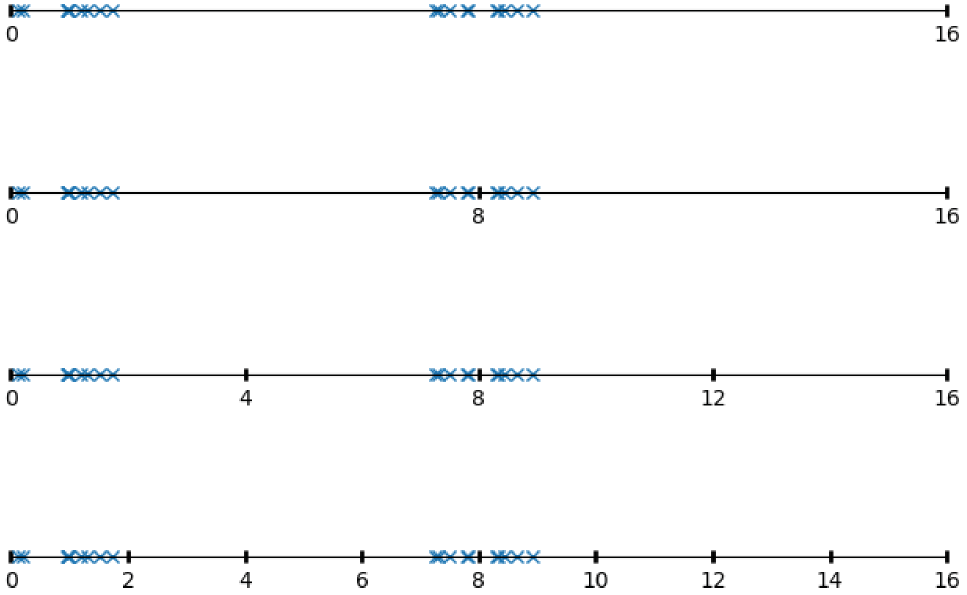


Figure 1.3: Example of time segmentation

character of data. In a case of complex networks, this is even more crucial, as the analysis of dynamics of network structure may be influenced by the inappropriate time discretization. One possible solution is to make discretization time based on a small time interval but that lead to high computational cost.

1.1.5 Summary of Introduced Issues

A number of critical issues that have been addressed in this work can be summarized in the form of following points:

- latent class models are not able to deal with situations where all relations within/across specific groups are masked;
- models of complex network assume data independence that can lead to inability to discover latent relations and dependencies that can be addressed using deep learning methods;
- Bayesian models consider only either entity-based clustering or linkage-

based clustering; and that may lead to two problems: lack of interpretability and decreased model precision;

- Bayesian models are not able to capture temporal changes in relations between entities due to a fixed size of model's components;
- discrete time-series models of continuous data with time discretization lead to a lost of information or/and characteristics of the data.

1.2 Objectives

The objectives of the work presented here is to address the mentioned above problems associated with analyzing and processing of complex networks. They be presented as a set of selected activities:

- Modelling coupling relations of latent groups and dependencies of nodes' latent features: this task focuses on relaxation of limitations related to conditional independence in the mixed membership stochastic blockmodel and deal with cases when relations with/across specific groups are fully masked. Specifically, we model the coupling relations within/among nodes (mixed membership) over time, and the dependence of relation within/across groups (similarity matrix).
- Clustering nodes and edges simultaneously: the current relational models focus on either node clustering or edge clustering. We identify the weakness of these approaches and propose to cluster the nodes and edges simultaneously in a discrete time.
- Modelling continuous data: process of time discretization lead to reduction of valuable information, especially in the case of reciprocating information between nodes. We apply the continuous clustering methods to model the community evolution, and stochastic process to model the reciprocating information.

1.3 Research Contributions

It is anticipated that the presented research work will lead to the following undertakings:

- proposing a deep learning method (multilayer network) that partially relaxes conditional independence assumption related to node's latent membership in Bayesian models – this has led to proposing an approximation of mixed membership stochastic blockmodel (Chapter 3);

- proposing a deep learning method (long short term memory) able to capture changes in the correlation between nodes' information (mixed membership) over time (Chapter 3);

- proposing the matrix factorization method to model the dependence of group relation (Chapter 3);

- constructing a two-level structure of complex network (enriching the model structure) in order to ensure preserving the full network structure by integrating the Bayesian nonparametrics method with mixed membership stochastic blockmodel (Chapter 4);

- utilizing the Bayesian nonparamterics method to model evolution of a network (correlation of network structure) over time (Chapter 4);

- using the Polya-Gamma data augmentation to increase the efficiency of the inference (Chapter 4);

- extending the static Bayesian nonparametric Hawkes process to the dynamic version utilizing the Bayesian nonparametric method (Chapter 5).

1.4 Organization

In Chapter 2, we briefly introduce the background and methods that are fundamental to our research, including neural network, lstm, Gibbs sampling, Poisson process. Besides, we also review the current literature related to relational models.

In Chapter 3, we refine the mixed membership stochastic blockmodel so the constructed model enable expressing: 1) coupling relations within/across the nodes by the multilayer network under the static setting; 2) coupling relations among communities using the matrix factorization; and 3) coupling relations among nodes using the long short term memory.

In Chapter 4, we enhance the ability of relational models via clustering the nodes and edges simultaneously. We use a discrete fragmentation coagulation process to cluster the nodes and mixed membership stochastic blockmodel to cluster the edges.

In Chapter 5, we focus on modelling the relational data in the continuous time. Specifically, we used the continuous fragmentation coagulation process to model the community evolution and Hawkes process to model the reciprocating relation among the nodes.

In Chapter 6, we provide the conclusion to our study and propose some ways to extend the current model.

Chapter 2

Preliminaries and Literature Review

2.1 Deep Learning

2.1.1 Neural Network

Human brain is capable of natural language process, pattern recognition, logic reasoning and etc. One research field of artificial intelligence is to imitate the function of the human brain by machine. The basic structure of the human brain is composed of interconnected biological neurons. Each neuron processes a simple operation, such as outputting a $[0, 1]$ signal according to the input. However, with the cooperation of millions of neurons, the human brain can perform a task like sensing, recognition and reasoning which is hard for the machine. Inspired by the structure of the human brain and the architecture of the biological neuron, the artificial neuron network is invented. One basic neuron network is composed of the input layer, hidden layer and the output layer. A basic neuron network is shown in Figure 2.1. Each layer is composed of the neuron nodes. The function of one neuron node is expressed as:

$$y = f(\sum_i w_i x_i + b)$$

where x_i is the input of the neuron node, and w_i and b is its associate weight and bias and f is the activation function. One choice of the activation function is the sigmoid function $\sigma(x)$:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

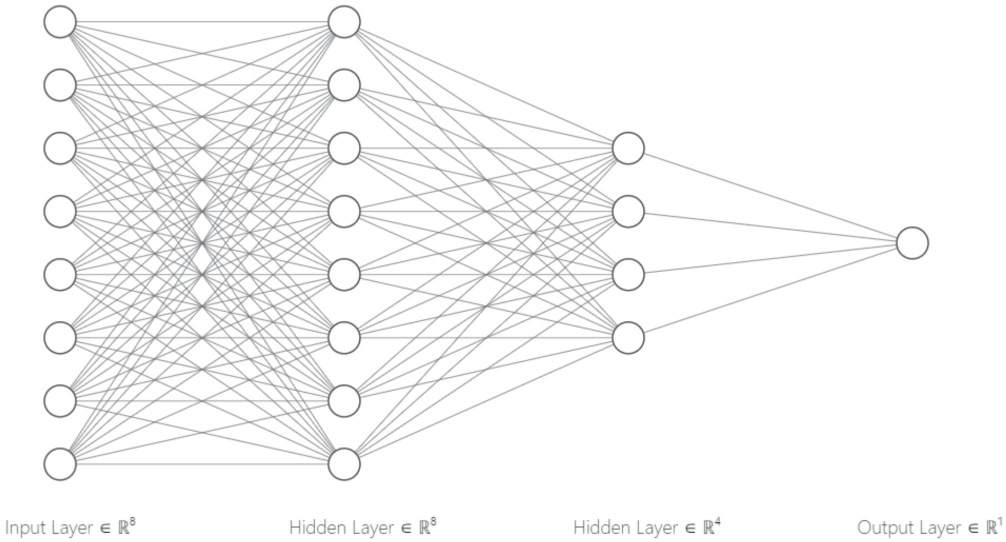


Figure 2.1: Example of neural network

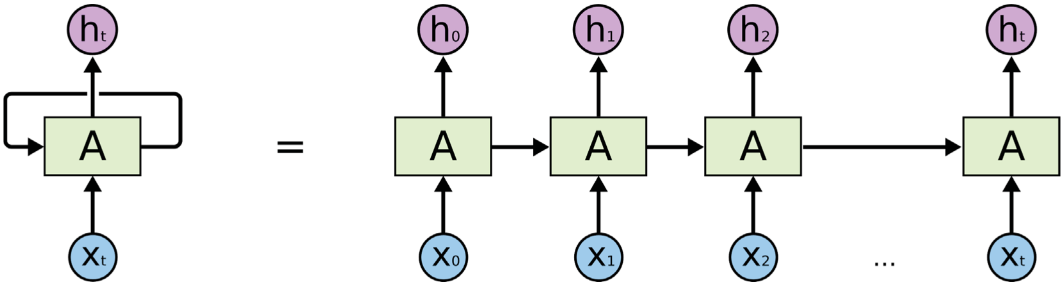


Figure 2.2: Recurrent neural network[1]

The reason behind that is the sigmoid function provides the nonlinearity and is easy to perform the derivative. Besides, for inference, the forward and backward propagation is utilized to learn the network weight $\boldsymbol{w}$. There are several types of neuron networks, such as perceptron [69], Hopfield net [33] and Kohonen Maps [42]. In our work, the perceptron is chosen to release the assumption of conditional independence.

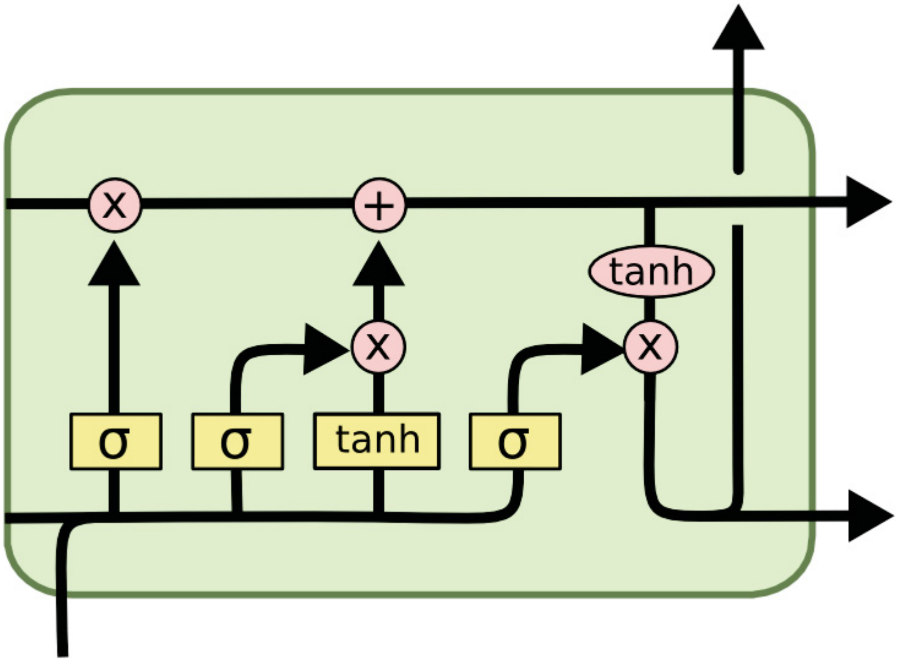


Figure 2.3: Long Short Term Memory[1]

2.1.2 Long Short Term Memory

The long short term memory [1] [30], called 'LSTM', is inspired by the recurrent neural network (RNN)[1]. The recurrent neural network is a deep learning method commonly used in the sequence learning such as speech recognition, text annotation and etc. The structure of RNN is shown in Figure 2.2. However, one limitation of the RNN is the lack of long dependency in useful information. For example, there is a sentence 'I speak English. ... I am fluent in English'. The task is to predict the word after the 'in'. However, due to the structure of RNN, the information about 'English' may be lost due to the information pass with a long interval. To overcome the limitation, the LSTM allows the past information to pass through the gate. The common structure of LSTM is shown in Figure 2.3. Like the σ in Figure 2.3, it will generate the output within the range $[0,1]$. If the output is really small, the past information will be thrown out and restored vice visa. There are several variations of LSTM, such as Gated RNNs [87] and Clockwork RNNs [45]. In our work, the classical LSTM is utilized to model the dependency between the feature.

2.2 Bayesian Nonparamertic

2.2.1 Dirichlet Process

In probability theory, Dirichlet process belongs to a bundle of stochastic processes of which realizations are a set of probability distributions. The Dirichlet process is composed of two parts: H, a base distribution and α a positive called the concentration parameter, denoted as $DP(\alpha, H)$. Formally, DP can be defined as follows [20]:

Theorem 1. *Let H be a probability distribution over a measurable space θ and α be a positive real number. Consider any finite measurable partition $A_1, ..., A_r$ of θ,*

$$\cup_{r=1}^{R} A_r = \theta, A_r \cap A_s = \emptyset, \forall r, s \tag{2.1}$$

[1] colah.github.io

Let G a random probability measure on θ. We say that G is a DP with base distribution H and concentration parameter α, denoted as $G \sim DP(\alpha, H)$, if the probability measure G on each partition over θ follows a Dirichlet distribution

$$(G(A_1), ..., G(A_r)) \sim Dir(\alpha H(A_1), ..., \alpha H(A_r)) \tag{2.2}$$

Theorem 2. *Let G be a Dirichlet process on (θ, H) with parameter α, and let $X_1, ..., X_n$ be a sample size n from G.Then the conditional distribution of G given $X_1, ..., X_n$, is as a Dirichlet process with parameter $\alpha + \sum_{i=1}^{n} \delta_{X_i}$:*

$$G|X_1, ..., X_n, \alpha, H \sim DP(\alpha + n, \frac{\alpha}{\alpha + n} H + \sum_{i=1}^{n} \delta_{X_i}) \tag{2.3}$$

2.2.2 Chinese Restaurant Process

The Chinese Restaurant Process (CRP) is a distribution over partitions of entities with a single parameter α. CRP can be easily depicted by showing how to draw the sample from it. Consider such a scenario that there is an infinite number of tables with infinite capacity in a restaurant. A sequence of N customers come to the restaurant to choose a table to sit. The indicator of customer i sitting at table k is denotes as $z_i = k$. The table to be chosen by the customer n can be sampled from:

$$\Pr(z_n|z_1, ..., z_{n-1}, \alpha) = \frac{\alpha}{\alpha + n - 1} \delta_K + \frac{1}{\alpha + n - 1} \sum_{k=1}^{K-1} n_k \delta_k \tag{2.4}$$

where $n_k = \sum_{i=1}^{n-1} \delta(z_i = k)$ is the number of first $n - 1$ customers sitting at table k.

2.3 Inference

2.3.1 Markov Chain Monte Carlo Methods

One common target for PGM is to find the optimal value for the latent variable. However, explicit analysis can be hard to perform on the PGM. In other words, it is intractable to find an analytical solution. Instead, normally an approximate solutions needs to be applied on the real problem. Therefore, Markov chain Monte Carlo (MCMC) methods become appealing and attract attention in the field of machine learning.

In general, MCMC methods provide a numerical solution to estimate a certain integration expression. The estimated integration based on the MCMC methods can be expressed:

$$\mathbb{E}_{x\sim f_X(x)}[g_(x)] = \int_x f_X(x)g_(x)dx \approx \frac{1}{N}\sum_{n=1}^{N} g(x_n) \tag{2.5}$$

The sequence of $x_1, ..., x_N$ is the samples based on $f_X(x)$ probability density distribution (PDF) constructed by Markov chain. Here we introduce two sampling techniques based on MCMC methods: Metropolis-Hastings sampling and Gibbs sampling.

Metropolis-Hastings Sampling

A reversible Markov chain can be constructed by the Metropolis-Hastings (MH) sampling. The reversible property is that the Markov chain must fulfill the requirement of the detailed balance:

$$\pi(x_t)p(x_{t-1}|x_t) = \pi(x_{t-1})p(x_t|x_{t-1}) \tag{2.6}$$

Where $\pi(x)$ is the target distribution (equivalent to the above PDF $f_X(x)$) and $p(x_{t-1}|x_t)$ is the transition probability. One important conclusion can be deduced by the detailed balance that:

$$\pi(x_t) = \int_{x_{t-1}} \pi(x_{t-1})p(x_t|x_{t-1}))dx_{t-1} \tag{2.7}$$

Therefore one way to check if a sampling method works is to verify whether the sampling method is satisfied with the detailed balance condition. In MH, $q(x_t|x_{t-1})$, proposal transition distribution, is used to generate the Markov chain. The basic MH sampling scheme is shown in Algorithm 1. A simple proof is given to show MH meets with the detail balance.

$$\pi(\mathbf{x}_t)q(\mathbf{x}_*|\mathbf{x}_t)\min(1, \frac{\pi(\mathbf{x}_*)q(\mathbf{x}_t|\mathbf{x}_*)}{\pi(\mathbf{x}_t)q(\mathbf{x}_*|\mathbf{x}_t)}) = \min(\pi(\mathbf{x}_t)q(\mathbf{x}_*|\mathbf{x}_t), \pi(\mathbf{x}_*)q(\mathbf{x}_t|\mathbf{x}_*)) \tag{2.8}$$

$$= \min(\pi(\mathbf{x}_*)q(\mathbf{x}_t|\mathbf{x}_*), \pi(\mathbf{x}_t)q(\mathbf{x}_*|\mathbf{x}_t)) \tag{2.9}$$

$$= \pi(\mathbf{x}_*)q(\mathbf{x}_t|\mathbf{x}_*)\min(1, \frac{\pi(\mathbf{x}_t)q(\mathbf{x}_*|\mathbf{x}_t)}{\pi(\mathbf{x}_*)q(\mathbf{x}_t|\mathbf{x}_*)}) \tag{2.10}$$

Algorithm 1 Metropolis-Hastings Sampling

```
Initialize $\mathbf{x}_0$
for $t = 0$ to $N - 1$ do
    Generate a uniform variate $u \sim U(0, 1)$
    Generate a proposed variate $\mathbf{x}_* \sim q(\mathbf{x}_*|\mathbf{x}_t)$
    if $u \leq \frac{\pi(\mathbf{x}_*)q(\mathbf{x}_t|\mathbf{x}_*)}{\pi(\mathbf{x}_t)q(\mathbf{x}_*|\mathbf{x}_t)}$ then
        $\mathbf{x}_{t+1} = \mathbf{x}_*$
    else
        $\mathbf{x}_{t+1} = \mathbf{x}_t$
    end if
end for
```

Gibbs Sampling

Gibbs sampling (GS), as a special case of MH, is widely used in MCMC methods. Consider a random vector variable $\mathbf{x} \in \mathbf{R}^2$ with a joint distribution $\pi(x^1, x^2)$. Suppose that the marginal distribution $\pi(x^1), \pi(x^2)$ over x^1 and x^2 can be derived and $\pi(x^1|x^2), \pi(x^2|x^1)$ are the conditional distributions of x^1 and x^2. The GS with 2 blocks scheme is shown in Algorithm 2. (Note: it is easy to extend GS with 2 blocks to GS with M blocks.)

Algorithm 2 Gibbs Sampling

```
Initialize $x_0^1, x_0^2$
for $t = 0$ to $N - 1$ do
    Generate a proposed variate $x_{t+1}^1 \sim \pi(x_{t+1}^1|x_t^2)$
    Generate a proposed variate $x_{t+1}^2 \sim \pi(x_{t+1}^2|x_{t+1}^1)$
end for
```

2.4 Stochastic Process

2.4.1 Counting Process

Definition 2.4.1. *Counting Process A stochastic process is a counting process $N(t)$ and satisfied with:*

1. $N(t) >= 0$

2. $N(t)$*is monotonic*

3. $N(t) \in \mathbb{Z}^+ = \{0, 1, \ldots\}$

2.4.2 Poisson Process

Definition 2.4.2. *Poisson Process A Poisson process is a counting process $N(t)$ with rate λ, of which the value belongs to $\mathbb{Z}^+ = \{0, 1, \ldots\}$, if satisfied with:*

1. $N(0) = 0$ *and* $N(t)$ *is monotonic;*

2. *The transition probability*

$$Pr_{mn}(h) = Pr[N(t+h) = n|N(t) = m]$$

are stationary, such that

$$Pr_{mn}(h) = \begin{cases} 1 - \lambda h + o(h) & \textit{if } n = m \\ \lambda h + o(h) & \textit{if } n = m+1 \\ o(h) & \textit{if } n > m+1 \end{cases}$$

The sequence of X_n are called the interarrival times of a Poisson process $N(t)$. The interarrival times X_n are independent and identically distributed RVs which follow an exponential distribution with parameter λ.

$$f(t) = \begin{cases} \lambda e^{-\lambda t} & \text{if } t >= 0 \\ 0 & \text{if } t < 0 \end{cases}$$

Theorem 3. *The Poisson process has the distribution with mean λt as:*

$$Pr(N(t) = n) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}$$

Memoryless Property of Poisson Process

Suppose the interarrival time X_n has arrived and the X_{n+1} has not arrived yet. And assume that there is no event arrived between $[t_n, t_n + m]$. The probability that the event X_{n+1} will not arrive before $t_n + m + t$ is calculated as

$$\begin{aligned}\Pr(\tau > t_n + m + t | \tau > t_n + m) =& \frac{e^{-\lambda(t_n+m+t)}}{e^{-\lambda(t_n+m)}} \\ =& e^{-\lambda t}\end{aligned}$$

It is found that the probability is independent of the previous events and the waiting time m. So the property of Poisson process is called memoryless.

Superposition of Poisson Process

Suppose that there are M independent Poisson process $N_m(t)$ with the rate λ_m where $m = 1, \dots, M$. The sum of Poisson process $N_m(t)$ is still a Poisson process $N(t)$ with the rate λ, $\lambda = \sum_{m=1}^{M} \lambda_m$.

Proof.

$$\begin{aligned}\mathbb{E}[e^{-tN}] =& \mathbb{E}[e^{-t\sum_{m=1}^{M} N_m}] \\ =& \prod_{m=1}^{M} \mathbb{E}[e^{-tN_m}] \\ =& \prod_{m=1}^{M} [\sum_{n=1}^{\infty} \frac{\lambda_m^n}{n!} e^{-\lambda_m} e^{-tn}] \\ =& \prod_{m=1}^{M} [e^{-\lambda_m} \sum_{n=1}^{\infty} \frac{\lambda_m^n}{n!} e^{-tn}] \\ =& \prod_{m=1}^{M} [e^{-\lambda_m} e^{\lambda_m e^{-t}}] \\ =& e^{\sum_{m=1}^{M} \lambda_m (1-e^{-t})}\end{aligned}$$

□

Decomposition of Poisson Process

Suppose there is a Poisson process $N(t)$ with the rate λ. Then the Poisson process $N(t)$ can be split into M independent Poisson process $N_m(t)$ with the rate $r_m\lambda$, $m = 1, \ldots, M$ and $\sum_{m=1}^{M} \lambda_m = \lambda$. Assume there are n_m arrival of events with Poisson process $N_m(t)$ and $n = \sum_{m=1}^{M} n_m$.

Proof.

$$\mathrm{P}(n_1, \ldots, n_m | n) = \frac{n!}{n_1! \cdots n_M!} r_1^{n_1} \cdots r_M^{n_M}$$

As n follows Poisson distribution with λ, then

$$\begin{aligned} \mathrm{P}(n_1, \ldots, n_m) =& \mathrm{P}(n_1, \ldots, n_m | n)\mathrm{P}(n) \\ =& \frac{n!}{n_1! \cdots n_M!} r_1^{n_1} \cdots r_M^{n_M} \frac{\lambda^n}{n!} e^{-\lambda} \\ =& \frac{(r_1\lambda)^{n_1}}{n_1!} e^{-r_1\lambda} \cdots \frac{(r_M\lambda)^{n_M}}{n_M!} e^{-r_M\lambda} \end{aligned}$$

□

2.4.3 Hawkes Process

Definition 2.4.3. *Hawkes Process (Hawkes process) Let a stochastic process be a counting process $N(t)$ with the history $\mathcal{H}_t$. The counting process $N(t)$ is a Hawkes process if the conditional intensity function $\lambda(t|\mathcal{H}_t)$ satisfies with the form:*

$$\lambda(t|\mathcal{H}_t) = \lambda^*(t) + \sum_{i;t>T_i} \phi(t - T_i)$$

where $\lambda^*(t)$ is called the base intensity function and $\phi(\cdot)$ is called the kernel function. The sequence of T_i is the event happened before t.

Likelihood for Hawkes Process

The conditional intensity $\lambda(t|\mathcal{H}_t)$ can be expressed in forms of the conditional probability density function $f(t|\mathcal{H}_{t_n}$ and its corresponding cumulative density function $F(t|\mathcal{H}_{t_n})$. For simplicity, denote the shorthand $\lambda(t)$ for the conditional intensity $\lambda(t|\mathcal{H}_t)$. The $\lambda(t)$ can be expressed as follows:

$$\begin{aligned}
\lambda(t)\,dt =& \frac{f(t|\mathcal{H}_{t_n})\,dt}{1-F(t|\mathcal{H}_{t_n})} \\
=& \frac{\Pr[t_{n+1} \in (t, t+dt)|\mathcal{H}_{t_n}]}{\Pr[t_{n+1} \notin (t_n, t)|\mathcal{H}_{t_n}]} \\
=& \frac{\Pr[t_{n+1} \in (t, t+dt), t_{n+1} \notin (t_n, t)|\mathcal{H}_{t_n}]}{\Pr[t_{n+1} \notin (t_n, t)|\mathcal{H}_{t_n}]} \\
=& \frac{\Pr[t_{n+1} \in (t, t+dt), t_{n+1} \notin (t_n, t), \mathcal{H}_{t_n}]}{\Pr[t_{n+1} \notin (t_n, t), \mathcal{H}_{t_n}]} \\
=& \Pr[t_{n+1} \in (t, t+dt)|t_{n+1} \notin (t_n, t), \mathcal{H}_{t_n}] \\
=& \Pr[t_{n+1} \in (t, t+dt)|\mathcal{H}_t]
\end{aligned}$$

To rewrite the $\lambda(t)$,

$$\begin{aligned}
\lambda(t) =& \frac{f(t|\mathcal{H}_{t_n})}{1-F(t|\mathcal{H}_{t_n})} \\
=& \frac{\frac{\partial}{\partial t}F(t|\mathcal{H}_{t_n})}{1-F(t|\mathcal{H}_{t_n})} \\
=& -\frac{\frac{\partial}{\partial t}(1-F(t|\mathcal{H}_{t_n}))}{1-F(t|\mathcal{H}_{t_n})} \\
=& -\frac{\partial}{\partial t}\ln(1-F(t|\mathcal{H}_{t_n}))
\end{aligned}$$

Integrate both side from t_n to t,

$$\begin{aligned}
\int_{t_n}^{t} \lambda(s)\,ds =& -\int_{t_n}^{t} \frac{\partial}{\partial s}\ln(1-F(s|\mathcal{H}_{t_n}))\,ds \\
=& -\ln(1-F(s|\mathcal{H}_{t_n}))|_{t_n}^{t} \\
=& -\ln(1-F(t|\mathcal{H}_{t_n}))
\end{aligned}$$

Arrange the above equation,

$$\begin{aligned}
&\int_{t_n}^{t} \lambda(s)\,ds = -\ln(1-F(t|\mathcal{H}_{t_n})) \\
&e^{-\int_{t_n}^{t}\lambda(s)\,ds} = 1-F(t|\mathcal{H}_{t_n}) \\
&F(t|\mathcal{H}_{t_n}) = 1-e^{-\int_{t_n}^{t}\lambda(s)\,ds}
\end{aligned}$$

Replace the $F(t|\mathcal{H}_{t_n})$ with the above formula into the expression of $\lambda(t)$,

$$\lambda(t) = \frac{f(t|\mathcal{H}_{t_n})}{1 - F(t|\mathcal{H}_{t_n})}$$
$$\lambda(t) = \frac{f(t|\mathcal{H}_{t_n})}{e^{\int_{t_n}^{t} \lambda(s)\, ds}}$$
$$f(t|\mathcal{H}_{t_n}) = \lambda(t) e^{-\int_{t_n}^{t} \lambda(s)\, ds}$$

Suppose the sequence of $T_1, \ldots, T_n$ are observed, the likelihood function can be expressed as:

$$\mathcal{L} = e^{-\int_0^{T_n} \lambda(s)\, ds} \prod_{i=1}^{n} \lambda(T_i)$$

2.5 Literature Review

2.5.1 Latent Class Model

The latent class model (LCM) is to model the data by its latent class. One classical LCM is the Gaussian mixture model (GMM). In GMM, to generate a data point x_i, a latent class z_i for the node u_i is first assigned, and then the data point is sample from the associated the Gaussian distribution $\mathcal{N}(\mu_{z_i}, \sigma_{z_i})$. In complex network, the stochastic blockmodel (SBM) [4], [32] is the first representative in LCM. The procedure to generate the relations between two entities in SBM can be decomposed to two steps. The entity i and j are firstly assigned to the latent class z_i and z_j respectively. Secondly, the relation will be generated according their associated entry of the class similarity matrix. One extension of SBM focuses on the prior of by Bayesian nonparametric. [38] applied the Chinese restaurant process (CRP) on the number of latent communities. One important extension can be traced back to the mixed membership stochastic blockmodel (MMSB) [3]. The key contribution of which is to allow each entity to hold multiple groups in a network. As MMSB is of importance to our work, the details of MMSB are given as follows:

MMSB aims at modelling the relation between entities. What distinguishes MMSB from previous work is that MMSB allows each entity to belong to each group to some degree. Assuming that there are K groups in one community.

MMSB uses a k dimensional vector θ_i of entity i to depict its membership with K latent groups. In MMSB, the sender indicator s_{ij} and the receiver indicator r_{ij} specify the latent group indicator for each directed relation x_{ij} between entity i and j which are derived by their membership respectively. Then each x_{ij} is determined by a compatibility matrix B with the sender indicator s_{ij} and the receiver indicator r_{ij} where the compatibility matrix represents the group relation. The generative process of MMSB can be described as follows:

- $\forall i \in \{1, 2, ..., N\}$
  - draw membership distribution $\theta_i \sim \text{Dirichlet}(\alpha)$
- $\forall k, l \in \{1, 2, ..., K\} \times \{1, 2, ..., K\}$
  - draw community relation $\mathbf{B}_{kl} \sim \text{Beta}(\lambda_1, \lambda_2)$
- $\forall i, j \in \{1, 2, ..., N\} \times \{1, 2, ..., N\}$
  - draw sender indicator $s_{ij} \sim \text{Multinomial}(\theta_i)$
  - draw receiver indicator $r_{ij} \sim \text{Multinomial}(\theta_j)$
  - draw relation $x_{ij} \sim \text{Bernoulli}(\mathbf{B}_{s_{ij} r_{ij}})$

MMSB also introduced a sparsity parameter to account for non-interaction between entities being due to limited opportunities for interaction rather than holding entities different groups.

Bi-LDA [66]implemented the MMSB in recommendation system. In the model, each user and movie is represented as a mixed membership vector and an rating matrix is used to present the relation between communities from users and movies. [49] made extensions to Bi-LDA to develop a dynamic version to model changes of users' interests. [23] forwarded a more efficient inference on the prediction of recommendation system.

Many works made extensions to MMSB. The assortative mixed membership stochastic blockmodel (a-MMSB) [25] derived an efficient inference on MMSB by stochastic variational inference (SVI). Integrating a-MMSB with nested Chinese restaurant process (nCRP) which constructed a hierarchical

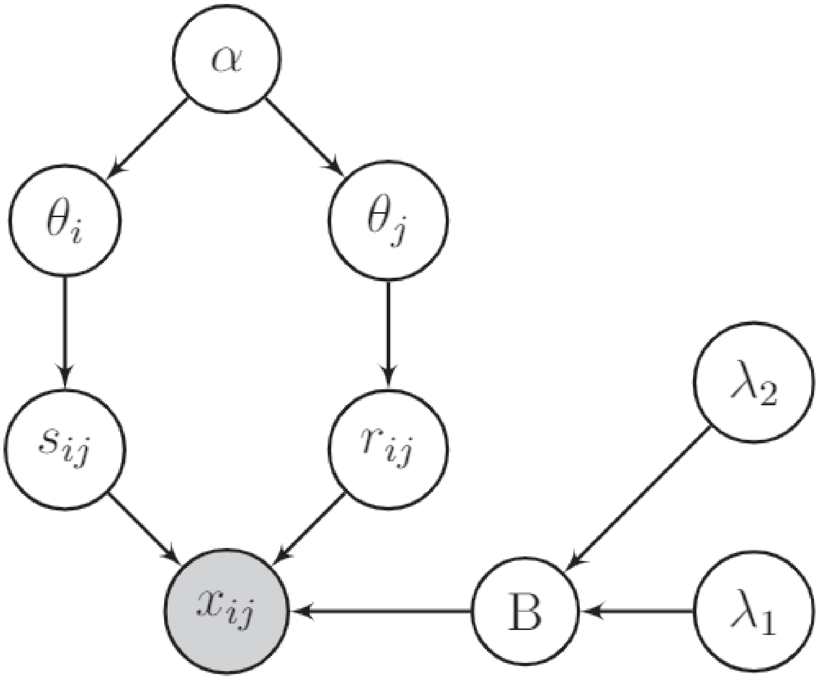


Figure 2.4: Graphical model of MMSB.

structure with the CRP, multiscale community blockmodel (MCBM) [29] leads to the construction of a tree with infinite children (nodes). Each entity belongs to one node that represents a community. To construct relations between entities at different levels, the entities share the compatibility matrix of the same parent nodes. [52] introduced the stochastic gradient Markov chain Monte Carlo (SG-MCMC) derive an efficient inference. One type of extensions focuses on the communities modelling by Bayesian nonparametric. Similar to the extensions, the infinite relational model (IRM) [38], to SBM, [37] also introduced the CRP to model the evolution of communities including (split, merge, appearance and disappearance). The Bayesian community detection (BCD) [60] paid attention on the difference between the relation of within/across communities. [39] proposed the hierarchical Dirichlet process (HDP) on the model with a scalable inference algorithm. [16] also implemented the HDP on the MMSB, and introduced the correlation and dynamics to the communities. Another extension, the copula mixed membership stochastic blockmodel (cMMSB) [17], used the Copula function to introduce a dependence between

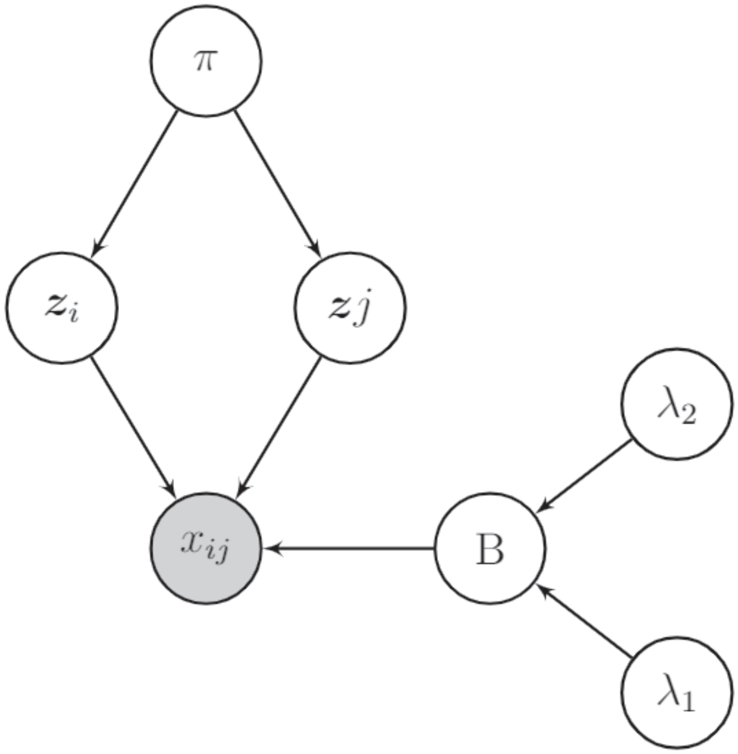


Figure 2.5: Graphical model of latent feature model.

membership indicators. [22], [83] incorporated the dynamics to MMSB by introducing the correlation of latent communities via Gaussian distribution. [19] proposes a deep and scalable version of the MMSB.

2.5.2 Latent Feature Model

Another class of relational models is the latent feature model (LFM). The LFM provides another way to model the relations between entities via assigning the latent features to each entity instead of the latent classes. And another feature compatibility to represent the relation between features. One classical graphical model of LFM is presented in Figure 2.5. Assume that there are N entities and K features. The procedure of the LFM generative model is given as follows:

- $\forall i \in \{1, 2, ..., N\}$
  - $\forall k \in \{1, 2, ..., K\}$

    - ∗ draw feature indicator $z_{ik} \sim \text{Bernoulli}(\pi_k)$

- $\forall k, l \in \{1, 2, ..., K\} \times \{1, 2, ..., K\}$
  - draw feature relation $\mathbf{B}_{kl} \sim \text{Beta}(\lambda_1, \lambda_2)$
- $\forall i, j \in \{1, 2, ..., N\} \times \{1, 2, ..., N\}$
  - draw relation $x_{ij} \sim \text{Bernoulli}(\sigma(\boldsymbol{z}_i^{\intercal}\mathbf{B}\boldsymbol{z}_j))$

where $\sigma(\cdot)$ is the sigmoid function.

One representative of LFM is the latent feature relational model (LFRM) [57] which induced the Indian buffet process (IBP) [79],[27] to determine the number of features and feature selection for each entities. The dynamic relational infinite feature model[21] furtherly extended LFRM for the longitudinal network. The infinite multiple relational model (IMRM) [61] addressed the challenge of computation cost from the LFM. [63] extended [57] by introducing the subcluster within the latent feature by Dirichlet process in the infinite latent attribute model (ILA). [92] focuses on utilizing hierarchical gamma process on static networks mainly. [86][85] make substantial contributions of incorporating the completely random measures into the modelling.

2.5.3 Matrix Factorization Model

Matrix factorization model is another important class in relation modelling. And it is also closely related to the recommender system. One classical matrix factorization model is the singular value decomposition (SVD) where the rating matrix $R^{M \times N}$ is factorized into two matrix: the user matrix $U^{M \times K}$ and the movie matrix $M^{K \times N}$. The learned user and movie matrix can be used to predict the unobserved ones. On basis of SVD, [43], called SVD++, took the bias of users and movies into consideration. [44] introduced the dynamics into the SVD++ by combining the static features of uses and movies with the time dependent components. The probabilistic matrix factorization (PMF) [59] is the first probabilistic method on matrix factorization. The Bayesian nonparametric method [26] is also taken into the consideration. [25] used the hierarchical structure to capture the sparse factors and modelled the long-tail properties of users and items. The Bayesian Poisson tensor factorization (BPTF) [71] considered the tensor factorization by introducing another component along with the component of the user, item and time. The Bayesian Poisson tucker decomposition (BPTD) [72] took the advantage of the event tokens, event types and multinetwork snapshots to learn the structure of the relations. [91] placed the Gaussian process and Hawkes process on tensor decomposition to model the effects between events. [34] incorporated the multi-level side information with the non-negative matrix factorization. [76] levaraged the multilayer and temporal networks to capture the information of both nodes and linkages.

2.5.4 Deep Learning Model

With the rise of deep learning, several deep architectures have been proposed for network modeling. The discussion is limited within two types of networks: the word-embedding model and the graph convolutional network.

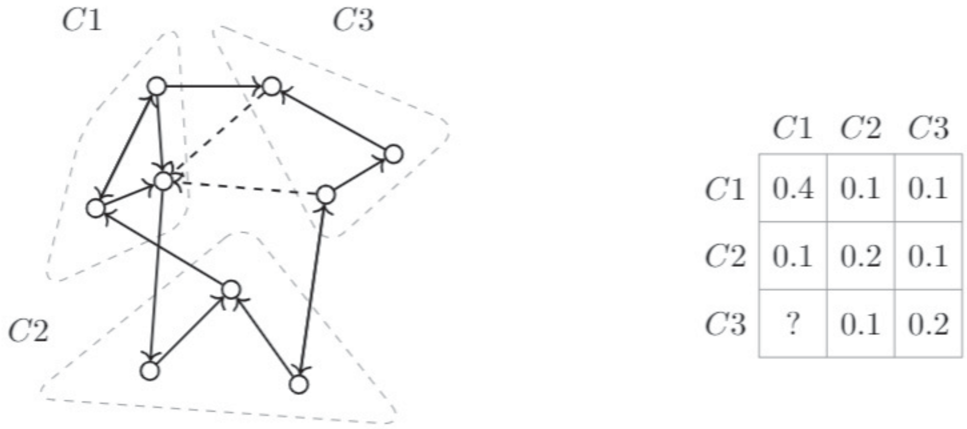
Word embedding models present the opportunity to capture more complex interactions than the flat methods mentioned previously. DeepWalk [64] fused deep learning with network modeling by applying the SkipGram language model on information obtained from random walks on the network. This allows it to learn a latent feature representation for each entity. While successful at representing network entities as low dimensional embeddings, this approach has not performed as well on link prediction tasks [90]. The *relational deep learning* [82] model uses a deep hierarchical Bayesian structure which captures relational information necessary for link prediction. Recently, [35] have proposed the hierarchical latent feature model which incorporates side information in the form of node attributes and models latent features using a network structure similar to that of deep belief nets. In terms of incorporating varying relational information, the multiplex network embedding model (MNE) [90] unifies different relations into a single embedding space.

The graph convolutional network (GCN) [41] is another architecture in complex networks. The GCN is supposed to generalize the convolutional neural network (CNN) from the domain of computer vision, natural language processing to the domain of complex networks or social networks. One important component of the CNN is the convolutional layer which can be interpreted as a convolution operation on a signal or an image. The convlutional layer will aggregate the characteristic of the information capture the structure in a localized area as the input of the next layer. In the complex network, it is hard to implement the convolutional operator directly. Instead, the spectral graph theory [9] is developed to encode the graph structure. More recent development [74] in this area established a theoretical formulation of CNN on graphs motivated by graph signal processing (GSP). Spectral Networks and Deep Locally Connected Networks on Graphs [8] proposed a spectral construction on the graph Laplacian. [11] extended the [8] to enhance the spectral filter and approximate it with the Chebyshev polynomials [28]. The GCN furtherly simplified [11] with the first order approximation of the Chebyshev polynomials. [73] as a application of the GCN had a good performance on the task of the knowledge graph completion and [88] worked well in the recommender system.

Chapter 3

Deep Dynamic Mixed Membership Stochastic Blockmodel Based Network for Link Prediction

Latent community models are successful at statistically modeling network data by assigning network entities to communities and modelling entity relations as the relations of their communities. In this chapter, our contributions are from two aspects: (1) We describe the limitation of these models in inferring relations between two communities when the entity relations between these communities are unobserved. We propose a solution to this problem by factorizing the community relations matrix into two community feature matrices, thereby adding a dependency between community relations. (2) We utilize the deep learning techniques to approximate the components of classical probabilistic relational model for complex network. We introduce the *deep dynamic mixed membership stochastic blockmodel based network* (DDBN) to demonstrate the feasibility of such an approach. Our model takes the advantage of the matrix factorizaiton to solve the above problem and marries the *mixed membership stochastic blockmodel* (MMSB) with deep neural networks for rich feature extraction and introduces a temporal dependency in latent features using a long short-term memory unit for dynamic network modeling. We evaluate our model on the link prediction task in static and dynamic networks and find that our model achieves comparable results with state-of-the-art methods.



	$C1$	$C2$	$C3$
$C1$	0.4	0.1	0.1
$C2$	0.1	0.2	0.1
$C3$	?	0.1	0.2

Figure 3.1: An example illustrating the problem of unobserved community relations.

3.1 Introduction

In today's increasingly interconnected world, networks are a useful tool for capturing complex structures in relations such as those found between friends, sports teams, email exchanges, and academic papers. Statistical modeling of these networks is a challenging and long studied problem going back to the social sciences at the turn of the 20^{th} Century [24]. Its goal is to discover a statistical representation of a network's entity relations and, if the network is dynamic, model their changes over time. This representation can then be used to solve common problems in artificial intelligence research such as missing data completion, clustering, and network forecasting. Statistical modeling has been used on data found in areas ranging from biology [3] to social networks [21] to recommender systems [49].

Consider an input network as represented by an adjacency matrix describing the relations (edges) between network entities (nodes). One approach at modeling such a network which has received considerable attention, blockmodeling, decomposes it into communities that share similar properties and assigns entities with membership to them. Conceptually, these communities may be thought of as clusters of entities. Relations between communities are the degree of compatibility between two communities and are modeled in a community relations matrix. Thus, relations exist on two levels: entity and community. Blockmodels model a relation between two entities as the relation between their respective communities. One limitation of this approach arises when all entity relations between communities are unobserved. To illustrate this, consider the toy example in Figure 3.1 which presents a graphical representation of a network as well as the community relations matrix for its three communities ($C1$, $C2$, $C3$). Solid lines represent observed relations between entities (white circles) whereas dashed lines represent unobserved relations. In this network, relations from entities belonging to $C3$ going to entities belonging to $C1$ are unobserved. Blockmodels, therefore, have no information to draw upon when inferring the community relation between $C3$ and $C1$. Thus the value of the relation is reduced to a prior, represented by a question mark in the community relations matrix.

In this work, we propose the *deep dynamic mixed membership stochastic blockmodel based network* (DDBN) which extends the *mixed membership stochastic blockmodel* (MMSB) [3] and overcomes the aforementioned limitation by factorizing the community relations matrix into two community feature matrices. This approach introduces a dependency between community relations which allows our model to use information from observed community relations to infer unobserved community relations. Furthermore, we replace the probabilistic framework of the MMSB with a multilayer network (MLN) architecture. This hierarchical approach allows for extracting richer latent features and for modeling the interactions between them. Finally, our model introduces a temporal dependence between latent features via a long short-term memory (LSTM) [30] recurrent neural network (RNN). The RNN learns the temporal changes in latent features, thereby providing a natural extension of the MMSB to the dynamic setting and eliminating the need for modeling each time-step in the dynamic network separately.

The remainder of this chapter is organized as follows. Section 2 formalizes the problem of unobserved relations in the MMSB and describes our proposed model. Section 3 compares the factorized and unfactorized community relations matrices and presents our model's performance on link prediction tasks

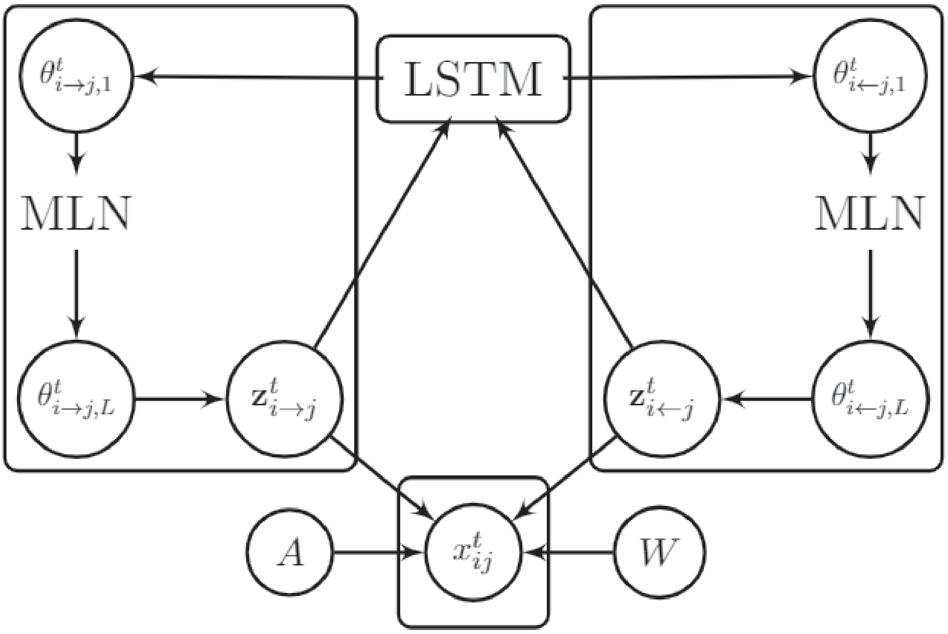

$\theta^t_{i\rightarrow j,1}$
LSTM
$\theta^t_{i\leftarrow j,1}$
MLN
MLN
$\theta^t_{i\rightarrow j,L}$
$\mathbf{z}^t_{i\rightarrow j}$
$\mathbf{z}^t_{i\leftarrow j}$
$\theta^t_{i\leftarrow j,L}$
A
x^t_{ij}
W

Figure 3.2: Graphical model of DBNN

using real world and synthetic datasets. Section 4 concludes the work.

3.2 Model Description

In this section, we first formulate the problem of network modeling, then describe the MMSB modeling approach along with its limitation. Further, we present our proposed model as it applies to static networks. Finally, we extend our static model to handle dynamic networks thereby completing the description of our full model.

3.2.1 Problem Formulation

We define a static network as an $N \times N$ binary adjacency matrix, $\mathbf{X}$, that represents the directed relationships between N entities such that $x_{ij} = 1$ if a relationship from entity e_i to entity e_j exists and $x_{ij} = 0$ otherwise. Given a fixed number of communities, K, we represent the set of entities that belong to community p as C_p for $p \in \{1, 2, ..., K\}$. The relations between communities are modeled by a $K \times K$ community relations matrix, $\mathbf{B}$, where b_{pq} represents the probability of a relation from an entity in C_p to an entity in C_q. In the MMSB, the community membership of entities are represented by two membership indicators, $\mathbf{z}_{i\rightarrow j}$ and $\mathbf{z}_{i\leftarrow j}$ for sender e_i and receiver e_j, respectively. Both $\mathbf{z}_{i\rightarrow j}$ and $\mathbf{z}_{i\leftarrow j}$ are one-hot vectors of size K that assign e_i and e_j to one community. The generative process is described as follows:

- $\forall i \in \{1, 2, ..., N\}$
  - draw membership distribution $\theta_i \sim \text{Dirichlet}(\alpha)$
- $\forall p, q \in \{1, 2, ..., K\} \times \{1, 2, ..., K\}$
  - draw community relations $b_{pq} \sim \text{Beta}(\lambda_1, \lambda_2)$
- $\forall i, j \in \{1, 2, ..., N\} \times \{1, 2, ..., N\}$
  - draw sender's indicator $\mathbf{z}_{i\rightarrow j} \sim \text{Multi}(\theta_i)$
  - draw receiver's indicator $\mathbf{z}_{i\leftarrow j} \sim \text{Multi}(\theta_j)$
  - draw relation $x_{ij} \sim \text{Bernoulli}(\mathbf{z}_{i\rightarrow j}\mathbf{B}\mathbf{z}_{i\leftarrow j})$

Where α and λ are the priors for θ and $\mathbf{B}$, respectively. One limitation of this model arises when the relations between two communities are unobserved. To illustrate this, consider a scenario where for two communities, p and q, all relations from C_p to C_q are unobserved. The value of $\mathbf{B}$ is inferred on its posterior: $p(\mathbf{B}|\mathbf{X}, \lambda_1, \lambda_2) \sim p(\mathbf{X}|\mathbf{B})p(\mathbf{B}|\lambda_1, \lambda_2)$. When all entity relations are unknown, the posterior of $\mathbf{B}$ is reduced to its priors: $p(\mathbf{B}|\lambda_1, \lambda_2)$. Thus, when relations from C_p to C_q are masked, b_{pq} is reduced to its priors, $p(b_{pq}|\lambda_1, \lambda_2)$.

3.2.2 Static Model

Our model overcomes this limitation by introducing matrices $\mathbf{A} \in \mathbb{R}^{K\times M}$ and $\mathbf{W} \in \mathbb{R}^{M\times K}$ and factorizing the community relations matrix as $\mathbf{B} = \mathbf{AW}$. $\mathbf{A}$ and $\mathbf{W}$ may be thought of as feature matrices for communities such that $\mathbf{a}_p$ and $\mathbf{w}_q^T$ are the M dimensional feature vectors for C_p when it is the sender and C_q when it is receiver. (Here $\mathbf{a}_p$ is the p^{th} row vector of $\mathbf{A}$ and $\mathbf{w}_q^T$ is the q^{th} column vector of $\mathbf{W}$.) The value of $M \geq 1$ is a hyperparameter that adds flexibility to our model by controlling the size of the community features. The relation from community p to community q is therefore modeled as the dot product of community p's sender feature and community q's receiver feature, $b_{pq} = \mathbf{a}_p \mathbf{w}_q^T$. It should be noted that our approach introduces a dependency between communities and models the correlations between them. This is a departure from MMSB in which community relations are independent. To increase the interpretability of $\mathbf{AW}$, we normalize it to the range $(0, 1)$ by passing it through the logistic sigmoid function, $\sigma(\mathbf{AW}) = 1/(1+\exp(-\mathbf{AW}))$.

We replace the MMSB sampling scheme for $\mathbf{z}_{i\rightarrow j}$ and $\mathbf{z}_{i\leftarrow j}$ with a MLN of L fully-connected layers. This architecture takes advantage of the hierarchical nature of MLNs to extract latent entity features and model the correlations between them. We denote these features as $\theta_{i\rightarrow j,l}$ and $\theta_{i\leftarrow j,l}$ for $l \in \{1, 2, ..., L\}$, representing the latent feature at layer l for e_i when it is the sender and receiver, respectively. We follow feature extraction with a softmax layer applied to the last latent features in the MLN, $\theta_{i\rightarrow j,L}$ and $\theta_{i\leftarrow j,L}$. In contrast to MMSB, this allows for partial membership of an entity to a community. It is important to note that community membership indicators are independent of the other actor in the relation but dependent on their role in it (i.e. sender or receiver).

The features at each layer l as well as the values $\mathbf{A}$ and $\mathbf{W}$ are learned by stochastic gradient descent, minimizing the following objective function:

$$\mathcal{L}(\mathbf{X}, \mathbf{X}') = -\sum_{i,j} [\mathbb{I}[x_{ij} = 1]\ln(x'_{ij}) + \mathbb{I}[x_{ij} = 0]\ln(1 - x'_{ij})]$$

Where $\mathbf{X}'$ is the matrix of predicted entity relations and $\mathbb{I}$ is the indicator function. The full generative model is described as follows:

- $\forall i, j \in \{1, 2, ..., N\} \times \{1, 2, ..., N\}$
  - $\forall l \in \{1, 2, ..., L\}$
    - update $\theta_{i\rightarrow j,l} = \sigma(\mathbf{R}_{S,l}\theta_{i\rightarrow j,l-1} + \mathbf{d}_{S,l})$
    - update $\theta_{i\leftarrow j,l} = \sigma(\mathbf{R}_{R,l}\theta_{i\leftarrow j,l-1} + \mathbf{d}_{R,l})$
  - update sender's membership indicator
    $\mathbf{z}_{i\rightarrow j} = \text{Softmax}(\mathbf{R}_{S,z}\theta_{i\rightarrow j,L} + \mathbf{d}_{S,z})$

– update receiver's membership indicator
  $\mathbf{z}_{i\leftarrow j} = \text{Softmax}(\mathbf{R}_{R,z}\theta_{i\leftarrow j,L} + \mathbf{d}_{R,z})$

– linkage $x_{ij} \sim \text{Bernoulli}(\sigma(\mathbf{z}_{i\rightarrow j}[\mathbf{AW}]\mathbf{z}_{i\leftarrow j}))$

Here, $\mathbf{R}_{S,l}$, $\mathbf{R}_{R,l}$, $\mathbf{d}_{S,l}$, $\mathbf{d}_{R,l}$ are the MLN weights and biases at layer l for the sender and receiver, respectively. The softmax weights and biases are denoted similarly, using z in their subscripts: $\mathbf{R}_{S,z}$, $\mathbf{R}_{R,z}$, $\mathbf{d}_{S,z}$, $\mathbf{d}_{R,z}$.

3.2.3 Dynamic Model

We begin by considering the drawbacks of applying our static model independently for each time-step. First, the indices of communities in the community relations matrix may change, resulting in uninterpretable community relations and membership indicators across time-steps. Furthermore, there is no mechanism for transmitting information from previous time-steps. Having such a mechanism would counteract the problem of missing data in sparse datasets. Finally, such an approach does not allow for network forecasting as there is no capacity for predicting θ values in successive time-steps.

Before we proceed with the description of the dynamic model, we extend our notation from static networks to the dynamic setting. We represent a dynamic network with T time-steps, $\mathbf{X}$, as a time-series of T static $N \times N$ binary matrices $\mathbf{X}^t$ for each time-step $t \in \{1, 2, ..., T\}$. We extend our notation by adding a temporal dimension to our membership indicators and latent features as $\mathbf{z}^t_{i\rightarrow j}$, $\mathbf{z}^t_{i\leftarrow j}$, $\theta^t_{i\rightarrow j,l}$, $\theta^t_{i\leftarrow j,l}$.

In our proposed dynamic model, the community relations matrix is shared across time-steps, thus assuming that communities and their relations are invariant across time. This assumption is not applied to community memberships, which we model by adding a temporal dependence on latent entity features. Markov probability models, used in previous dynamic models [21], [83], apply the Markov assumption, $p(\theta^{t+1}|\theta^t, ..., \theta^1) = p(\theta^{t+1}|\theta^t)$, to model the relation between θ^{t+1} and θ^t. We relax this assumption to account for long term temporal dependencies between latent features. To model these dependencies, we employ an LSTM component. LSTMs are a type of RNN that learn dependencies by controlling the inflow and outflow of information into their memories. This allows them to model long and sporadic temporal dependencies without encountering the problem of vanishing gradients suffered by other types of RNNs. The key component of the LSTM, the memory cell c^t, stores information from previous time-steps. Its status is controlled by three gates: input i^t, forget f^t, output o^t. Our input for predicting θ_i^{t+1} is the concatenation of membership indicators at the previous time-step, denoted as $[\mathbf{z}_{i\to j}^t, \mathbf{z}_{i\leftarrow j}^t]$. This approach is similar to the topic variant of Latent LSTM Allocation [89]. Our LSTM can be formalized as follows:

$$
\begin{aligned}
&\mathbf{x}^t = [\mathbf{z}_{i\to j}^t, \mathbf{z}_{i\leftarrow j}^t] \\
&\mathbf{i}^t = \sigma(\mathbf{S}_{xi}\mathbf{x}^t + \mathbf{S}_{yi}\mathbf{y}^{t-1} + \mathbf{S}_{ci} \circ \mathbf{c}^{t-1} + \mathbf{d}_i) \\
&\mathbf{f}^t = \sigma(\mathbf{S}_{xf}\mathbf{x}^t + \mathbf{S}_{yf}\mathbf{y}^{t-1} + \mathbf{S}_{cf} \circ \mathbf{c}^{t-1} + \mathbf{d}_f) \\
&\mathbf{c}^t = \mathbf{f}^t \circ \mathbf{c}^{t-1} + \mathbf{i}^t \circ \tanh(\mathbf{S}_{xc}\mathbf{x}^t + \mathbf{S}_{yc}\mathbf{y}^{t-1} + \mathbf{d}_c) \\
&\mathbf{o}^t = \sigma(\mathbf{S}_{xo}\mathbf{x}^t + \mathbf{S}_{yo}\mathbf{y}^{t-1} + \mathbf{S}_{co}\mathbf{c}^t + \mathbf{d}_o) \\
&\mathbf{y}^t = \mathbf{o}^t \circ \tanh(\mathbf{c}^t) \\
&\mathbf{y}^t = [\theta_{i\to j,1}^{t+1}, \theta_{i\leftarrow j,1}^{t+1}]
\end{aligned}
$$

Where $\mathbf{S}$ and $\mathbf{d}$ are the weights and biases of the LSTM, subscripted by the gates they are associated with. $\mathbf{x}$ and $\mathbf{y}$ are the LSTM inputs and outputs and $\circ$ is the Hadamard product.

We distribute a fixed static model across time such that $\mathbf{A}$, $\mathbf{W}$, $\mathbf{R}$, and $\mathbf{d}$ are the same at each time-step. The LSTM links the static models through the first latent features at each time-step, $\theta_{i\to j,1}^t$ and $\theta_{i\leftarrow j,1}^t$. This relationship is outlined graphically in Figure 3.2, which describes the model architecture at one time-step. We train our model alternately such that at each training iteration we first backpropagate on the full dynamic model and then backpropagate the static model. In our experiments, this increases stability and decreases convergence time. Algorithm 3 outlines the training procedure for DDBN.

Algorithm 3 Training Procedure for DDBN

Input: Time series of adjacency matrices $\mathbf{X}'$; number of communities K; dimensionality of community features M

Output: Predicted entity relations as time series of adjacency matrices $\mathbf{X}$

```
Initialize MLN and LSTM weights $\mathbf{R}$ and $\mathbf{S}$
repeat
    for $i, j \in \{1, 2, ..., N\} \times \{1, 2, ..., N\}, i \neq j$ do
        Obtain $\theta^1_{i\rightarrow j,L}$ and $\theta^1_{i\leftarrow j,L}$ by MLN forward pass
        Update $\mathbf{z}^1_{i\rightarrow j} = \text{Softmax}(\mathbf{R}_{S,z}\theta^1_{i\rightarrow j,L} + \mathbf{d}_{S,z})$
        Update $\mathbf{z}^1_{i\leftarrow j} = \text{Softmax}(\mathbf{R}_{R,z}\theta^1_{i\leftarrow j,L} + \mathbf{d}_{R,z})$
        Update $x^1_{ij} = \sigma(\mathbf{z}^1_{i\rightarrow j}[\mathbf{AW}]\mathbf{z}^1_{i\leftarrow j})$
    end for
    for $t \in \{2, 3, ..., T\}$ do
        for $i, j \in \{1, 2, ..., N\} \times \{1, 2, ..., N\}, i \neq j$ do
            Obtain $\theta^t_{i\rightarrow j,1}$ and $\theta^t_{i\leftarrow j,1}$ by LSTM forward pass
            Obtain $\theta^t_{i\rightarrow j,L}$ and $\theta^t_{i\leftarrow j,L}$ by MLN forward pass
            Update $\mathbf{z}^t_{i\rightarrow j} = \text{Softmax}(\mathbf{R}_{S,z}\theta^t_{i\rightarrow j,L} + \mathbf{d}_{S,z})$
            Update $\mathbf{z}^t_{i\leftarrow j} = \text{Softmax}(\mathbf{R}_{R,z}\theta^t_{i\leftarrow j,L} + \mathbf{d}_{R,z})$
            Update $x^t_{ij} = \sigma(\mathbf{z}^t_{i\rightarrow j}[\mathbf{AW}]\mathbf{z}^t_{i\leftarrow j})$
        end for
    end for
    for mini-batch do
        Compute the gradients for objective function $\mathcal{L}(\mathbf{X}, \mathbf{X}') = -\sum_{i,j,t}[\mathbb{I}[x^t_{ij} = 1]\ln(x' t_{ij}) + \mathbb{I}[x^t_{ij} = 0]\ln(1 - x' t_{ij})]$ w.r.t $\mathbf{A}$, $\mathbf{W}$, $\mathbf{R}$, $\mathbf{d}$, $\mathbf{S}$ and update using Adam optimizer [40], repeat for $\mathbf{A}$, $\mathbf{W}$, $\mathbf{R}$, $\mathbf{d}$
    end for
until convergence
```

3.3 Evaluation

We evaluate our model performance on the link prediction task wherein a subset of relations between pairs of entities is masked and the goal is to predict the value of these masked relations. We divide our model evaluation into three subsections, each building upon the previous in terms of model complexity. First we compare the performance of **B** to our factorized community relations matrix **AW** and provide the intuition behind this approach. This is followed by evaluating our static and dynamic models separately and comparing to results found in the literature. The metric used in evaluation is the area under the receiver operating characteristic curve (AUC) calculated by performing a threshold sweep on the true positive rate plotted against the false positive rate. In all of our experiments, we use the Adam [40] optimizer when performing stochastic gradient descent.

3.3.1 Dataset Overview

We use seven real-world datasets in our evaluation procedures: MIT Reality [14]; NIPS Collaboration [70]; Soccer transfers; Lazega [47]; Coleman [10]; Temporal EU Email [48].

- The MIT Reality dataset includes information about the amount of time spent in close proximity between 94 students and staff at a major university. We followed [17] in binarizing the data by assigning a 1 value when the time is greater than 10 minutes and 0 otherwise, producing a 94×94 asymmetric matrix. The entities are grouped into four communities: first year lab students, lab students with more than one year experience, lab staff, and Sloan Business School students.

- The NIPS dataset contains the co-authorships of papers published at the NIPS conference between 1987 and 2012. We follow the process in [17] to obtain a 92×92 symmetric matrix.

- The Soccer dataset consists of the player transfers made between 327 European and North American soccer clubs between seasons 2007/2008 and 2016/2017. The data was obtained from Soccer News[1]. Clubs are grouped into three communities based on the league they play in: Big Four (English Premier League, La Liga, Bundesliga, Serie A); Rest of Europe; North America. The matrix is binarized by assigning a 1 value if at least transfer exists between clubs, creating a 327×327 asymmetrical matrix.

- Lazega's law dataset describes the social network of 71 attorneys in an American law firm between the years 1988 and 1991. The different types of relations measured are binarized to produce a 71×71 matrix.

[1] www.soccernews.com

- Coleman's dataset describes the friendship network of 73 American high-school boys taken half a year apart, creating a $2 \times 73 \times 73$ asymmetric matrix.

- The Temporal EU Email dataset shows the relations of 1005 members of a European research institution as per the emails sent between them. The data was collected over 803 days and split into seven time-steps, giving a $7 \times 1005 \times 1005$ matrix.

3.3.2 Hyperparameter Selection

Applying the MMSB to a neural network framework adds an element of flexibility not offered by the original model. This flexibility is controlled by choosing the model hyperparameters: θ dimension, M, and number of MLN hidden layers. In our work, we set the dimension of θ_l to K for $l \in \{1, 2, ..., L\}$. We explore the effects of community feature size by running models with varying M values on the NIPS, MIT, and Lazega datasets. Each dataset is randomly split into a training set and testing set by masking 20% of the entity relations. Overfitting is prevented by using a held-out validation set to select the best model after 10000 training iterations. The average test AUC scores of five runs on each dataset are provided in Figure 3.3. We notice that low values of M give low test AUC values, suggesting the community features are not large enough to capture the information required to model complex community relations. Increasing M leads to better results up to $M = 4$, after which point the test AUC plateaus in each dataset.

Similarly, the test AUC results for various hidden layer sizes, shown in Table 3.3.2, suggest that simpler models are preferable. We notice models with one hidden layer outperform deeper models on all datasets, with an inverse correlation between number of hidden layers and test AUC. All models reach a training AUC ≥ 0.99 after 10000 iterations and the best model is chosen via a held-out validation set. This suggests that deeper networks are overfitting to the dataset and are less capable of extracting the generalized features required to successfully predict masked entity relations.

For an intuitive grasp of our approach, consider that in matrices $\mathbf{A}$ and $\mathbf{W}$ each community feature vector is represented in M dimensional Euclidean space, $\mathbb{R}^M$. We use the inner product between community features, $\mathbf{a}$ and $\mathbf{w}^T$, passed through a logistic sigmoid operation to represent community relations. The angle between two community features is the cosine similarity between their two vectors. The vector length of a community feature can be thought of as the community's influence since longer vectors push community relations closer to 0 or 1. Therefore even if we mask a community relation, we can use the cosine similarity and vector length to infer the masked community's features relying solely on the unmasked community features. To demonstrate this, we design a synthetic dataset by randomly placing two dimensional community sender and receiver features equidistant on the unit circle. By using such a dataset, we can ensure that the true community features in $\mathbf{A}$ and $\mathbf{W}$ can be represented in a two dimensional feature space.

We compare the performance of the unfactorized community relations matrix $\mathbf{B}$ to our factorized community relations matrix $\mathbf{AW}$ by truncating the static model such that community memberships are fixed, performing inference on the community relations matrices only. The truncated model inputs, $\mathbf{z}_{i\rightarrow j}$ and $\mathbf{z}_{i\leftarrow j}$, are obtained from the ground truth community memberships of the Synthetic, MIT, and Soccer datasets. For each dataset, we mask entity relations from community p to community q and train two truncated models, one for $\mathbf{B}$ and one for $\mathbf{AW}$. We repeat this process K^2 times, masking and training all community relations separately. Both models report a mean training AUC of 0.7316, 0.6994, and 0.6993 for the Synthetic, MIT, and Soccer datasets, respectively.

We obtain the true community relations from community p to community q by calculating the proportion of entity relations present from C_p to C_q:

$$\mathbf{B}_{pq}^{true} = \frac{\sum_{i\in C_p}\sum_{j\in C_q} x_{ij}}{|C_p||C_q|}$$

Where $|C_p|$ and $|C_q|$ is the number of entities that belong to communities p and q, respectively.

We compare the inferred value of the missing community relation, $\mathbf{a}_p\mathbf{w}_q^T$,

Dataset	1 Hidden	2 Hidden	3 Hidden
NIPS	.9660	.9405	.9322
MIT	.9040	.8846	.8530
Lazega	.8550	.8361	.7840

Table 3.1: Mean test AUC for DDBN with various number of hidden layers for NIPS, MIT, and Lazega datasets.

to two prior assumptions one could make when inferring $\mathbf{B}_{pq}$: average and symmetry. The average prior assumes the value of $\mathbf{B}_{pq}$ to be the average of all known non-diagonal outgoing relations. The symmetric prior assumes that relations between communities are symmetric, $\mathbf{B}_{pq} = \mathbf{B}_{qp}$. Note that the symmetric prior can only be applied to off-diagonal community relations.

The mean and standard deviation of absolute errors between inferred community relations and their true values are reported in Table 3.3.3. We split our results into on-diagonal and off-diagonal representing intra and inter community relations, respectively. On the Synthetic dataset, our model outperforms both priors, thereby proving the intuition behind our factorized approach. The symmetric prior outperforms our factorized model on off-diagonal relations in MIT and Soccer which can be explained by a high degree of symmetry in these datasets. Our model outperforms the average prior in all cases.

3.3.3 Static Model Evaluation

We evaluate the performance of our static model on the NIPS, MIT, and Lazega datasets. Each dataset is randomly divided into training and testing sets comprising 80% and 20% of the data, respectively. The training set is further split into a validation subset, the size of which is dataset dependant with smaller datasets receiving a larger fraction of training relations for validation. We first perform a hyperparameter exploration on our static model to find the best performing K and mini-batch size. We train our static model on each dataset for 10000 iterations and use the AUC of a held-out validation set to select the best model and prevent overfitting. Our model is trained five times to account for different results due to random weight initializations. The test

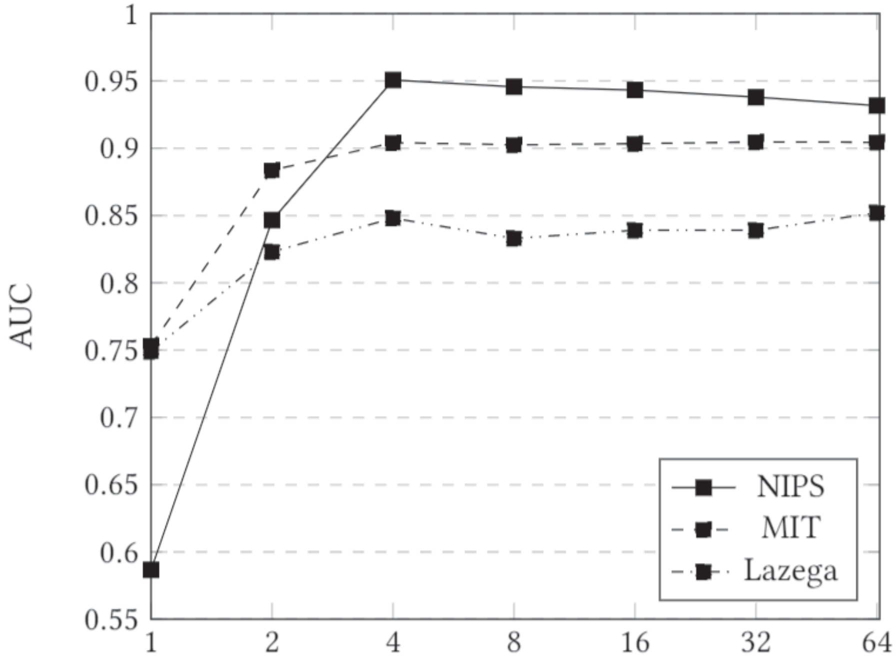


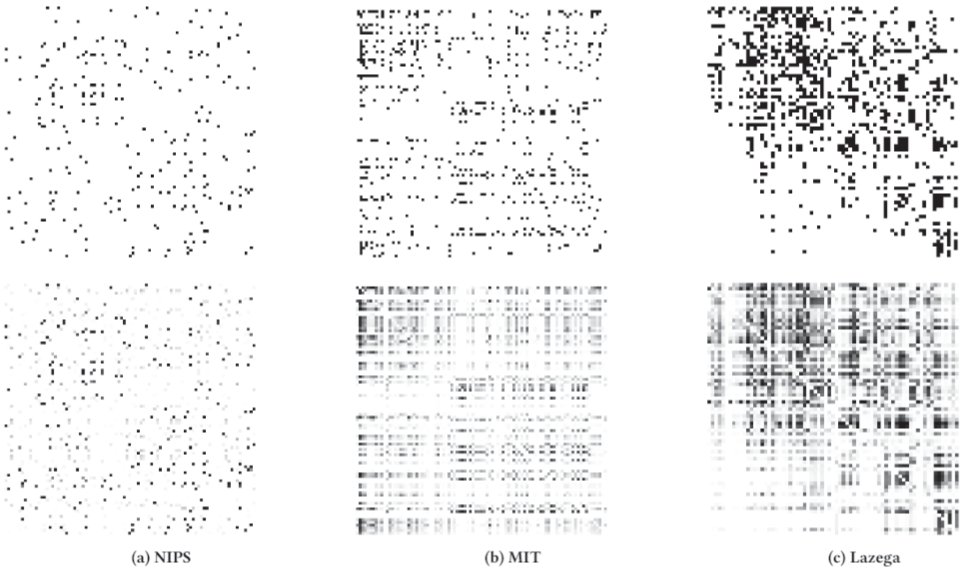
Figure 3.3: Comparison of mean test AUC for various M values on the NIPS, MIT, and Lazega datasets

Dataset	Avg. Prior	Sym. Prior	**AW**
Synth. Off-Diagonal	.2114	.2594	**.0206**
	±.1162	±.1529	±**.0172**
MIT Off-Diagonal	.0270	**.0112**	.0227
	±.0146	±**.0082**	±.0211
Soccer Off-Diagonal	.0105	**.0070**	.0090
	±.0094	±**.0039**	±.0095
Synth. On-Diagonal	.2106	–	**.0095**
	±.1286	–	±**.0059**
MIT On-Diagonal	.0567	–	**.0486**
	±.0672	–	±**.0639**
Soccer On-Diagonal	.0861	–	**.0483**
	±.0498	–	±**.0662**

Table 3.2: The absolute error (mean ± standard deviation) of **AW** and two priors on **B**.

Model	NIPS	MIT	Lazega
IRM	.8901	.8261	.7056
	±.0162	±.0047	±.0167
LFRM	.9348	.8529	.8170
	±.1667	±.0179	±.0197
MMSB	.9524	.8561	.7989
	±.0215	±.0176	±.0102
iMMM	.9574	.8617	.8074
	±.0155	±.0124	±.0141
NMDR	−	.8569	.8285
	−	±.0138	±.0114
cMMSB	.9581	.8794	.8273
	±.0153	±.0159	±.0148
DDBN	**.9660**	**.9040**	**.8550**
	±.0064	**±.0055**	**±.0054**

Table 3.3: Static model performance (test AUC mean ± standard deviation) on the link prediction task for the NIPS, MIT, and Lazega datasets.



(a) NIPS

(b) MIT

(c) Lazega

Figure 3.4: Adjacency matrix heatmaps of the ground truth relations (top) and DDBN learned relations (bottom) for three static datasets. Darker colours indicate a higher probability of a relation between entities.

AUC means and standard deviations are reported in Table 3.3.3.

We compare our static DDBN with benchmarks obtained in [17] on the following models: IRM, LFRM, MMSB, *infinite mixed membership model* (iMMM) [46], *nonparametric metadata dependent relational model* (NMDR) [39], and the better performing variant of cMMSB for each dataset. Our model reliably outperforms these models on the tested datasets. We notice that the difference in performance is greater on the asymmetric and denser datasets. Furthermore, we notice lower standard deviations than the other models, suggesting DDBN is less sensitive to weight initializations and stochasticity in the training process.

Figure 3.4 shows the heatmaps of the ground truth and DDBN predictions for the three datasets. The heatmaps illustrate the network adjacency matrix with darker colours indicating a higher probability of a relationship between entities.

3.3.4 Dynamic Model Evaluation

We follow a similar evaluation procedure for our dynamic model as we did for our static model, this time using the dynamic Coleman and Email datasets. Before applying the dynamic model, pretraining is performed on the static model. In this step, the static model is trained on all training data, regardless of time-step, until a plateau in validation AUC, after which the dynamic component is added to the pretrained model and the full DDBN is trained. This is done to decrease the amount of training steps performed on the dynamic model and therefore decrease training time. As before, each dataset is trained five times. The mean Test AUC results are provided in Table 3.3.4.

We compare DDBN with structure and embedding based approaches using baselines obtained from independent runs of implementations provided in [90] of the following models: Common Neighbour, Jaccard Coefficient, Adamic/Adar [1], MNE, DeepWalk, and *principled multilayer network embedding* (PMNE) [55]. In addition, we compare to tensor factorization method BPTF, using the implementation provided in [71]. Overall, our model outperforms structure and embedding based approaches and is competitive with

Model	Coleman	Email
Common Neighbour	.8794 ±..0210	.9150 ±.0029
Jaccard Coefficient	.8821 ±.0196	.9057 ±.0027
Adamic/Adar	.8823 ±.0204	.9186 ±.0029
MNE	.8990 ±.0203	.8816 ±.0045
DeepWalk	**.9107** ±**.0221**	.7605 ±.0051
PMNE	.9085 ±.0119	.7598 ±.0062
BPTF	.8895 .0246	**.9592** **.0149**
DDBN	.8920 ±.0067	.9481 ±.0026

Table 3.4: Dynamic model performance (test AUC mean ± standard deviation) on the link prediction task for the Coleman and Email datasets.

BPTF. We notice that, like its static counterpart, DDBN is more consistent in its results, as can be seen in the lower test AUC standard deviations.

3.4 Conclusion

In this paper we described the problem of unobserved community relations in latent community models and proposed a solution by factorizing the community relations matrix. To this end, we introduced the *deep dynamic mixed membership stochastic blockmodel based network* (DDBN), an extension of the MMSB applied to the deep neural network setting. The main contributions of this model are summarized as follows:

- The community relations matrix, **B**, is factorized into two feature matrices **A** and **W** similar to the approach used in Collaborative Filtering models.

- We apply the MMSB to a deep neural network, enabling the model to capture more complex latent interactions between entities. Inference via

Gibbs sampling is replaced with stochastic gradient descent.

- The MMSB is extended to the dynamic setting through an LSTM which models the changes in latent features across time.

Empirical results show that this approach outperforms the unfactorized community relations matrix of the MMSB. Our model achieves comparable results on the link prediction task to state-of-the-art models on both static and dynamic real world datasets.

Chapter 4

Fragmentation Coagulation Based Mixed Membership Stochastic Blockmodel

The Mixed-Membership Stochastic Blockmodel (MMSB) is proposed as one of the state-of-the-art Bayesian relational methods suitable for learning the complex hidden structure underlying the network data. However, the current formulation of MMSB suffers from the following two issues: (1), the prior information (e.g. entities' community structural information) can not be well embedded in the modelling; (2), community evolution can not be well described in the literature. Therefore, we propose a non-parametric fragmentation coagulation based Mixed Membership Stochastic Blockmodel (fcMMSB). Our model performs entity-based clustering to capture the community information for entities and linkage-based clustering to derive the group information for links simultaneously. Besides, the proposed model infers the network structure and models community evolution, manifested by appearances and disappearances of communities, using the discrete fragmentation coagulation process (DFCP). By integrating the community structure with the group compatibility matrix we derive a generalized version of MMSB. An efficient Gibbs sampling scheme with Polya Gamma (PG) approach is implemented for posterior inference. We validate our model on synthetic and real world data.

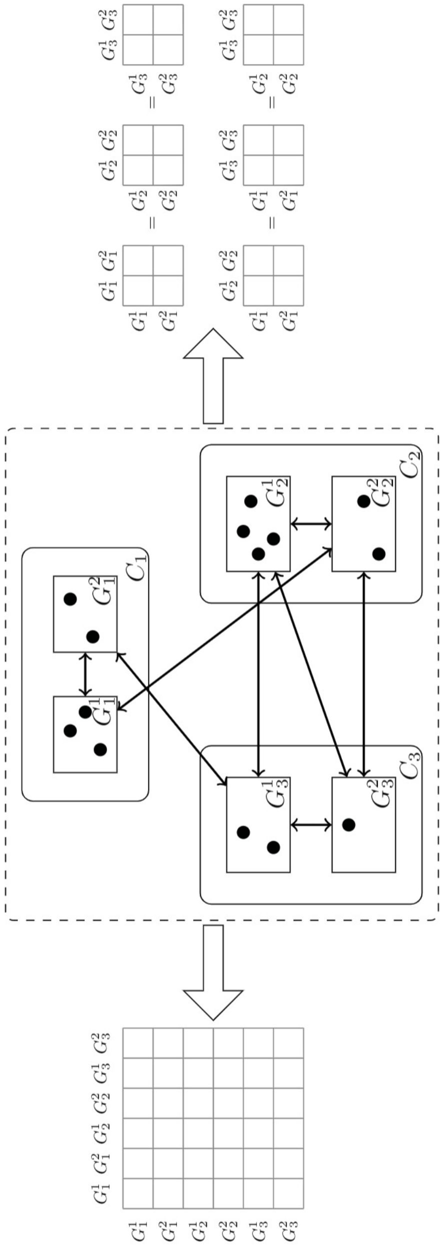


Figure 4.1: An example to illustrate the intuition of the proposed model. Each community (C_1, C_2 and C_3) consists of two groups G^1 and G^2. Entities within each group are represented by black dots. Four types of interactions are considered: within/across groups and within/across communities. In MMSB, a 6×6 compatibility matrix can be used (left part). In our model, it is represented by 2 compatibility matrices: one representing the group relations within communities and another representing the group relations across communities. (right part)

4.1 Introduction

Analysis of complex networks is an important research topic leading to a variety of useful applications. To this end, many interesting and promising approaches have been proposed to address various challenges in investigating these complex networks. The Mixed-Membership Stochastic Blockmodel (MMSB) [3] is one such state-of-the-art model in using Bayesian methods to discover meaningful underlying hidden structure. In general, MMSB assumes each entity in the network has a mixed-membership distribution over the groups. To generate the link between two entities, each entity would sample a belonging group from its mixed-membership distribution. The compatibility value between these two sampled groups would then determine the probability of generating this link.

MMSB has garnered considerable interest in recent years, however, it is not good at embedding certain prior information such as, for instance, the entities' community structure. When the entities in the network are assumed to have a mixed-membership distribution over the groups, the entity itself would belong to only one community. That is to say, we should consider two types of clustering in MMSB: entity-based clustering (i.e. communities for entities) and linkage-based clustering (i.e. groups for links.)

For example, each footballer can play multiple positions (groups) in one match while only belonging to one team (community). This situation is quite common in the real world. Besides, consider the more general example in Figure 4.1 where there are three communities $\{C_i | i \in 1, 2, 3\}$ in the network, each composed of two groups (G_i^1, G_i^2). If we use a 6×6 compatibility matrix, this will hinder interpretability because entities that should belong to groups in the same community may belong to groups in different communities. Under this setting, the MMSB can't not infer any community information about entities. Moreover, the size of the compatibility matrix is bigger than the true one (or the proposed one in Figure 4.1.) which may lead to an overfitting problem.

Furthermore, another issue will also be prominent under the dynamic setting. Recall that with respect to temporal dynamics, most of MMSB-based temporal models focus on correlation among groups in the adjacent time slice. However, the size of their compatibility matrices is same across time which leads to another shortcoming. Consider, for instance, a simple case where there is a complex network with just 2 time slices. At time slice 1, there is one community that consists of 4 groups. It is reasonable to use MMSB with a 4×4 compatibility matrix to represent it. However, at time slice 2, the community splits into two communities. Each community still consists of 4 groups but the entities originally in the same group may have different relations based on the community they belong to. Thus a compatibility matrix of size 8×8 is more suitable at time slice 2. This causes a problem when selecting the compatibility matrix size in the MMSB. Choosing the 4×4 matrix will lead to an underfitting problem while choosing the 8×8 one will lead to an overfitting problem.

In this work, we focus on the following problems:

- In a complex network, we should consider two types of clustering: entity-based clustering (communities for entities); and linkage-based clustering (groups for links). MMSB-based models only adapt the second one in both static and dynamic setting and this will hinder community interpretability.

- Community evolution exists in complex networks across time. MMSB-based models are not able to capture these changes by merely adjusting the size of the compatibility matrix as they use a fixed size compatibility matrix across time.

To handle these two problems, we propose the fragmentation coagulation based Mixed Membership Stochastic Blockmodel (fcMMSB).

To enrich the structure of MMSB, we introduce a community level to MMSB in which the Chinese restaurant process (CRP) is used to partition entities. Due to the nonparametric property of CPR, the number of communities doesn't need to be specified and this makes the model more flexible. For entities in the same community, MMSB is carried out independently to enable each entity to hold multiple groups.

To distinguish the group relations within/across communities, we make use of two compatibility matrices, one for modeling relations between groups in the same community and one for modeling relations between groups in different communities. Specifically, we introduce an across community adjustment parameter which acts as a modifier on the intra group relations across communities so that intra group relations are different if the groups belong to different communities.

Furthermore, to handle the issue in the dynamic setting, we incorporate the discrete fragmentation coagulation process (DFCP) [15], [56] to model the community evolution across time. This allows us to release the limitation of the fixed size compatibility matrix in MMSB across time. The reason is that DFCP can automatically learn the number of communities at each time slice. Also, the changes in the number of communities would influence the entities' group membership. Therefore, this will influence the size of compatibility matrix implicitly. Besides, DFCP helps to model situations such as community splitting and merging while also generalizing MMSB such that when there is only one community in the network, it just turns back to the vanilla MMSB. With this approach, communities can merge into super communities or split into small communities.

The remainder of the chapter is organized as follows. In Section 2, the preliminaries are introduced. We formulate our model, and describe the generative procedures in detail in Section 3. In Section 4, we provide the inference procedure via a Polya-Gamma (PG) approach whereas in Section 5, we evaluate our method on both synthetic and real datasets. Finally, we conclude the chapter in Section 6.

4.2 Model Formulation

In fcMMSB, our task is to do link prediction for the unobserved entity interactions, based on the observed ones. We focus on binary-valued interaction with

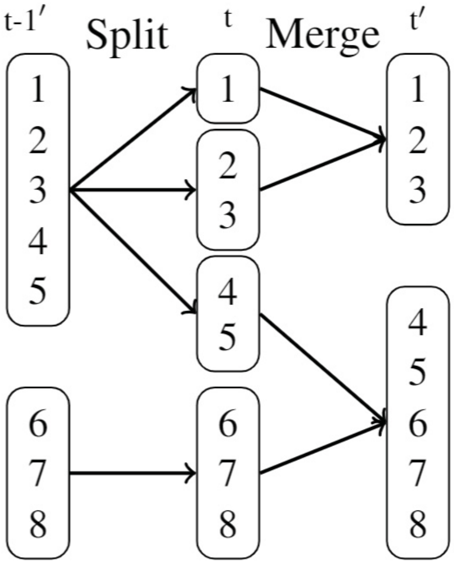


Figure 4.2: Visualization of fragmentation and coagulation processes in fcMMSB. For example, the community of $\{1, 2, 3, 4, 5\}$ at time $t - 1'$ will first be split into 3 small sub-communities $\{1\}, \{2, 3\}, \{4, 5\}$ and then be re-clustered into communities at time t'.

a total number of N entities at T time slices. Formally, these interactions can be defined as a binary 3-d tensor $\mathbf{X} \in \{0, 1\}^T_{N\times N}$, where $x^t_{ij} = 1$ represents a directed interaction between entity u_i and entity u_j at time slice t, and $x^t_{ij} = 0$ represents no interaction. Other format of the observed interactions is possible by considering different forms of the likelihood functions.

4.2.1 Modelling Community Evolution Using DFCP

In our model, each entity (individual) is associated with a community, so community evolution influences relations between entities. Consider, for example, a scenario where corporations are communities, the branches within these corporations (IT, accounting, etc.) are groups, and the network models relations between employees. In the case of a corporate merger, the interactions between employees in the same branches of the merging corporations will increase. In general, we can categorize community evolution into four types: appearance, disappearance, split, and merge. We use fragmentation and coagulation to depict all four types of changes such that coagulation and fragmentation correspond to merging and splitting, respectively. Community appearance and disappearance can be viewed as extensions of community splits and merges. Since communities evolve, it is hard to know the number of communities a priori, thus our model infers the number of communities using non-parametric Bayes.

We adopt the DFCP framework to implement these two operations. DFCP is a non-parametric dynamic clustering process where clusters are first split (fragmentation) and then merged (coagulation). DFCP performs the fragmentation and coagulation processes alternately. To describe the procedures of fragmentation and coagulation, we define a set of disjoint non-empty subsets, $\nu^t = \{\chi_1^t, ..., \chi_r^t\}$ where χ_h^t is a latent community h at t and r is the number of communities at time t. Furthermore, each subset χ_h^t consists of disjoint entities u_i in the network. Figure 4.2 provides the visualization of fragmentation and coagulation processes. In our model, we process fragmentation and coagulation at times $t-1'$ and t, respectively. At time $t-1'$, the fragmentation process partitions each community $\chi_h^{t-1'}$ from $\nu^{t-1'}$ while at time t the obtained partitions are coagulated into a new set of communities $\nu^{t'} = \{\chi_1^{t'}, ..., \chi_r^{t'}\}$.

Now, we provide the generative process for communities using DFCP. To sample community indicator z_i^t for each entity u_i where $i \in \{1, ..., N\}$, we start an initialization with CRP at $t = 0$ as:

$\text{Init}({z_i}^t) : \text{p}({z_i}^0 = h|\mathbf{z}_{-i}^0)$

$$= \begin{cases} |\chi_h^0|/(N+\zeta-1) & \text{if } \chi_h^0 \in \nu_{-i}^0 \\ \zeta/(N+\zeta-1) & \text{if } \chi_h^0 = \emptyset \end{cases}$$

where $\mathbf{z}_{-i}^0$ is the community indicator for all entities excluding entity u_i, ζ is concentration parameter, ν_{-i}^0 is the set ν^0 excluding u_i, $|\chi_h^0|$ is the number of entities in χ_h^0 and $\emptyset$ is a new community at $t = 0$.

In the fragmentation part, each community splits into small communities and executes a CRP partition independently. The fragmentation process at $t \neq 0$ is summarized as:

$\text{Frag}(z_i{}^t) : \text{p}(z_i^t = h|\nu_{-i}^{t-1'}, \nu_{-i}^t, z_i^{t-1'} = q) =$

$$\begin{cases} |\chi_h^t|/(|\chi_q^{t-1'}| + \zeta - 1) & \text{if } \chi_q^{t-1'} \in \nu_{-i}^{t-1'}, \chi_h^t \in \nu_{-i}^t \\ \zeta/(|\chi_q^{t-1'}| + \zeta - 1) & \text{if } \chi_q^{t-1'} \in \nu_{-i}^{t-1'}, \chi_h^t = \emptyset \\ 1 & \text{if } \chi_q^{t-1'} = \emptyset, \chi_h^t = \emptyset \\ 0 & \text{otherwise} \end{cases}$$

We note that all the elements in χ_h^t also belong to $\chi_q^{t-1'}$.

In the coagulation part, we execute a CRP partition on the set of communities. The coagulation process at t' is summarized as:

$\text{Coal}(z_i{}^{t'}) : \text{p}(z_i{}^{t'} = e|\nu_{-i}^{t'}, \nu^t, z_i{}^t = h) =$

$$\begin{cases} 1 & \text{if } \chi_e^{t'} \in \nu_{-i}^{t'}, \chi_h^t \in \nu_{-i}^t \\ |\Omega|/(|\nu^t| + \eta - 1) & \text{if } \chi_e^{t'} \in \nu_{-i}^{t'}, \chi_h^t = \emptyset \\ \eta/(|\nu^t| + \eta - 1) & \text{if } \chi_e^{t'} = \emptyset, \chi_h^t = \emptyset \\ 0 & \text{otherwise} \end{cases}$$

where η is the concentration parameter for the coagulation process and Ω represents the communities at t which belong to the community set with index e at time t'. $= \{\chi_v^t | \chi_v^t \subseteq \chi_e^{t'}\}$.

4.2.2 Generating Relations

In reality, it is common that an entity plays roles in multiple groups. For example, a doctor may be the supervisor of a nurse and the subordinate of the hospital director. Therefore, we induce MMSB to each entity at the group level by imposing a mixed membership vector θ_i^t on each entity u_i at a time slice t. (θ_i^t is a membership of entity u_i over K groups where $\sum_k \theta_i^{t,k} = 1$). For each pair of entities u_i and u_j, we sample group indicators $g_{i\to j}^t, g_{i\leftarrow j}^t$ from Multinomial(θ_i^t) and Multinomial(θ_j^t). The arrow in $g_{i\to j}^t$ and $g_{i\leftarrow j}^t$ indicates the sender (from u_i to u_j) and the receiver (from u_j to u_i), respectively.

Now, we construct a compatibility matrix to predict entity relations x_{ij}^t based on the community and group indicators. Imagine that there are several communities consisting of groups inside a complex network. It is quite common that the inner structure (group relations) of each community is similar. For example, each company has sales and marketing departments. Besides,

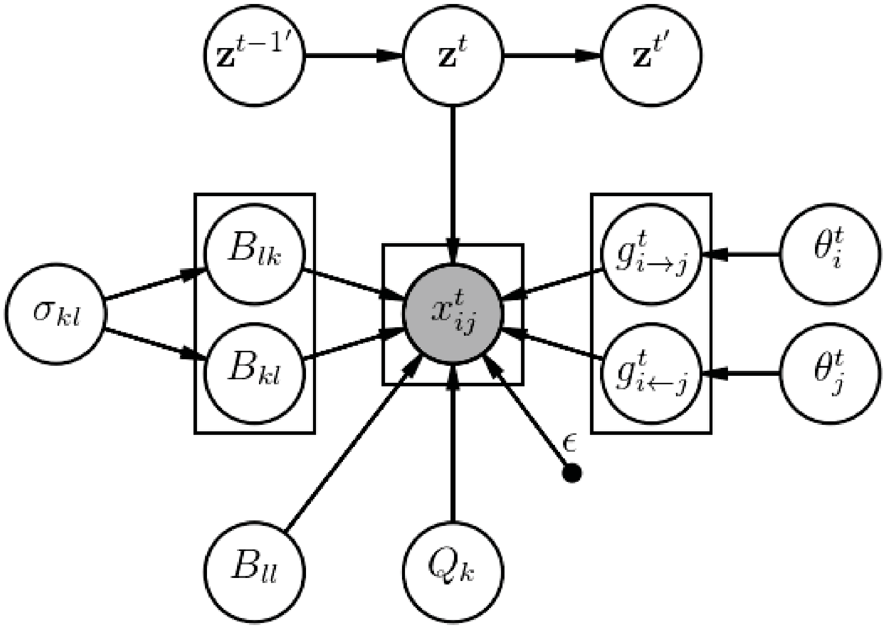


Figure 4.3: Graphical model of fcMMSB. Hyperparameters are not shown. $\cdot^t$ and $\cdot^{t'}$ denote the time index of fragmentation and coagulation process respectively. Notation: $\mathbf{z}^t = \{z^t_i | i \in \{1, ..., N\}\}$.

groups within the community are more likely to have tighter interactions than ones across communities. Moreover, across community, groups with similar functionality are more probable to have interactions. Therefore two assumptions are made to construct these relations. First, group pair relations within communities are consistent. We use a compatibility matrix, $\mathbf{B}$, to model all within community group relations. Second, interactions between entities from the same group but in different communities may be different from ones in the same group and community. To account for this we add a K-array across community adjustment parameter $\mathbf{Q}$ to on-diagonal values of the $\mathbf{B}$. This provides a flexible way to model the differences of within-group entity relations based on whether the entities are in the same community. Furthermore, we set the value of relations between entities that do not share community nor group to a small value, ϵ. For each pair of entities u_i and u_j, we sample x^t_{ij}

from Bernoulli($\frac{1}{1+\exp{(-y_{ij}^t)}}$) where

$$y_{ij}^t = \begin{cases} B_{lk} & \text{if } z_i^t = z_j^t,\, g_{i\rightarrow j}^t = l,\, g_{i\leftarrow j}^t = k \\ B_{kk} + Q_k & \text{if } z_i^t \neq z_j^t,\, g_{i\rightarrow j}^t = g_{i\leftarrow j}^t = k \\ \epsilon & \text{otherwise} \end{cases}$$

Group pairs are always correlated in the real world. For example, employee-employer relations can be unidirectional while employee-employee may be bidirectional. We are interested in the correlation of group pairs so the Inverse-Wishart prior is imposed on the variance σ_{kl} of the normal distribution of B_{lk} and B_{kl}. Finally, we share the group-level compatibility matrix $\mathbf{B}$ and adjustment parameter K-ary Q across time due to the data sparsity.

In summary, the fcMMSB generative model is as follows:

- To generate compatibility matrix $\mathbf{B}$
  - sample $\sigma_{kl} \sim \text{Invwishart}(\upsilon, \varrho)$
  - sample $(B_{lk}, B_{kl}) \sim \mathcal{N}(\mu_{\text{kl}}, \sigma_{\text{kl}})$
  - sample $B_{kk} \sim \mathcal{N}(\mu_B, \sigma_B)$
- For each across community adjustment parameter Q_k
  - sample $Q_k \sim \mathcal{N}(\mu_{\text{Q}}, \sigma_{\text{Q}})$
- For each mixed membership of entity u_i
  - sample $\theta_i^t \sim \text{Dirichlet}(\alpha)$
- For each community indicator z_i^t
  - sample $z_i^0 \sim \text{Init}({z_i}^0)$
  - sample $z_i^t \sim \text{Frag}({z_i}^t)$
  - sample $z_i^{t'} \sim \text{Coal}({z_i}^{t'})$
- To generate each directed relations x_{ij}^t
  - sample sender group $g_{i\rightarrow j}^t \sim \text{Multinomial}(\theta_i^t)$

- sample receiver group $g^t_{i\leftarrow j} \sim \text{Multinomial}(\theta^t_j)$
- sample $x^t_{ij} \sim \text{Bernoulli}(\frac{1}{1+e^{-y^t_{ij}}})$

We give the graphical model of fcMMSB in Figure 4.3.

4.3 Inference

Our model is intractable for exact inference, instead we derive a Gibbs sampling scheme for posterior inference. The target is to predict the unobserved relations between entities by inferring parameters $\mathbf{z}, \theta, \mathbf{B}, \mathbf{Q}, \mathbf{g}$ and σ. The parameter in bold represents its total set.

The joint distribution $\mathrm{p}(\mathbf{x}, \mathbf{z}, \theta, \mathrm{B}, \mathbf{Q}, \mathbf{g}|\epsilon, \alpha, \zeta, \eta)$ can be expressed as:

$$\begin{aligned}
&\prod_{i,j,t} \mathrm{p}(x^t_{ij}|z^t_i, z^t_j, Q_{g^t_{i\to j}}, \mathrm{B}_{g^t_{i\to j} g^t_{i\leftarrow j}}, \epsilon) \prod_i \mathrm{Init}(z^0_i) \\
&\prod_{i,t} \mathrm{Frag}(z^t_i)\mathrm{Coal}(z^{t'}_i) \prod_k \mathrm{p}(Q_k|\mu_Q, \sigma_Q)\mathrm{p}(\mathrm{B}_{kk}|\mu_\mathrm{B}, \sigma_\mathrm{B}) \\
&\prod_{i,j,t} \mathrm{p}(g^t_{i\to j}|\theta^t_i) \prod_{l,k,l\neq k} \mathrm{p}(\mathrm{B}_{lk}, \mathrm{B}_{kl}|\mu_{kl}, \sigma_{kl}) \prod_{i,t} \mathrm{p}(\theta^t_i|\alpha)
\end{aligned}$$

4.3.1 Sampling $B_{lk}, B_{kl} (l \neq k)$ Using Polya-Gamma

For simplicity, the (B_{lk}, B_{kl}) pair is denoted as a vector $\hat{\mathbf{B}}$ in this section. The Polya-Gamma (PG) data augmentation is implemented for $\hat{\mathbf{B}}$. Following [65], $\frac{(e^\phi)^m}{(1+e^\phi)^n}$ can be expressed as $2^{-n}e^{\kappa\phi}\mathbb{E}\{e^{-w\phi^2/2}\}$ with a PG variable $\omega \sim \mathrm{PG}(n, 0)$, where $\kappa = m - n/2$. Furthermore, with conditional distribution $p(w|\phi)$, we have $\omega|\phi \sim \mathrm{PG}(n, \phi)$. Assuming that the prior of ϕ follows $\mathcal{N}(\mu, \sigma)$ with likelihood $\frac{(e^\phi)^m}{(1+e^\phi)^n}$, the posterior of ϕ is a Gaussian distribution. Therefore, the true posterior of ϕ can be derived by updating ϕ and ω alternately.

In our model, $\hat{\mathbf{B}}$ is updated via PG approach by alternately sampling $\hat{\mathbf{B}}, \omega_{lk}, \omega_{kl}$:

$$\hat{\mathbf{B}}|- \sim \mathcal{N}(\mu^*, \sigma^*)$$

$$\omega_{lk} \sim \mathrm{PG}(n_{lk}, B_{lk}), \omega_{kl} \sim \mathrm{PG}(n_{kl}, B_{kl})$$

where

$$\mu^* = \sigma^*(\kappa + \sigma_{kl}\mu_{kl})$$
$$\sigma^* = (\Omega + {\sigma_{kl}}^{-1})^{-1}$$

$\kappa = (\kappa_{lk}, \kappa_{kl})$. Ω is a diagonal matrix of ω_{lk} and ω_{kl}. $\kappa_{lk} = n^1_{lk} - n_{lk}/2$. Here $n_{lk} = \sum_{t,i,j} \mathbb{I}[g^t_{i\to j} = l] \cdot \mathbb{I}[g^t_{i\leftarrow j} = k] \cdot \mathbb{I}[z^t_i = z^t_j]$ and $n^1_{lk} = \sum_{t,i,j} \mathbb{I}[g^t_{i\to j} = l] \cdot \mathbb{I}[g^t_{i\leftarrow j} = k] \cdot \mathbb{I}[z^t_i = z^t_j] \cdot \mathbb{I}[x^t_{ij} = 1]$ where $\mathbb{I}$ is an indicator function. As the sampling scheme of B_{ll} and Q_l is similar with $\hat{\mathbf{B}}$, we omit the procedure here.

4.3.2 Sampling $g^t_{i\to j}$

Collapsed Gibbs sampling is used on $g^t_{i\to j}$ by marginalizing over θ^t_i. The posterior of $g^t_{i\to j}$ can be expressed as:

$$\mathrm{p}(g^t_{i\to j} = k|-) \propto \frac{[e^{y^t_{ij}}]^{\mathbb{I}[x^t_{ij}=1]}}{1 + e^{y^t_{ij}}} \frac{n_k^{i\neg j}(t) + \alpha_k}{\sum_k n_k^{i\neg j}(t) + \alpha_k}$$

where $n_k^{i\neg j}(t) = \sum_{l,l\neq j} \mathbb{I}[g^t_{i\to l} = k]$.

4.3.3 Sampling z

The prior of latent communities sequence $\mathbf{z}_i$ is:

$$\mathrm{p}_{\text{prior}}(\mathbf{z}_i) = \mathrm{Init}({z_i}^0) \cdot \mathrm{Coal}({z_i}^{0'}) \cdot \ldots \cdot \mathrm{Frag}({z_i}^{T-1})$$

so the posterior of $\mathbf{z}_i$ can be described as:

$$\begin{aligned}\mathrm{p}(\mathbf{z}_i|-) &\propto \mathrm{p}(\mathbf{x}_{i\cdot}, \mathbf{x}_{\cdot i}|\mathbf{z}, \theta, \mathrm{B}, \mathrm{Q}, \mathbf{g}, \epsilon) \cdot \mathrm{p}_{\text{prior}}(\mathbf{z}_i) \\ &= \prod_{j,t} \frac{[e^{y^t_{ij}}]^{\mathbb{I}[x^t_{ij}=1]}}{1 + e^{y^t_{ij}}} \cdot \frac{[e^{y^t_{ji}}]^{\mathbb{I}[x^t_{ji}=1]}}{1 + e^{y^t_{ji}}} \cdot \mathrm{p}_{\text{prior}}(\mathbf{z}_i)\end{aligned}$$

where y^t_{ij} follows the previous definition in section 3. For computational simplicity, we use forward-backward algorithm on $\mathrm{p}(z_i^{\mathbf{T}}|-)$. Here $\mathbf{x}_{i\cdot} = \{x^t_{ij}|j \in \{1, ..., N\}, t \in \{0, ..., T-1\}\}, \mathbf{x}_{\cdot i}$ is defined similarly.

4.3.4 Sampling σ_{kl}

As the prior and likelihood of σ_{kl} are a conjugate pair, we give the posterior of σ_{kl} directly.

$$\sigma_{kl}|- \sim \mathrm{Invwishart}(1 + \upsilon, \varrho + (\hat{\mathbf{B}} - \mu_{kl})(\hat{\mathbf{B}} - \mu_{kl})^{\intercal})$$

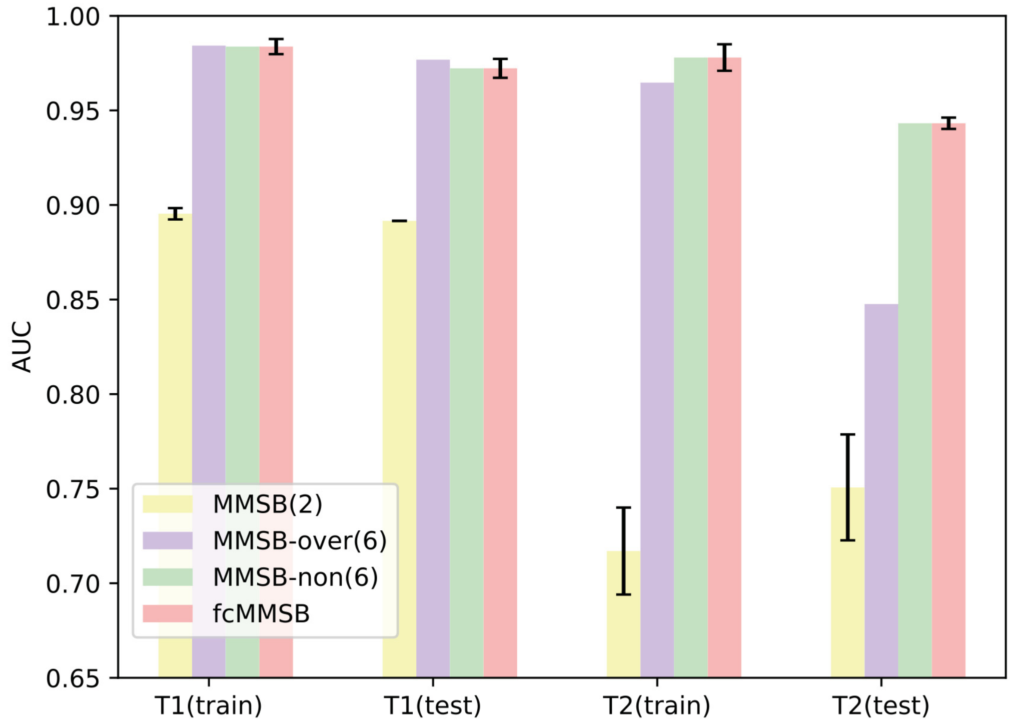


Figure 4.4: AUC comparison on synthetic data.

4.3.5 Prediction

In the previous sections, we derived the samples at each iteration. We would like to use these samples to estimate the unobserved relations. Our prediction target at iteration s, $\hat{x}_{ij}^{t[s]}$, is expressed as $\hat{\theta}_i^{t\intercal} \cdot \bar{\mathbf{B}} \cdot \hat{\theta}_j^t$, where the superscript of $\hat{\theta}_i^{t\intercal}$ is the transpose of the vector. Here each dimension of θ_j^t is $\hat{\theta}_i^{t,k} = \frac{n_k^i(t)+\alpha_k}{\sum_k n_k^i(t)+\alpha_k}$ and $n_k^i(t) = \sum_j \mathbb{I}[g_{i\to j}^t = k]$. Each entry $\bar{\mathbf{B}}_{lk}$ of $\bar{\mathbf{B}}$ is $\frac{1}{1+\exp(-\bar{\mathbf{Y}}_{lk})}$ and $\bar{\mathbf{Y}}_{lk} = \mathbb{I}[z_i^t = z_j^t]\mathbf{B}_{lk} + \mathbb{I}[l = k]\mathbb{I}[z_i^t \neq z_j^t](\mathbf{B}_{lk} + Q_k) + \mathbb{I}[l \neq k]\mathbb{I}[z_i^t \neq z_j^t]\epsilon$.

4.4 Evaluation

4.4.1 Synthetic Data

To demonstrate the problem of the MMSB mentioned in the introduction, we generate a synthetic dataset with $N = 100$ and $T = 2$, the generative process for which is described as follows:

1. Instantiate a network structure of three communities containing two

groups each. For each time slice, generate the mixed membership for 100 entities by sampling the Dirichlet distribution with parameters $[0.8, 0.2]$ or $[0.2, 0.8]$ depending on the group. Set $\mathbf{B}$ to be a 2×2 compatibility matrix with high on-diagonal values and low off-diagonal values.

2. For time slice 1, if both entities belong to the same community perform step 3, otherwise set the entity relation to 0. For time slice 2, if both entities belong to the same community and group perform step 3, otherwise set the entity relation to 0.

3. Generate entity relations using the Bernoulli distribution with parameter $(\theta_i^\intercal \mathbf{B} \theta_j)$ for the relation between u_i and u_j.

For evaluation, we randomly split the data into 2 subsets: 80% for training and 20% for testing. We compare our model with two different MMSB models varying in the number of groups in the compatibility matrix. The train and test AUC results are provided in Figure 4.4. We notice that when the number of groups in MMSB is 2, it is underfitting relative to fcMMSB with 2 groups. When the number of groups in MMSB is 6, there are two possible outcomes: overfitting and not overfitting. The overfitting of the MMSB is demonstrated by the higher train AUC and lower test AUC on time slice 2 compared to our model. Overfitting is not always the outcome, however, and the stochastic nature of the MMSB means that on different runs, the MMSB may achieve similar results to our model, as shown by MMSB-non in Figure 4.4. This demonstrates the problem of choosing the number of groups in the MMSB.

4.4.2 Prediction Relations

To demonstrate the potential of our fcMMSB model, we use five real-world datasets for validation. We use the relation prediction task to validate our model. The area under the ROC (Receiver Operating Characteristic) curve (AUC) is used as a performance metric. Here, we randomly select 80% data for training and leave the 20% for testing. Each experiment is run for five times, and we report the AUC results with their mean and standard deviation values. Five real-world datasets are described as follows:

Model	Coleman	Student net	Mining reality	Hypertext 2009	Infectious
CN	0.881 ± 0.018	0.839 ± 0.019	0.873 ± 0.004	0.776 ± 0.006	0.883 ± 0.014
MMSB	0.880 ± 0.016	0.914 ± 0.011	0.885 ± 0.007	0.867 ± 0.005	0.965 ± 0.001
T-MBM*	0.881 ± 0.005	0.896 ± 0.010	0.861 ± 0.002	0.790 ± 0.004	0.838 ± 0.008
BPTF*	0.908 ± 0.013	0.909 ± 0.021	0.922 ± 0.001	0.874 ± 0.006	0.843 ± 0.011
SVD++*	—	—	0.833 ± 0.006	0.735 ± 0.004	0.614 ± 0.011
DRGPM*	—	0.823 ± 0.014	0.933 ± 0.003	$\mathbf{0.904 \pm 0.008}$	$\mathbf{0.988 \pm 0.000}$
MNE	0.891 ± 0.024	0.940 ± 0.020	0.813 ± 0.004	0.872 ± 0.008	0.900 ± 0.017
DeepWalk	$\mathbf{0.914 \pm 0.018}$	0.910 ± 0.018	0.759 ± 0.004	0.816 ± 0.005	0.910 ± 0.014
fcMMSB	0.908 ± 0.009	$\mathbf{0.954 \pm 0.006}$	$\mathbf{0.935 \pm 0.004}$	0.902 ± 0.001	0.981 ± 0.001

Table 4.1: Model performance: AUC (mean and standard deviation) on the real dataset. Note: * represents a dynamic model.

- The Coleman dataset [10] contains the information about the friendships of boys in an Illinois high-school. It records the three closest friends for each student in the fall of 1957 and spring of 1958. The binarized dataset is a $73 \times 73 \times 2$ asymmetric matrix.

- The Student net dataset [16] describes the relations between students. We binarize the relations at each time slice, leading to a $50 \times 50 \times 3$ asymmetric matrix.

- Mining Reality dataset [13] records contact data of 96 students at the Massachusetts Institute of Technology (MIT) over 9 months in 2004. The dataset is split into 10 time slices, then we set each entity pair value to be 1 at that time slice if they have at least one contact during that time. Thus, it leads to a $96 \times 96 \times 10$ symmetric matrix.

- The Hypertext 2009 dataset [36] records the contact network ACM Hypertext 2009 conference attendees. The relation between two attendees is 1 if they have a face-to-face contact over 20 seconds. We split the dataset into 10 time slices and binarize it, leading to $113 \times 113 \times 10$ symmetric matrix.

- The Infectious dataset [36] describes the face-to-face interactions between people during the exhibition INFECTIOUS: STAY AWAY in 2009 at the Science Gallery in Dublin. Each relation is 1 if those two people had face-to-face contact for at least 20 seconds. We binarize the relations at each time slice, leading to a $410 \times 410 \times 10$ symmetric matrix.

4.4.3 General Performance

We use eight baseline methods for comparison:

- One structure-based model: Common neighbor (CN) [62].

- Five feature or cluster based models: Mixed Membership Stochastic Blockmodel (MMSB) with Gibbs sampling [3], Temporal Tensorial Mixed Membership Stochastic Blockmodel (T-MBM) [76], Bayesian Poisson

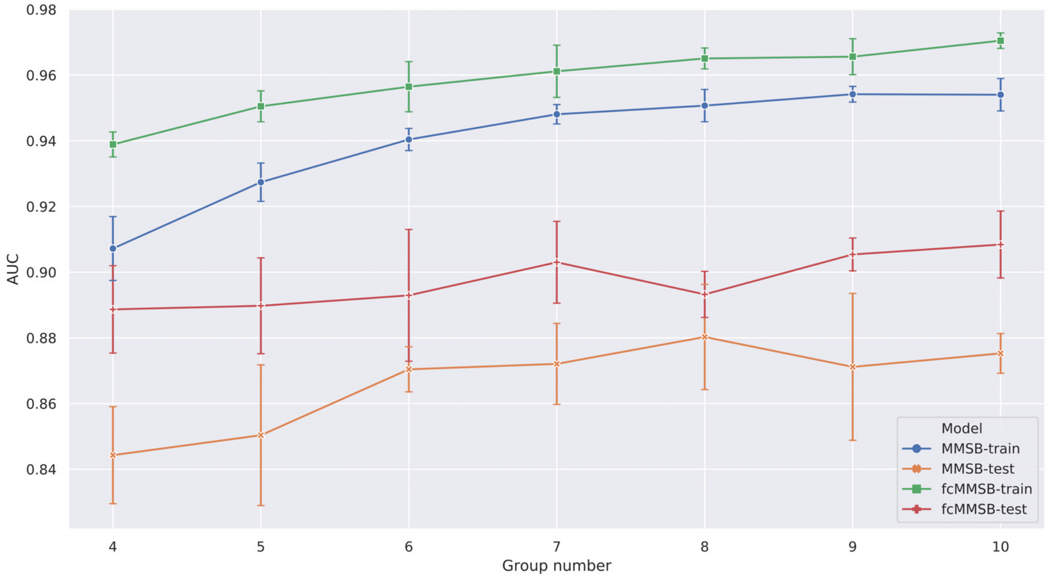


Figure 4.5: Comparison of AUC between MMSB and fcMMSB on the Coleman dataset.

Tensor Factorization (BPTF) [71], Collaborative filtering with temporal dynamics (SVD++) [44] and Dependent relational gamma process model (DRGPM) [86].

- Two embedding based models: Scalable Multiplex Network Embedding (MNE) [90] and DeepWalk [64].

We show the results in Table 4.1. The overall result of fcMMSB is competitive with DRGPM and outperforms the other state-of-the-art models. This may result from fcMMSB, with its flexible structure, being more suitable for long time series datasets in which the number of communities may vary across time. The DRGPM performance on Student net dataset may suffer from the short time sequence of the dataset.

Compared with vanilla MMSB, fcMMSB also shows its advantage on both short and long time series dataset. We compare our model with MMSB by varying the group number parameter on the Coleman dataset in Figure 4.5. fcMMSB achieved better AUC on both train and test sets. When we increased the group number, train AUC on both models increased. Due to the flexible structure of fcMMSB, the margin of train and test AUC between fcMMSB and

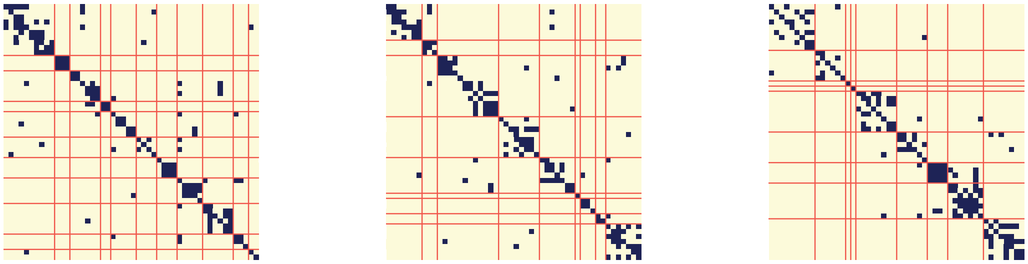


Figure 4.6: Visualization of community clustering on the Student net dataset across time.

MMSB is relatively larger with smaller group numbers. While the train AUC of MMSB is relatively close to that of fcMMSB, the test AUC is lower.

Besides, we compare fcMMSB with vanilla MMSB by looking at the trained compatibility matrix for the Hypertext dataset in Figure 4.8. We see that the MMSB compatibility matrix is similar to the within-community matrix in fcMMSB. However, there is a moderate difference in fcMMSB between the within-community and across communities matrices for the entry (2,2). This shows that the group-pair relation within a community is not same as the one across communities, therefore MMSB with its single compatibility matrix, cannot properly model this network. This is why the fcMMSB is better than MMSB on the Hypertext dataset; its more flexible structure is better at modeling multiple communities. In comparing the compatibility matrices of other datasets, we find this to be the case with other datasets as well. We also observe that the second role of the membership covers the main part for most people due to sparsity which can be interpreted as the inactive role. This interpretation is consistent with the compatibility matrix. In Figure 4.6, we visualize the community clustering result on the Student net dataset. We find the data points are dense along the diagonal. This is consistent with our assumption that the interactions within the community are tighter than the ones across communities. Besides, we find that most entities belong to the same communities across time, even though the community index may change. This shows why DFCP is used in our model since DFCP constructs a temporal dependency for communities across time.

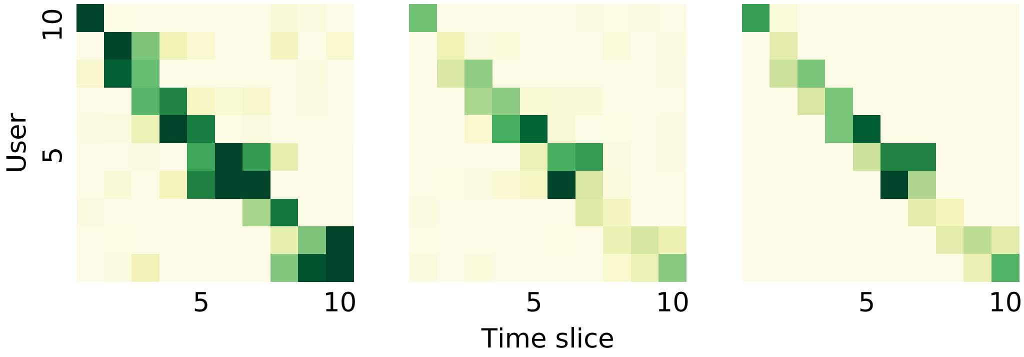


Figure 4.7: Left: User activeness across time. Middle: The recovered number of user interactions. Right: The original number of user interactions.

Furthermore, to show the dynamic of membership in the Infectious dataset, for each time slice, we randomly select one user who is active at that time slice. To show the intensity of user activeness, we define activeness, AC, for each user i at time slice t to be $\mathrm{AC}_i^t = \theta_i^{t\intercal} \cdot \mathbf{B} \cdot \mathbf{1}$. We present the user activeness and the recovered number of users' interactions with the original one in Figure 4.7. It is interesting that the user is active in consecutive time segments. Meanwhile, comparing the user activeness with the original user interactions, it is easy to observe that they have correlations. This shows the membership used for user activeness really reflects the characteristic of the data. Also the recovered number of user interactions is similar with the original one in Figure 4.7. Besides, we find that BPTF got the relatively low AUC compared with the other four datasets. It seems that the tight correlation of features across time inherent in BPTF does not fit this dataset. Overall, fcMMSB is stable in both dense (Coleman, Student net, Mining reality) and sparse (Hypertext, Infectious) datasets.

4.5 Discussion of the limits

Though we use Gibbs sampling and data augmentation via PG approach, the computational complexity is still $O(TKN^2)$, mainly contributed by group indicator parameter $g_{i\to j}^t$. This makes our work hard to apply for large-scale

$$\begin{bmatrix} .997 & .001 & .001 \\ .002 & .001 & .001 \\ .001 & .001 & 996 \end{bmatrix} \qquad \begin{bmatrix} .997 & .002 & .001 \\ .002 & .001 & .001 \\ .001 & .001 & .998 \end{bmatrix} \qquad \begin{bmatrix} .993 & .002 & .001 \\ .002 & .035 & .001 \\ .001 & .001 & .992 \end{bmatrix}$$

Figure 4.8: Left: compatibility matrix in MMSB. Middle: compatibility matrix within community in fcMMSB. Right: compatibility matrix across community in fcMMSB.

datasets, although acceptable on medium datasets.

However, a good sampling scheme for other parameters is necessary. For example, though the compatibility matrix B does not dominate the computational complexity, as the matrix B working as global parameters impact all other local parameters including g, a good sampling scheme for B is essential and it will allow the log-likelihood to converge fast. If we use a common sampling method like Metropolis-Hasting instead, it may lead to slower convergence.

Besides, even though computational complexity is fixed, it is feasible to adjust the order of parameters during the sampling procedure to scale our work up to larger datasets. First, the group indicator parameter **g** is independent across time. This means we can sample parameter **g** at each time slice in a parallel way. Second, during the procedure of each sampling iteration, grouping **g** into N/2 groups will also improve the sampling speed. For example, we can sample $g^t_{i\to j}$ and $g^t_{m\to n}$ simultaneously iff $i \neq m$ and $j \neq n$. These operations will save time during sampling. At this moment, due to the computational complexity $O(TKN^2)$, our work is suitable for small to medium problems.

4.6 Conclusion

In this work, we highlight two problems in MMSB: the structure in MMSB is unable to encapsulate the prior information like the community structure of entities in the static case; and modelling the community evolution using a fixed size compatibility matrix may suffer underfitting/overfitting in the dynamic case. To overcome these two problems, we developed the fragmentation coagulation based Mixed Membership Stochastic Blockmodel (fcMMSB). Specifically, we used CRP for entity-based clustering to capture the community information of entities and MMSB for linkage-based clustering to derive the group information for links simultaneously. Besides, we utilized DFCP to infer the community structure (including the number of communities) among entities and evolution (appearance/disappearance or split/merge). Our model combines a group-level compatibility matrix with a community adjustment parameter to satisfy the four types of entity pair relations: within and across communities and groups. Our model unifies these techniques to derive a generalized MMSB. Furthermore, a PG approach is implemented for an efficient sampling scheme to infer hidden variables. Finally, we demonstrate the fcMMSB outperforms and is competitive with the state-of-the-art methods through experiments on real datasets.

Chapter 5

Reciprocating Interactions Simulation in Continuous Time

Reciprocating interactions are commonly seen in real-world scenarios. The notable approach of [7] proposes to use pairwise Hawkes processes to model this reciprocating interactions. As It assumes the clustering of nodes keeps invariant along the time, this assumption over-simplifies the complexity of the data as nodes may change to different groups along with the time. By using the Fragmentation Coagulation Process, we allow the nodes have continuous nonparametric clustering effect along the time.

5.1 Introduction

Bayesian relational modelling on temporal complex networks is gaining increasing attention as discovering the evolution and structure of complex networks provides the valuable information in politics, business, sports, etc. A lot of works [22], [71] achieved success in these areas. In particular, the temporal Bayesian relational model can be categorized into two types: Poisson matrix factorization model and latent community model. Both type of methods construct the correlation of features to model the dependency across the time. However, due to the Markov property of Bayesian model, they lack the ability to capture the long-term dependency across time. Therefore, the reciprocating interactions modelling studying the influence of the previous interactions on the future ones arouse tremendous interest in recent years. As Hawkes process (HP) models the self-exciting phenomenon, and owns long memory, many recent works rely on HP to address the long-term issue.

The CRP Hawkes [7] can be considered as the first attempt to use HP to model the reciprocating interactions between communities. [53] tried to infer the implicit network underlying the financial activities by HP. [84] embedded each individual in Euclidean space to capture the reciprocity and homophily in the network. [75] introduced the Indian Buffet Hawkes process model which allows the multiple evolving factors from the past events to drive the future event. [58] utilized compound completely random measure (CCRM) to model the base intensity of HP, and derived both temporal reciprocity and community structure.

The CRP Hawkes received considerable attention in recent years. However, the model has two main issues: 1, The model is unable to capture the evolution of community structure across time. For example, members of the Soviet Union falling apart can be considered as a large community splitting into smaller communities. While countries joining the North Atlantic Treaty Organization (NATO) can be considered as the communities merging into a large community. So to capture the community evolution is essential in the real world. 2, the fixed characterization of the HP between groups across time is not suitable. With the evolution of community structure, the interactions between communities will change. So the static characterization of the HP between groups is unable to model the real situation.

To handle these two problems, we propose the Fragmentation Coagulation Hawkess process model (FCHP).

To allow the community structure to be flexible across time, we incorporate Fragmentation Coagulation Process (FCP) [77] to model the community evolution across time. FCP helps to model situations such as community splitting and merging. Also community appearance and disappearance can be viewed as extensions of community splits and merges. So FCP plays a role in enriching the community structure and constructing the dependency of communities across time.

Meanwhile, FCP also provides the solution to allowing the dynamic characterization of the HP between groups. With the process of FCP, the newly community will be generated by the fragmentation or coagulation operation. In bayesian model, each community can be considered as a latent state, which parameterizes the corresponding HP. Compared with [78], which use Gaussian process (GP) to model the intensity of point process, our work also dramatically reduces the computational complexity .

Interestingly, HP also is widely used in change detection task [50], [51]. Our work integrating FCP with HP also brings such a side effect. This is because FCP captures the evolution of community structure, while the community fragmentation or coagulation can be viewed as the consequence resulting from a sudden change (an event happened).

5.2 Preliminary knowledge

Here we assume there is a sequence of interactions $\{t_{ij}^s\}_{s=1}^S$. t_{ij}^s represents an interaction from entity i to entity j happened at time t_s where $i, j \in \{1, ..., N\}$ and $t_s \in [0, T]$.

5.2.1 Fragmentation Coagulation Process

The Fragmentation coagulation process with two parameters ν and ξ is described as follows:

- At time 0, the community indicator of entity i, $z_{i,0}$, conditioned on the community indicators of entities excluding i, $\mathbf{z}_{\text{-}i,0}$, follows the CRP as:

$$z_{i,0}|(\mathbf{z}_{\text{-}i,0}) \propto \begin{cases} |c| & \text{if } z_{i,0} = c \\ \frac{\nu}{\xi} & \text{if } z_{i,0} = \emptyset \end{cases}$$

- Assume that $z_{i,t^{\text{-}}} = c$ is the existing community, at time t there are three cases:

  - The community c splits into two communities: a and b. The probability of choosing one of the communities is:

$$z_{i,t^-}|(\mathbf{z}_{\text{-}i,t}, z_{i,t^-} = c) = \begin{cases} \frac{|a|}{|c|} & \text{if } z_{i,t} = a \\ \frac{|b|}{|c|} & \text{if } z_{i,t} = b \end{cases}$$

  - The community c and another community are merged into a new community c'. The probability of following the community c' is 1.
  - Entity i leaves community c to form a new community with itself ,$z_{i,t} = \emptyset$, at rate of $\frac{\nu}{|c|}$.

- Assume that $z_{i,t^-} = \emptyset$ is the new community, the community will coagulate with another community at rate of ξ.

Here t^- represents an infinitesimal time before t and $|\cdot|$ represents the number of elements in the set.

5.2.2 Hawkes Process

The Hawkes process is an important class of point process with a wide variety of applications. It can be described by its conditional intensity:

$$\lambda(t) = b(t) + \int_0^t \psi(t)\, dN(t)$$

where $b(t)$ is the base intensity, $\psi(t)$ represents the triggering intensity and $N(t)$ is the counting process. When $b(t)$ is a constant and $\psi(t) = 0$, the Hawkes process will be reduced to the homogeneous Poisson process. One typical kernel of triggering intensity $g(t)$ is the exponential kernel: $g(t) = \eta e^{-\zeta t}$, where η is the scaling parameter and ζ is the bandwidth parameter. The kernel can be interpreted as the influence of past events on triggering new events will decay with time.

5.2.3 Modelling reciprocating interactions through Hawkes process

CRP Hawkes [7] presents to be the first work to use Hawkes process to model the reciprocating interactions in the continuous time line. In CRP Hawkes, it first uses Chinese Restaurant Process (CRP) to generate a *fixed* grouping $\boldsymbol{\pi}$ (assume K distinct groups are generated) on all the nodes in the interaction networks. Then, CRP Hawkesgenerates individual Hawkes processes (with intensity function noted as $\lambda_{pq}(t)$) for each group-pair $(p, q) \in \{(k_1, k_2)\}_{k_1,k_2}$ (in total, there are K^2 Hawkes processes). Each Hawkes process would generate group-wise interactions ($N_{pq}(\cdot)$) along the time and these group-wise interactions would be assigned to nodes' interactions ($N_{uv}(\cdot)$) according to the splitting rule of the Poisson process.

More formally, the corresponding generative process can presented as follows:

$$\boldsymbol{\pi} \sim \mathrm{CRP}(\alpha) \tag{5.1}$$

$$\lambda_{pq}(t) = \gamma_{pq} n_p n_q + \beta_{pq} \int_{-\infty}^{t} e^{-\frac{t-s}{w_{pq}}} dN_{pq}(s) \tag{5.2}$$

$$N_{pq}(\cdot) \sim \mathrm{HawkesProcess}(\lambda_{pq}(\cdot)) \tag{5.3}$$

$$N_{uv}(\cdot) \sim \mathrm{Thinning}(N_{\pi(u)\pi(v)}) \tag{5.4}$$

where α is the concentration parameter of the CRP, $\gamma_{pq}, \beta_{pq}, w_{pq}$ are the base scaling parameter of base intensity, scaling parameter of the self-exiting intensity and width parameter of the self-exiting intensity respectively and where n_p refers the the number of nodes belong to the p-th group and where $\pi(u)$ denotes the particular group that node u belongs to.

5.3 Fragmentation Coagulation Hawkes Interaction model

Our Fragmentation Coagulation Hawkes (FCH) IRM allows the dynamic evolving behaviours in both the groupings on the nodes and the intensity functions of the group-wise Hawkes processes. This is achieved by introducing the notable Fragmentation Coagulation Process [15], [77]. The detail operations are constructed as follows.

5.3.1 Nodes' clustering evolving

Fixing the groupings on the nodes is an over-simplified assumption in the real world scenario. Take the employee interaction network in a company as an example. It is quite common that employees belong to the same department of the company would have similar work interaction patterns. On the other

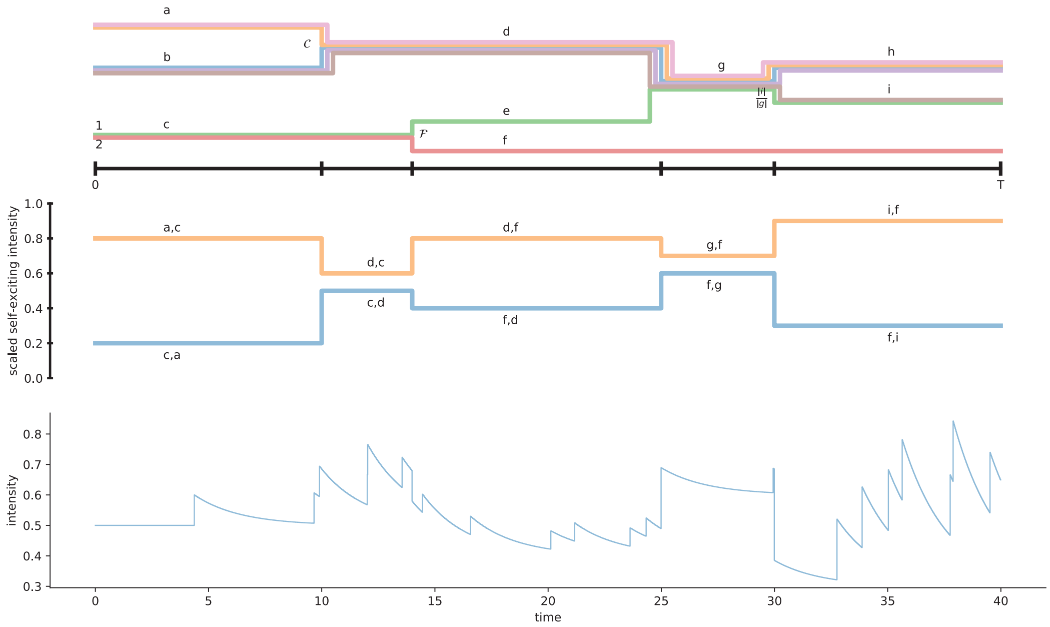


Figure 5.1: Top: FCP simulation. $\mathcal{F}$ and $\mathcal{C}$ represent the fragmentation and coagulation process respectively. Middle: scaled self-exciting intensity of Hawkes process for community relation along two community paths: $\{a, d, g, i\}$ and $\{c, f\}$. Bottom: the intensity of Hawkes process associated with interactions from entity 1 to entity 2.

hand, the company may merge or split different departments to adjust to the current challenges. Allowing the employees' group belonging seems quite a natural requirement.

We use the Fragmentation Coagulation Process to allow the nodes have dynamic grouping behaviours. On one hand, In our model, FCP mainly guides the construction of community evolution for entities in continuous time. An illustration of FCP-Hawkes for the entity partition is shown in Fig.5.1. At time $t = 0$, the entities are segmented into communities following the Chinese restaurant process (CRP). At any time $t, t \in (0, T]$, the community with its associated entities can be split into two communities or two communities can be merged into one community based on FCP. Each entity i is associated with a community indicator $z_{i,t}$. Community path $\mathbf{z}_i$ of entity i is $\mathbf{z}_i = \{z_{i,t} | t \in [0, T)\}$. And community pair path for entity i and j is $\mathbf{z}_{\{i,j\}} = \{(z_{i,t}, z_{j,t}) | t \in [0, T)\}$.

5.3.2 Hawkes Process' evolving

In reality, it is common that two aspects influence the interaction between entities: 1, community behaviour pattern by surrounded environment. For example, entities interaction share similar behaviour within same community; 2, personal interaction history (the previous interactions between entities) This corresponds to the general case that entity i will respond after entity j interacts with i. In order to incorporate the aforementioned two aspects, Hawkes process with the typical exponential form is implemented. The intensity of Hawkes process on the segment associated with community pair $\{p, q\}$ is parameterized by self exciting rate (base intensity) $b_{p,q}$ and reciprocating kernel (triggering intensity) $g_{p,q}$. Here $g_{p,q}$ represents the parameter $\eta_{p,q}, \zeta_{p,q}$ in the exponential kernel: $g_{p,q}(t) = \eta_{p,q} e^{-\zeta_{p,q} t}$. Self exciting rate $b_{p,q}$ and reciprocating kernel $g_{p,q}$ correspond to the current influence from community p to community q and the influence of history from community q to community p respectively. For further usage, We also define the correspondingly unscaled self exciting rate as $\overline{b}_{p,q}$ and reciprocating kernel $\overline{g}_{p,q}$ ($\overline{\eta}_{p,q}, \overline{\zeta}_{p,q}$).

To parameterize the intensity of Hawkes process for each community pair, we start with $t = 0$. The $\overline{b}_{p,q}$ and $\overline{g}_{p,q}$ for community p and q at $t = 0$ are sampled from Gaussian distribution respectively. When the community does the fragmentation or coagulation process at t^-, the new $\bar{b}_{z_{i,t},z_{j,t}}$ is generated from the Gaussian distribution with its mean $\bar{b}_{z_{i,t^-},z_{j,t^-}}$ or transformed from linear combination of $\bar{b}_{z_{i,t^-},z_{j,t^-}}$ and $\bar{b}_{z_{j,t^-},z_{i,t^-}}$, for $\bar{g}_{z_{i,t},z_{j,t}}$ vice verca. (Please refer to the generative process for more details.) Besides, To guarantee the non-negativity, the self exciting rate $b_{p,q}$ and reciprocating kernel $g_{p,q}$ are derived by passing the unscaled ones through a sigmoid function, and then scaled by corresponding b^* and g^* (η^*, ζ^*) for a upper bound.

Entity Interaction Generation

Finally, the interaction t_{ij}^s is generated by Hawkes process with the $b_{z_{i,t_s},z_{j,t_s}}$ and previous interactions $\mathcal{H}$ from entity j to i with their associated reciprocating kernel along the associated community pair path $\mathbf{z}_{\{i,j\}}$ where $\mathcal{H} = \{t_{ji}^r | \{t_{ji}^r < t_{ji}^s\}$.

For convenience, for the parameters in Hawkes process, we only introduce the generative process on the self-exciting related parameters, the parameters for the reciprocating related parameters is constructed in a similar way. In summary, the FCP-Hawkes generative model is as follows:

- At time $t = 0$,
  - Sample community indicator $\{z_{i,0}\}_{i=1}^N \sim \text{CRP}(\frac{\nu}{\xi})$ (let $K^* = \max_i z_{i,0}$)
  - For $k_1, k_2 \in \{1, \ldots, K^*\}$, sample Hawkes process' parameters $\bar{b}_{k_1,k_2} \sim \text{N}(\mu_b, \sigma^2)$
- Transform the parameters of the Hawkes process whenever coagulation or fragmentation on the communities in FCP occurs:
  - **Fragment** community z_{i,t^-} (z_{j,t^-}) containing entity i, j **into** community $z_{i,t}$ containing entity i and $z_{j,t}$ containing entity j, we have

$$\bar{b}_{z_{i,t},z_{k,t}}, \bar{b}_{z_{j,t},z_{k,t}} \sim \text{N}(\bar{b}_{z_{i,t^-},z_{k,t^-}}, \sigma^2) \tag{5.5}$$

$$\bar{b}_{z_{k,t},z_{i,t}}, \bar{b}_{z_{k,t},z_{j,t}} \sim \text{N}(\bar{b}_{z_{k,t^-},z_{i,t^-}}, \sigma^2) \tag{5.6}$$

- **Coagulate** community z_{i,t^-}, z_{j,t^-} **into** community $z_{i,t}$ $(z_{j,t})$, we have

$$\bar{b}_{z_{i,t},z_{k,t}} \sim \mathrm{N}(\frac{\bar{b}_{z_{i,t^-},z_{k,t^-}} + \bar{b}_{z_{j,t^-},z_{k,t^-}}}{2}, \sigma^2) \tag{5.7}$$

$$\bar{b}_{z_{k,t},z_{i,t}} \sim \mathrm{N}(\frac{\bar{b}_{z_{k,t^-},z_{i,t^-}} + \bar{b}_{z_{k,t^-},z_{j,t^-}}}{2}, \sigma^2) \tag{5.8}$$

- Scaling the parameters of the Hawkes process:
  - Sample the scaling parameter $b^* \sim \mathrm{gamma}(\alpha, \beta)$
  - $b_{z_{i,t},z_{i,t}} = b^* \frac{1}{1+\exp{(-\bar{b}_{z_{i,t},z_{i,t}})}}$
- For each interaction $t^s_{ij} \in [0, T)$ from entity i to j, we have $t^s_{ij} \sim \mathcal{PP}(b_{z_{i,t^s_{ij}},z_{j,t^s_{ij}}} + \sum_{\mathcal{H}} \psi_{g_{z_{i,t^r_{ji}},z_{j,t^r_{ji}}}}(t^s_{ji} - t^r_{ji}))$, $\mathcal{H} = \{t^r_{ji} | \{t^r_{ji} < t^s_{ji}\}$

5.4 Inference

In this section, we describe the inference scheme of the proposed FCP-Hawkes model. Along the flexibility of FCP-Hawkes model, it also faces two serious problems from the inference part: (1) the model introduces the FCP, so it brings the infinite communities; (2) the model with the HP breaks the Markov property, so the backward-forward algorithm for temporal models (e.g. hidden markov model) can't be used. To overcome these two problems, we firstly adopt the MCMC inference for Markov Jump Process (MJP) [68] to deal with FCP. The method mainly constructs a Markov chain with a set of "potential jump points", so it allows us to do sampling on discrete points. Secondly, we use conditional sequential monte carlo (C-SMC) to deal with the non-Markovian property of the sequence of hidden variables.

5.4.1 Sampling potential jump point $\mathcal{J}$

Discretizing time is a simple approach for inference for FCP, while it will lead to long Markov chains with the high computation cost. As FCP is a continuous Markov process, a discrete-time Markov chain can be constructed by uniformization for a random time-discretization. Assume that the marginal distribution π_0 of FCP at time 0 and rate matrix A are known. The marginal distribution π_t at time t can be expressed as:

$$\begin{aligned}\pi_t &= \exp(At)\pi_0 \\ &= \sum_{n=0}^{\infty}((\exp(-\Omega t)\frac{\Omega t^n}{n!})(B^n \pi_0))\end{aligned}$$

where $B = (I + \frac{1}{\Omega}A)$. Here in the summation the first term can be interpreted as a Poisson distribution with rate Ω and B of the second term can be considered as a probability transition matrix. So a discrete Markov chain can be constructed in such a way:

1. Generate potential jump points $\mathcal{J}$ by Poisson process ($\mathcal{PP}$) with rate Ω.

2. Set the transition probability matrix of the discrete Markov chain as B.

5.4.2 Sampling $\mathbf{z}$

Given the potential jump point $\mathcal{J}$, so $\mathbf{z}$ can be sampled from a discrete Markov chain. However, the classical backward-forward algorithm can't be used due to the non-Markovian property of HP. Instead, the conditional sequential Monte Carlo (C-SMC) is implemented to sample the set of community indicators $\mathbf{z}$. The full detail of C-SMC is provided in Alg.4, where the proposal distribution of z_t, $r_{\theta,t}(z_t|z_{1:t-1}^{a_t^i})$, is defined as $r_{\theta,t}(z_t|z_{1:t-1}^{a_t^i}) = p(z_t|z_{t-1}^{a_t^i})$, so the unnormalized weight w_t^i can be simplified as $w_t^i = p(y_t|x_{1:t}, y_{1:t-1})$.

5.4.3 Metropolis-Hasting Sampling for Hawkes Parameter

For simplicity, we just introduce the sampling scheme of main parameters b, g (η, ζ) in general HP. All the parameters in our model used to generate them via nonlinear transformation of $\overline{b}, \overline{g}$ $(\overline{\eta}, \overline{\zeta})$ with the scaling parameter b^*, g^* can be sampled in a similar way.

The likelihood $\mathcal{L}_{ij}^{pq}$ with respect to interactions from entity i to j where entity i and j go through community p and q respectively, conditioned on the interactions from entity j to i:

$$\begin{aligned}\mathcal{L}_{ij} &= (\prod_{s=1}^{S} \lambda(t_{ij}^{s})) \exp(-\int_{0}^{T} \lambda(t))\, dt \\ &= \prod_{s=1}^{S} (b_{z_i^{t_{ij}^{s}}, z_j^{t_{ij}^{s}}} + \sum_{t_{ji}^{r} < t_{ij}^{s}} g_{z_i^{t_{ji}^{r}}, z_j^{t_{ji}^{r}}} (t_{ij}^{s} - t_{ji}^{r})) \\ & \exp(-(\int_{0}^{T} b_{z_i^t, z_j^t}\, dt + \sum_{r} \int_{t_{ji}^{r}}^{T} g_{z_i^{t_{ji}^{r}}, z_j^{t_{ji}^{r}}} (t - t_{ji}^{r})\, dt))\end{aligned}$$

Here we describe the update of $b_{p,q}$, which is similar with $g_{p,q}$. The proposed distribution is the normal distribution with the mean of current value $b_{p,q}$. The acceptance ratio of the proposed sample $\hat{b}_{p,q}$ is given as: $A(\hat{b}_{p,q}, b_{p,q}) = \min[1, \frac{\hat{p}(\hat{b}_{p,q})}{\hat{p}(b_{p,q})}]$. The ratio can be calculated as:

$$\begin{aligned}\frac{\hat{p}(\hat{b}_{p,q})}{\hat{p}(b_{p,q})} &= \frac{p(\hat{b}_{p,q})}{p(b_{p,q})} \cdot \exp(-\int_{t_{p,q}^{-}}^{t_{p,q}^{+}} \hat{b}_{p,q} - b_{p,q}\, dt) \\ \prod_{t_{ij}^{s} \in [t_{p,q}^{-}, t_{p,q}^{+})} & \frac{\hat{b}_{p,q} + \sum_{t_{ji}^{r} < t_{ij}^{s}} g_{z_i^{t_{ji}^{r}}, z_j^{t_{ji}^{r}}} (t_{ij}^{s} - t_{ji}^{r})}{b_{p,q} + \sum_{t_{ji}^{r} < t_{ij}^{s}} g_{z_i^{t_{ji}^{r}}, z_j^{t_{ji}^{r}}} (t_{ij}^{s} - t_{ji}^{r})}\end{aligned}$$

Here we assume the community pair (p, q) exists among time $[t_{p,q}^{-}, t_{p,q}^{+})$

5.4.4 Inference Framework

Formally, given a sequence of interactions t_{ij}^{s}, the algorithm needs to infer the hidden variable $\mathbf{z}$, $\mathbf{z} = \{\mathbf{z}_i | i \in 1, ..., N\}$, the scaling parameter b^* and g^* (η^*, ζ^*), unscaled self-exciting and reciprocating kernel $\overline{b}_{p,q}$ and $\overline{g}_{p,q}$ $(\overline{\eta}_{p,q}, \overline{\zeta}_{p,q})$. We use the Ω to indicate the all variables excluding the hidden variable $\mathbf{z}$, $\Omega = \{b^*, g^*, \overline{b}_{p,q}, \overline{g}_{p,q}\ (\overline{\eta}_{p,q}, \overline{\zeta}_{p,q})\}$. Generally, for each iteration, we sample the "potential jump points" with a constant rate of Poisson process $(\mathcal{PP})$, and then we sample $\mathbf{z}$ and Ω separably.

The sampling scheme is given as follows:

- For k=0,
  - initialize set $\mathbf{z}(k), \Omega(k)$

- For iteration, $k \geq 1$,
  - for each entity i,
    - $*$ sample potential jumps $\mathcal{J}^{\text{aux}}(k) \sim \mathcal{PP}(\Lambda(t))$ with rate $\Lambda(t) = Q_t(z_i^t(k-1), z_i^t(k-1))$
    - $*$ sample $\mathbf{z}_i(k)$ conditioned on $\Omega(k-1), \mathbf{z}(k-1) \cup \mathcal{J}^{\text{aux}}(k)$
  - sample $\Omega(k)$ conditioned on $\mathbf{x}, \mathbf{z}(k-1)$

Notation:

- k is the iteration index
- $\omega_t > \max_{s \in C_{it}} - Q_t(s,s)$
- $\mathbf{z}_i$ is the community paths for entity i
- $Q_t(s,s) = -\sum_{s' \neq s} Q_t(s,s')$
- For $s, s' \in C_{it}, Q_t(s,s')$ is the transition rate from community s to s'
- $C_{it} = \mathbf{z}^t_{\text{-}i} \cup \emptyset$

Algorithm 4 C-SMC with non-markovian kernel for FCP-Hawkes

```
Input: $t \in \mathcal{J}' \cup \mathcal{J}^{\text{aux}}$
Draw $z_1^i \sim p(z_1^i)p(y_1|z_1^i)$ for $i = 1, ..., N$
Set $a_1^i = i$ for $i = 1, ..., N$
Set $w_1^i = W_{\theta,1}^i(z_1)$ for $i = 1, ..., N$
for $t = 2, ..., T$ do
    Draw $a_t^i \sim \frac{w_{t-1}^i}{\sum_l w_{t-1}^l}$ for $i = 1, ..., N$
    Draw $z_t^i \sim r_{\theta,t}(z_t|z_{1:t-1}^{a_t^i})$ for $i = 1, ..., N$
    Set $z_{1:t}^i = (z_{1:t-1}^{a_t^i}; z_t^i)$
    Set $w_t^i = W_{\theta,t}^i(z_{1:t})$
end for
Draw $k$ with $p(k = i) \sim w_T^i$
return $z_{1:T}^* = z_{1:T}^k$
```

5.5 Experiments

5.5.1 Synthetic Data

To demonstrate the problem of CRP-Hawkes we generate two synthetic dataset: static and dynamic to verify our model. The main difference between static and dynamic data is that the belonging community of each entity is allowed to change across time. Here we generally introduce the generative procedure of dynamic dataset.

1. Instantiate a network structure of two communities. At time 0, the entities are randomly assigned to one of the communities. At time $T/2$, one of the communities is allowed to split into two smaller communities.

2. An M-variate hawkes process is used to generate the interactions between entities. The M represents the number of communities. The simulation of the M-variate hawkes process is provided in Alg.5 where $\boldsymbol{b}_{M\times 1}, \boldsymbol{\eta}_{M\times M}, \boldsymbol{\zeta}_{M\times M}$ represent the parameter of self-exciting and reciprocating kernel in hawkes process respectively.

For evaluation, we use the visualization to directly show the performance of the FCP-hawkes model. The Figure 5.2 shows the real and inferred community evolution of the synthetic data respectively. The community of synthetic data have a split operation at time 5. However, such a operation can't be well modelled in the FCP-IRM method. From the figure, it is easily to find that the FCP-hawkes model completely recovered the community evolution of the dynamic synthetic data and the split operation only has a delay with 0.2 seconds. And all the entities belong to the corresponding correctly communities. The log-likelihood of both the static and dynamic synthetic data are stable as shown in Figure 5.3. The inferred parameters of the FCP-hawkes model is also shown in table 5.1. The inferred parameters related to the Hawkes process share the similar values with the real ones.

Algorithm 5 Simulation of M-variate hawkes process with exponential kernel: $\gamma_{mn}(u) = \eta_{mn}e^{-\zeta_{mn}u}$for synthetic data

```
Input: $\boldsymbol{b}_{M\times 1}, \boldsymbol{\eta}_{M\times M}, \boldsymbol{\zeta}_{M\times M}, T$
Initialize $\mathcal{T}^1, \ldots, \mathcal{T}^M$
while $s < T$ do
    $\overline{\lambda} = \sum_{m=1}^{k} \lambda^m(s)$
    $= \sum_{m=1}^{M}(b_m + \sum_{n=1}^{M} \sum_{\tau \in \mathcal{T}^n} \eta_{mn}e^{-\zeta_{mn}(s-\tau)})$
    Generate $u \sim$ uniform$(0, 1)$
    Set $w = -\ln(u)/\overline{\lambda}$
    Set $s = s + w$
    Generate $u \sim$ uniform$(0, 1)$
    if $u\overline{\lambda} < \sum_{m=1}^{M}(b_m + \sum_{n=1}^{M} \sum_{\tau \in \mathcal{T}^n} \eta_{mn}e^{-\zeta_{mn}(s-\tau)})$ then
        $k = 1$
        while $u\overline{\lambda} > \sum_{m=1}^{k} \lambda^m(s)$ do
            $k = k + 1$
        end while
        $\mathcal{T}^k = \mathcal{T}^k \cup s$
    end if
end while
if $s < T$ then
    return $\mathcal{T}^1, \ldots, \mathcal{T}^M$
else
    return $\mathcal{T}^1, \ldots, \mathcal{T}^k \setminus s, \ldots, \mathcal{T}^M$
end if
```

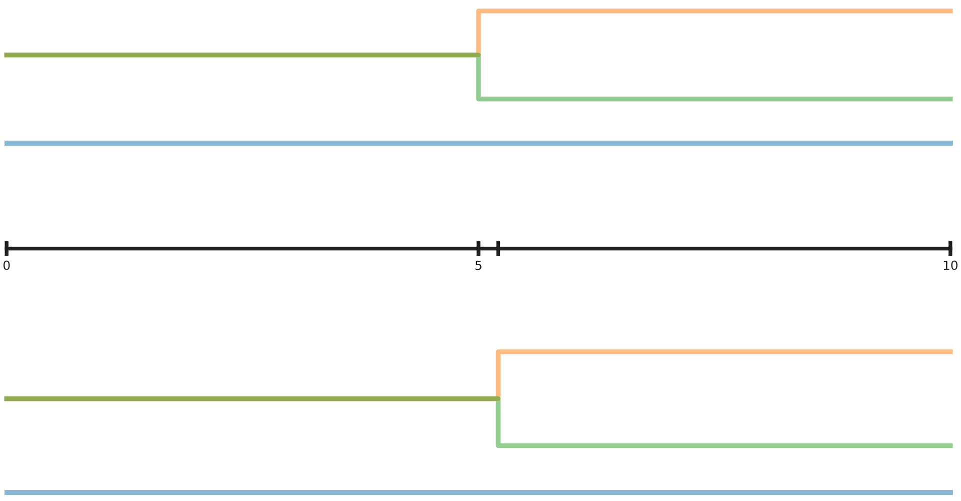


Figure 5.2: Cluster evolution for synthetic data. Top: real community evolution. Bottom: learned community evolution.

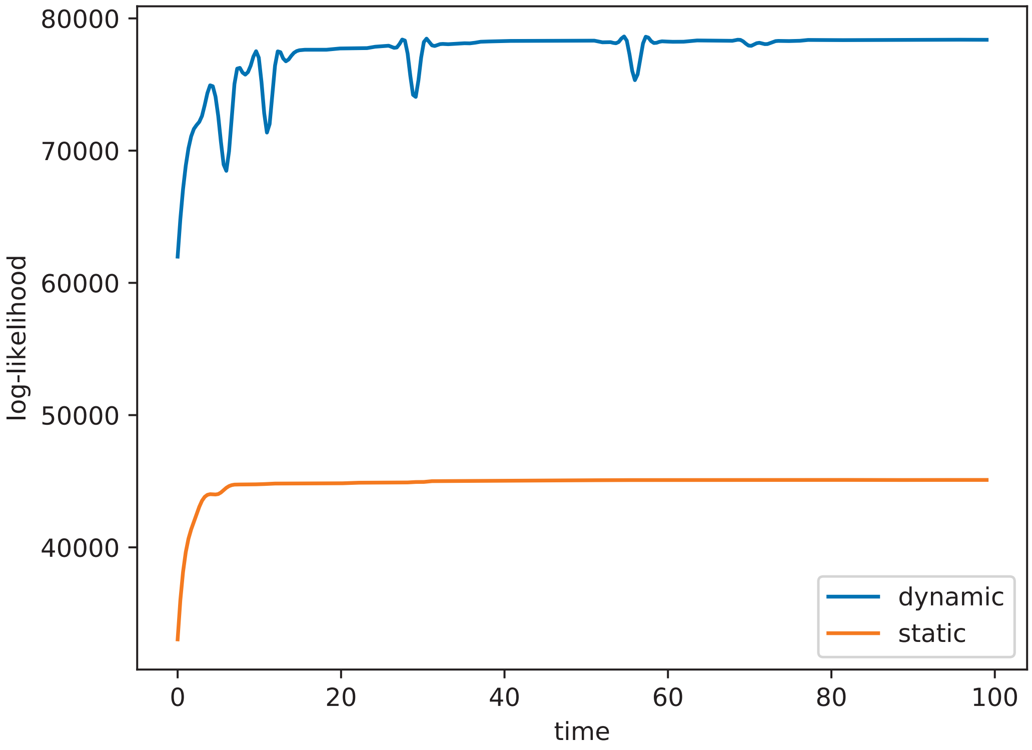


Figure 5.3: Log-likelihood of static and dynamic synthetic data.

		Real			Inferred		
send	receive	b	η	ζ	b	η	ζ
1	1	5.0	0.0	0.1	2.4	0.5	1.0
1	2	0.1	0.0	0.1	0.1	0.0	0.0
1	3	0.1	0.0	0.1	0.1	0.0	0.0
1	4	0.1	0.0	0.1	0.1	0.0	0.0
2	1	0.1	0.0	0.1	0.1	0.0	0.0
2	2	10	0.0	0.1	9.8	0.0	1.3
2	3	–	–	–	–	–	–
2	4	–	–	–	–	–	–
3	1	0.1	0.0	0.1	0.1	0.0	0.0
3	2	–	–	–	–	–	–
3	3	10	0.0	0.1	10.2	0.0	1.3
3	4	0.1	0.0	0.1	0.1	0.0	1.3
4	1	0.1	0.0	0.1	0.1	0.0	0.0
4	2	–	–	–	–	–	–
4	3	0.1	0.0	0.1	0.1	0.0	0.7
4	4	15	0.0	0.1	13.8	0.1	1.8

Table 5.1: The real and inferred parameter of the synthetic data

5.5.2 Test Case

Data Description

The UN dataset [5] contains the roll-call votes in the UN General Assembly between 1946 and 2019. We choose one sub-topic related to the Palestinian conflict as our test case. Overall 201 countries participated for vote. The first and last day for the vote are $1947-05-02$ and $2018-12-20$. All the dates are transformed into numerical values ranged from 0 to 27.1 in our model. There are 5 types of vote choices: yes, abstain, no, absent and not a member. Each two countries' vote with the same choice is considered as an interaction between them in our model. Therefore, for this test case, the interactions are un-directional.

Result Analysis

To validate the performance of our model, different visualizations are shown in this section. We compare our model with the FCP-Poisson and CRP-Hawkes model in Figure 5.4. And it is easily found that the FCP-Hawkes outperforms over other two models. In Figure 5.5, the statistics of the initialized communities are shown. It is easily observed that the mean of the country's first vote for the first community is far away from from the other three communities. And within these three communities, the elements are mainly constituted by the countries that have their first vote at time 0. Besides, looking at the compatibility of community relations, the intensity is really high within the second community while the intensity is relatively high between the second and fourth community. This explains the reason the FCP-Hawkes model has four communities at time 0.

We randomly choose several countries, and present the real and base intensity of the within/across community relation according their community paths in Figure 5.6 and 5.7. We find the FCP-Hawkes model almost recovered the real ones. Besides, as there are some operations happened in a very short time in the model, this leads to the spike in the real ones. Compared to the Poisson-Hawkes, it can be solved by the reciprocating part of the Hawkes process in our model. In Figure 5.8, we show the intensity of relation along two community paths which are originated from the same community at time 0. Firstly compared with the real ones shown in the top figure, the modelled ones follow the same pattern show in the bottom figure. Secondly, the difference is obvious between the intensity of each within community relation. This explains why around $t = 1.8$ there is split operation. Therefore, the fragmentation and coagulation process are useful to model the relations between communities and may indicate the change of the community pair relation. This phenomenon is also reflected in the Figure 5.10 which shows the number of operations (total number of fragmentation and coagulation) across the time. As around 30 countries had their first vote between time 10 and 15 as shown in the Figure 5.9, it is predictable that as more countries are involved, more dynamics with the community structure will happen. In the Figure 5.10, the number of operations matches the pattern and after time 10 the changes (operation) are more intense compared with the ones before time 10.

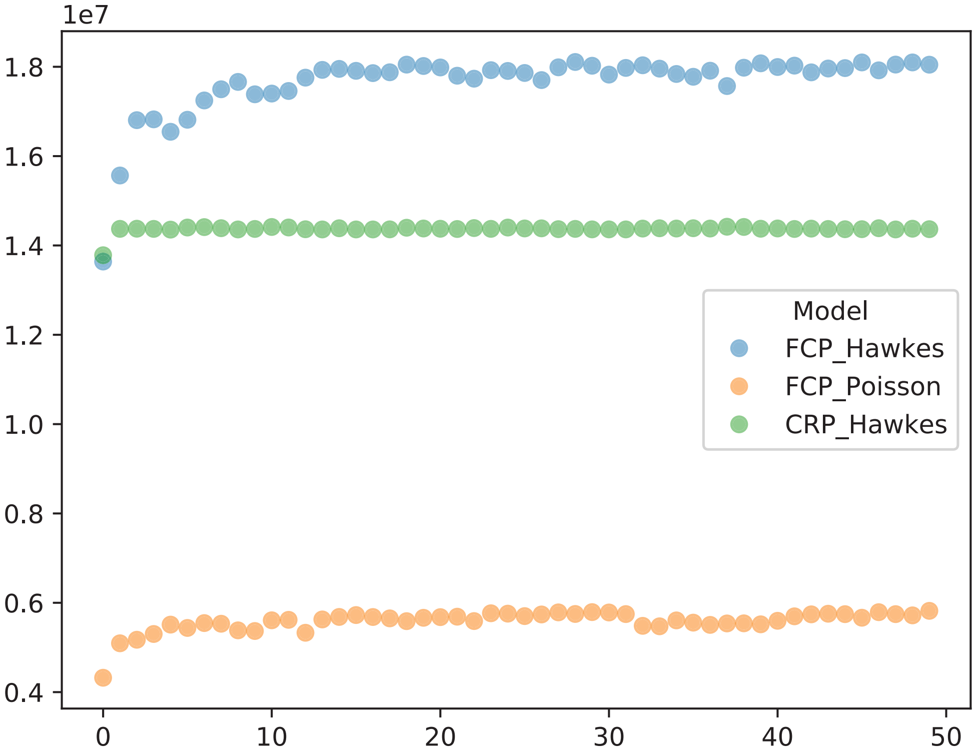


Figure 5.4: Likelihood for model comparison.

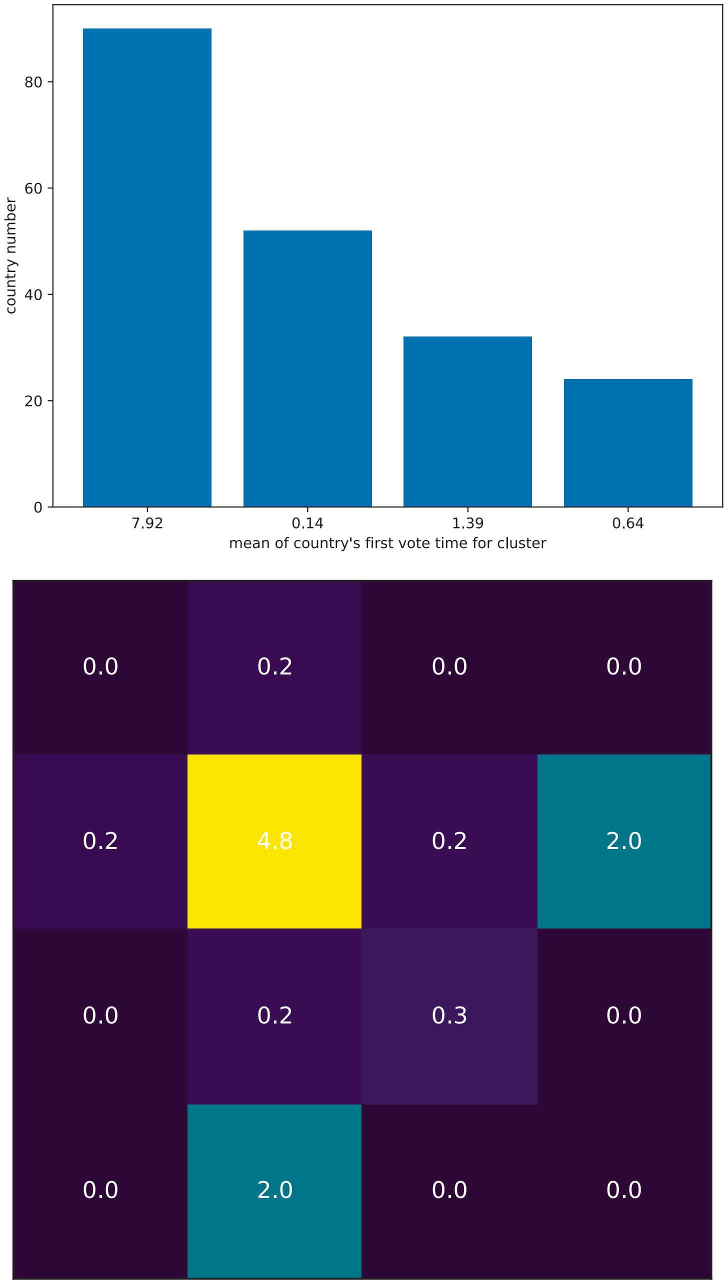


Figure 5.5: Statistics of the initialized communities.

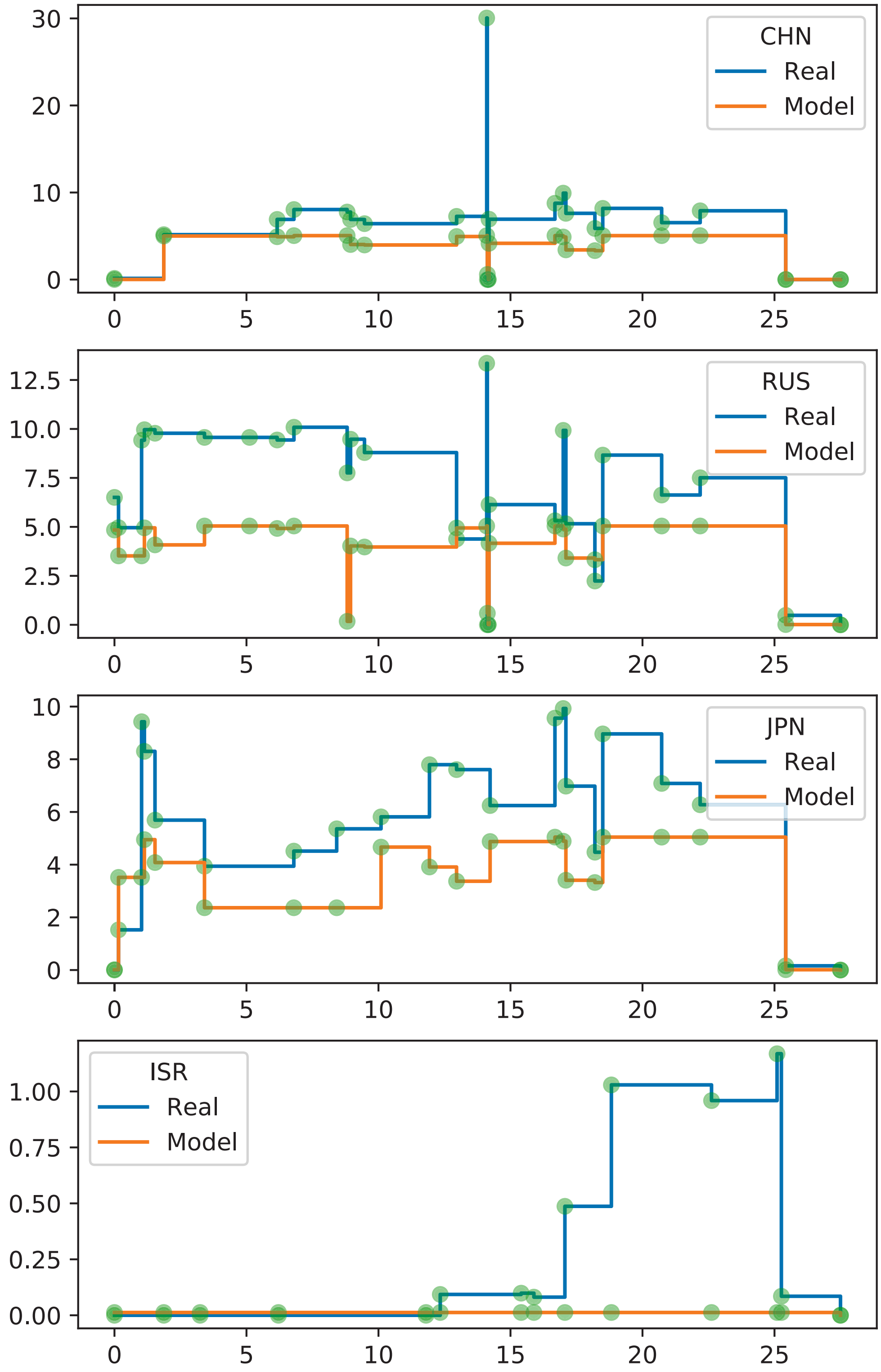


Figure 5.6: Real and modelled base intensity of within-community relations

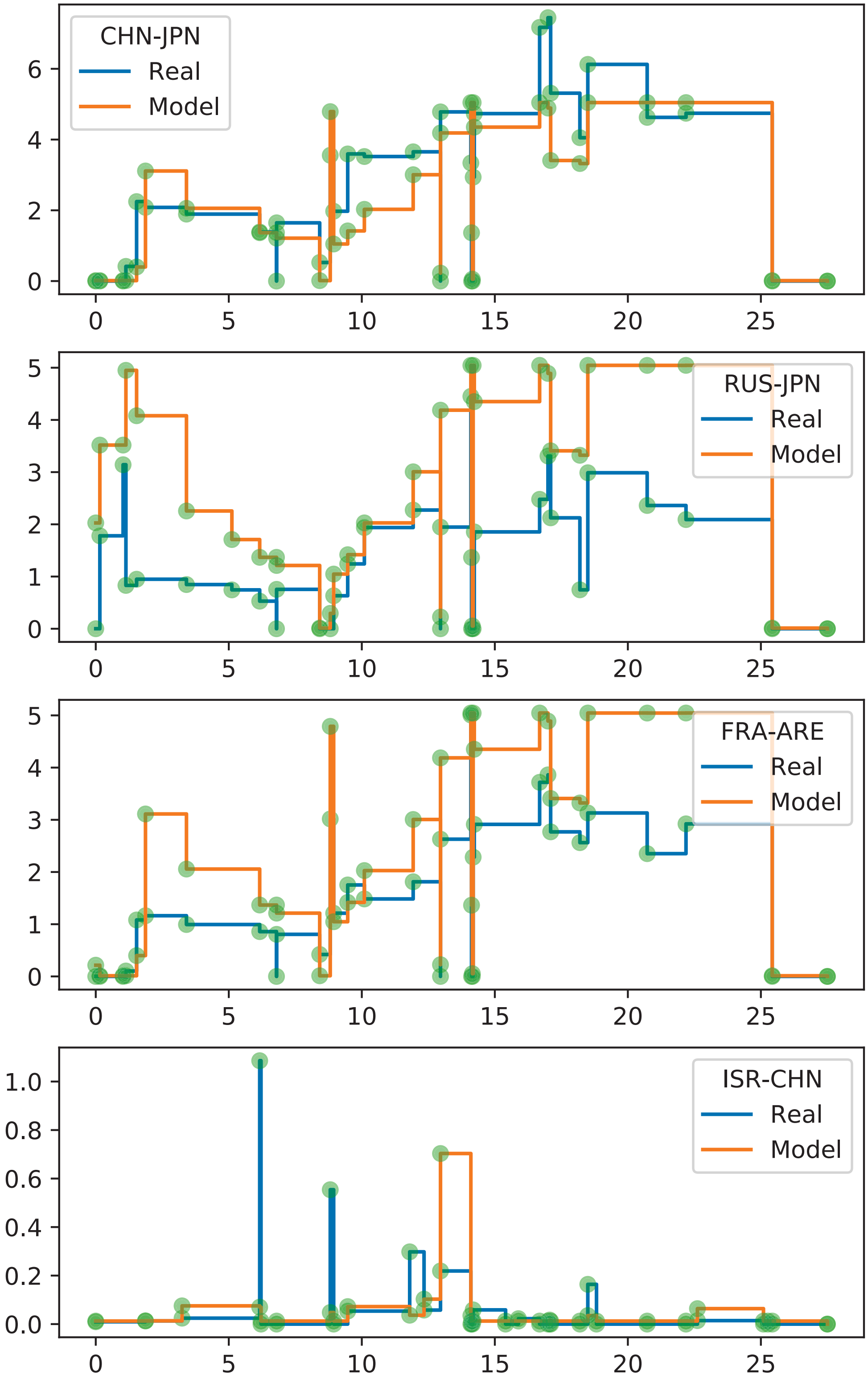


Figure 5.7: Real and modelled base intensity of across-community relations

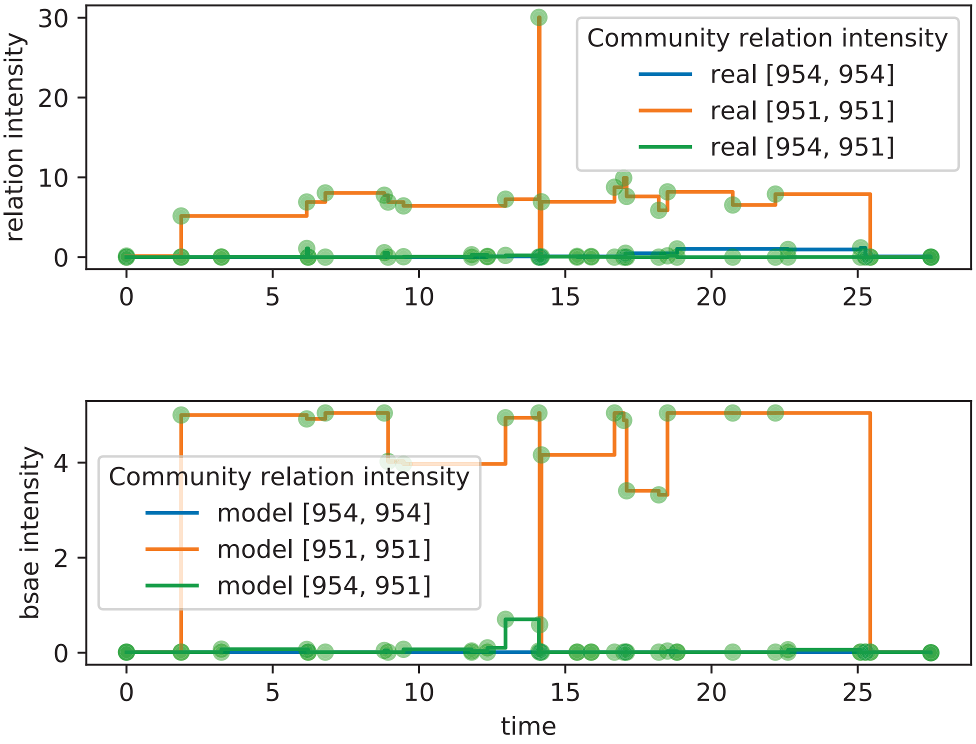


Figure 5.8: Example of split operation in UN data.

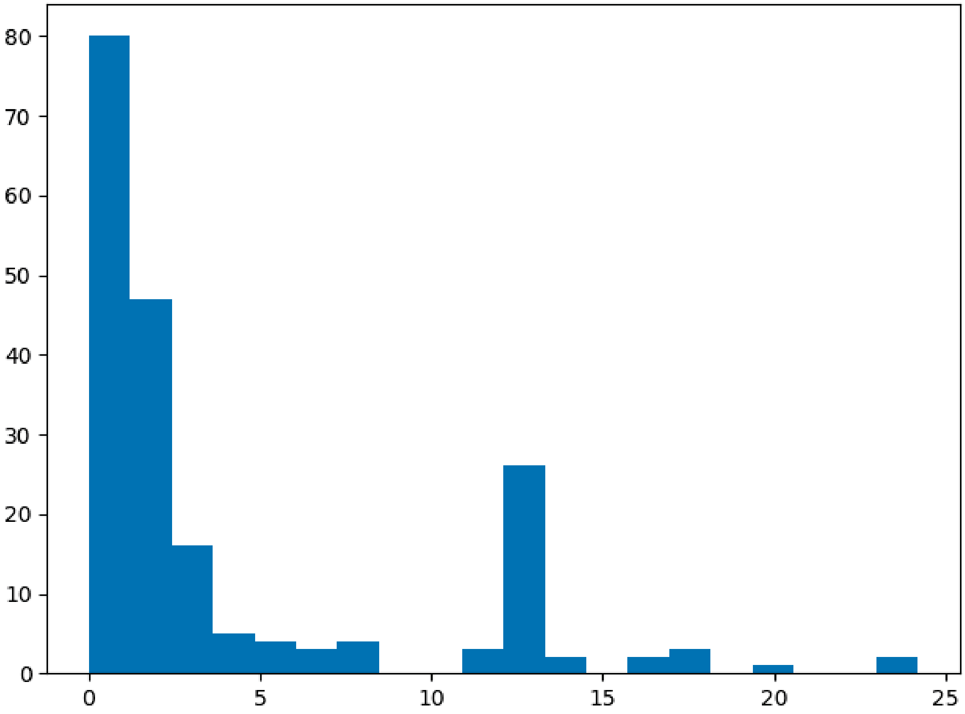


Figure 5.9: country first vote

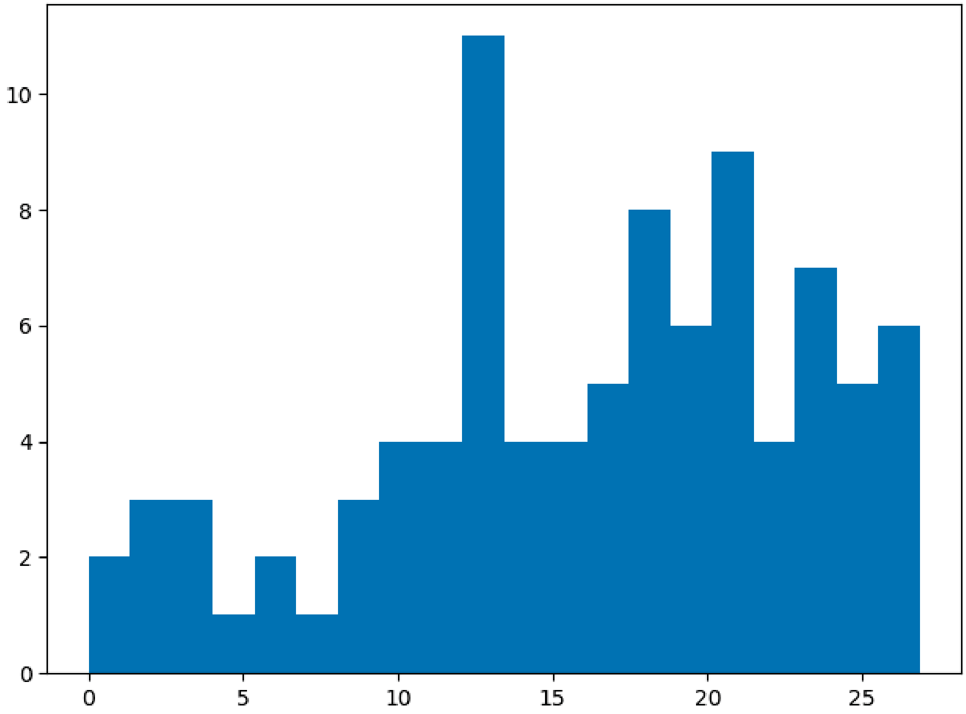


Figure 5.10: fcp operation time

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, several methods are presented to deal with different problems related to analysis of complex networks. In Chapter 1, the research field, problems and objectives are briefly introduced. Chapter 2 mainly focuses on the general methods used in this research and the related literature review. Chapters 3, 4 and 5 provide description of proposed methods to deal with the problems stated in Chapter 1. They focus on analysis of dependencies between entities' features and their changes over time, structure itself and structure evolution of complex networks, and continuous time-series data modelling.

Chapter 2 is composed of a number of background related topics: preliminary knowledge on the deep learning technique including the classical neural network and long short term memory, nonparametric Bayesian methods including the Dirichlet process and the Chinese restaurant process, the sampling method, Monte Carlo Markov Chain (MCMC), and several stochastic processes including the Poisson process and the Hawkes process. Also, the literature review covers the recent developments on the relational models including latent class methods, latent feature methods, matrix factorization methods and deep learning methods.

In Chapter 3, we describe the limitation of mixed membership stochastic blockmodels in inferring relations between two communities when the entity relations between these communities are unobserved. We propose a solution to this problem by factorizing the community relations matrix into two community feature matrices, thereby adding a dependency between community relations. Besides, the deep learning techniques is used to approximate the components of classical probabilistic relational model for complex network. We introduce the deep dynamic mixed membership stochastic blockmodel based network (DDBN) to demonstrate the feasibility of such an approach. The DDBN takes the advantage of the matrix factorizaiton to solve the above problem and marries the mixed membership stochastic blockmodel (MMSB) with deep neural networks for rich feature extraction and introduces a temporal dependency in latent features using a long short-term memory unit for dynamic network modeling.

In Chapter 4, we state that the current formulation of MMSB suffers from the following two issues: (1) inability to embed any prior information ,e.g., entities' community structural information, in the modelling process; (2) community evolution is not properly addressed in the literature. Therefore, we propose a non-parametric fragmentation coagulation based Mixed Membership Stochastic Blockmodel (fcMMSB). Our model performs entity-based clustering to capture the community information for entities and linkage-based clustering to derive the group information for links simultaneously. Besides, the proposed model infers the network structure and models community evolution, manifested by appearances and disappearances of communities, using the discrete fragmentation coagulation process (DFCP). By integrating the community structure with the group compatibility matrix we derive a generalized version of MMSB. An efficient Gibbs sampling scheme with Polya Gamma (PG) approach is also implemented for posterior inference.

In Chapter 5, we present a unified framework for processing continuous time-series data. It incorporates the fragmentation coagulation process with the Hawkes process. The fragmentation coagulation detects evolution of the community structure in the complex network occurring in the continuous time. This process models the reciprocating relations between entities. The propoed method can be considered as a continuous/dynamic CRP-Hawkes in contrast to the CRP-Hawkes [7].

6.2 Future Work

Nowadays, large amounts of data are easily accessible on the Internet. There are millions of user active on such platforms like Facebook and Youtube. The current methods used to construct Bayesian relational model are not able to deal with large-scale modelling. It is especially seen in Bayesian models that use the series-sampling method for inference – its computational complexity is $O(N^2)$. This prompts us to find or develop more efficient inference methods for the relational model.

One possible solution is to apply the variational inference which uses the variational bound (KL divergence) as guidance to optimize the objective function or the posterior distribution. A few works contributed to this field to a large extent. [6] firstly applied the variation inference (VI) on the Dirichlet process. Sebsequently, to deal with the efficiency problem of nested Chinese restaurant process (nCRP), [80] developed the corresponding variational method to nCRP. Furthemore, such techniques were also used in hierarchical Dirichlet process (HDP) [81] for online learning. [31] integrated the stochastic optimization with VI for the HDP. VI is also developed in the Poisson process [2] which can be a promising field to explore.

Besides, another approch to reduce the dimensionality is to consider different ways of constructing relational models. One interesting and promising method is to use Ber-Poisson link function [12]. For this method, observations between nodes with no linkage do not need to be sampled. Therefore, the computational cost of $O(N^2)$ can be reduced to $O(N_e)$. This makes the sampling method more efficient. Several works [92],[67] [35] and [18] that applied this approach have already shown good performance. This method could also be used to extend our work presented in Chapter 4.

However, the efficiency problem is the most crucial one in the case of continuous time-series modelling. One advanced solution to Continuous-Time Bayesian Networks (CTBN) has been proposed in [54]. In the paper, the authors have integrated a gradient-based approach with a variational method to address the issue of scalability. This approach deserves an attention as it may open up a new direction in process of continuous data modeling. Besides, another simple way to reduce the computational complexity is to use a sliding window over the history information for the Hawkes process. As we can see, the influence of the history for the Hawkes process will decrease with time, so selection of an appropriate size of a sliding window not only allows to keep most of useful information but also enables reduction of the computational complexity.

References

[1] L. A. Adamic and E. Adar, “Friends and neighbors on the web,” *Social networks*, vol. 25, no. 3, pp. 211–230, 2003.

[2] V. Aglietti, E. V. Bonilla, T. Damoulas, and S. Cripps, “Structured variational inference in continuous cox process models,” in *Advances in Neural Information Processing Systems*, 2019, pp. 12 437–12 447.

[3] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, “Mixed membership stochastic blockmodels,” *Journal of Machine Learning Research*, vol. 9, no. Sep, pp. 1981–2014, 2008.

[4] D. J. Aldous, “Representations for partially exchangeable arrays of random variables,” *Journal of Multivariate Analysis*, vol. 11, no. 4, pp. 581–598, 1981.

[5] M. A. Bailey, A. Strezhnev, and E. Voeten, “Estimating dynamic state preferences from united nations voting data,” *Journal of Conflict Resolution*, vol. 61, no. 2, pp. 430–456, 2017.

[6] D. M. Blei, M. I. Jordan, *et al.*, “Variational inference for dirichlet process mixtures,” *Bayesian analysis*, vol. 1, no. 1, pp. 121–143, 2006.

[7] C. Blundell, J. Beck, and K. A. Heller, “Modelling reciprocating relationships with hawkes processes,” in *Advances in Neural Information Processing Systems 25*, 2012, pp. 2600–2608.

[8] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” *arXiv preprint arXiv:1312.6203*, 2013.

[9] F. R. Chung and F. C. Graham, *Spectral graph theory*, 92. American Mathematical Soc., 1997.

[10] J. S. Coleman *et al.*, “Introduction to mathematical sociology.,” *Introduction to mathematical sociology.*, 1964.

[11] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Advances in neural information processing systems*, 2016, pp. 3844–3852.

[12] D. B. Dunson and A. H. Herring, “Bayesian latent variable models for mixed discrete outcomes,” *Biostatistics*, vol. 6, no. 1, pp. 11–25, 2005.

[13] N. Eagle and A. S. Pentland, "Reality mining: Sensing complex social systems," *Personal and ubiquitous computing*, vol. 10, no. 4, pp. 255–268, 2006.

[14] N. Eagle, A. S. Pentland, and D. Lazer, "Inferring friendship network structure by using mobile phone data," *Proceedings of the national academy of sciences*, vol. 106, no. 36, pp. 15 274–15 278, 2009.

[15] L. Elliott and Y. W. Teh, "Scalable imputation of genetic data with a discrete fragmentation-coagulation process," in *Advances in Neural Information Processing Systems*, 2012, pp. 2852–2860.

[16] X. Fan, L. Cao, and R. Y. Da Xu, "Dynamic infinite mixed-membership stochastic blockmodel," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 9, pp. 2072–2085, 2014.

[17] X. Fan, R. Y. Da Xu, and L. Cao, "Copula mixed-membership stochastic blockmodel.," in *IJCAI*, 2016, pp. 1462–1468.

[18] X. Fan, B. Li, C. Li, S. SIsson, and L. Chen, "Scalable deep generative relational model with high-order node dependence," in *Advances in Neural Information Processing Systems*, 2019, pp. 12 637–12 647.

[19] X. Fan, B. Li, S. A. Sisson, C. Li, and L. Chen, "Scalable deep generative relational model with high-order node dependence," in *NeurIPS*, 2019.

[20] T. S. Ferguson, "A bayesian analysis of some nonparametric problems," *The annals of statistics*, pp. 209–230, 1973.

[21] J. Foulds, C. DuBois, A. Asuncion, C. Butts, and P. Smyth, "A dynamic relational infinite feature model for longitudinal social networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 287–295.

[22] W. Fu, L. Song, and E. P. Xing, "Dynamic mixed membership blockmodel for evolving networks," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 329–336.

[23] A. Godoy-Lorite, R. Guimerà, C. Moore, and M. Sales-Pardo, "Accurate and scalable social recommendation using mixed-membership stochastic block models," *Proceedings of the National Academy of Sciences*, vol. 113, no. 50, pp. 14 207–14 212, 2016.

[24] A. Goldenberg, A. X. Zheng, S. E. Fienberg, E. M. Airoldi, *et al.*, "A survey of statistical network models," *Foundations and Trends® in Machine Learning*, vol. 2, no. 2, pp. 129–233, 2010.

[25] P. K. Gopalan, S. Gerrish, M. Freedman, D. M. Blei, and D. M. Mimno, "Scalable inference of overlapping communities," in *Advances in Neural Information Processing Systems*, 2012, pp. 2249–2257.

[26] P. Gopalan, F. J. Ruiz, R. Ranganath, and D. Blei, "Bayesian nonparametric poisson factorization for recommendation systems," in *Artificial Intelligence and Statistics*, 2014, pp. 275–283.

[27] T. L. Griffiths and Z. Ghahramani, “The indian buffet process: An introduction and review,” *Journal of Machine Learning Research*, vol. 12, no. Apr, pp. 1185–1224, 2011.

[28] D. K. Hammond, P. Vandergheynst, and R. Gribonval, “Wavelets on graphs via spectral graph theory,” *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[29] Q. Ho, A. Parikh, L. Song, and E. Xing, “Multiscale community blockmodel for network exploration,” in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 333–341.

[30] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[31] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, “Stochastic variational inference,” *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303–1347, 2013.

[32] D. N. Hoover, “Relations on probability spaces and arrays of random variables,” *Preprint, Institute for Advanced Study, Princeton, NJ*, vol. 2, 1979.

[33] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities,” *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[34] C. Hu, P. Rai, and L. Carin, “Non-negative matrix factorization for discrete data with hierarchical side-information,” in *Artificial Intelligence and Statistics*, 2016, pp. 1124–1132.

[35] ——, “Deep generative models for relational data with side information,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, 2017, pp. 1578–1586.

[36] L. Isella, J. Stehlé, A. Barrat, C. Cattuto, J.-F. Pinton, and W. Van den Broeck, “What’s in a crowd? analysis of face-to-face behavioral networks,” *Journal of theoretical biology*, vol. 271, no. 1, pp. 166–180, 2011.

[37] K. Ishiguro, T. Iwata, N. Ueda, and J. B. Tenenbaum, “Dynamic infinite relational model for time-varying relational data analysis,” in *Advances in Neural Information Processing Systems*, 2010, pp. 919–927.

[38] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda, “Learning systems of concepts with an infinite relational model,” 2006.

[39] D. I. Kim, M. Hughes, and E. Sudderth, “The nonparametric metadata dependent relational model,” *arXiv preprint arXiv:1206.6414*, 2012.

[40] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[41] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[42] T. Kohonen, *Self-organization and associative memory*. Springer Science & Business Media, 2012, vol. 8.

[43] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 426–434.

[44] ——, "Collaborative filtering with temporal dynamics," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2009, pp. 447–456.

[45] J. Koutnik, K. Greff, F. Gomez, and J. Schmidhuber, "A clockwork rnn," *arXiv preprint arXiv:1402.3511*, 2014.

[46] P.-S. Koutsourelakis and T. Eliassi-Rad, "Finding mixed-memberships in social networks.," 2008.

[47] E. Lazega *et al.*, *The collegial phenomenon: The social mechanisms of cooperation among peers in a corporate law partnership*. Oxford University Press on Demand, 2001.

[48] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 2, 2007.

[49] B. Li, X. Zhu, R. Li, C. Zhang, X. Xue, and X. Wu, "Cross-domain collaborative filtering over time," in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[50] S. Li, Y. Xie, H. Dai, and L. Song, "M-statistic for kernel change-point detection," in *Advances in Neural Information Processing Systems*, 2015, pp. 3366–3374.

[51] S. Li, Y. Xie, M. Farajtabar, A. Verma, and L. Song, "Detecting changes in dynamic events over networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 2, pp. 346–359, 2017.

[52] W. Li, S. Ahn, and M. Welling, "Scalable mcmc for mixed membership stochastic blockmodels," in *Artificial Intelligence and Statistics*, 2016, pp. 723–731.

[53] S. Linderman and R. Adams, "Discovering latent network structure in point process data," in *International Conference on Machine Learning*, 2014, pp. 1413–1421.

[54] D. Linzner, M. Schmidt, and H. Koeppl, "Scalable structure learning of continuous-time bayesian networks from incomplete data," in *Advances in Neural Information Processing Systems*, 2019, pp. 3741–3751.

[55] W. Liu, P.-Y. Chen, S. Yeung, T. Suzumura, and L. Chen, “Principled multilayer network embedding,” in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, IEEE, 2017, pp. 134–141.

[56] L. Luo, B. Li, I. Koprinska, S. Berkovsky, and F. Chen, “Tracking the evolution of customer purchase behavior segmentation via a fragmentation-coagulation process.,” in *IJCAI*, 2017, pp. 2414–2420.

[57] K. Miller, M. I. Jordan, and T. L. Griffiths, “Nonparametric latent feature models for link prediction,” in *Advances in neural information processing systems*, 2009, pp. 1276–1284.

[58] X. Miscouridou, F. Caron, and Y. W. Teh, “Modelling sparsity, heterogeneity, reciprocity and community structure in temporal interaction data,” in *Advances in Neural Information Processing Systems*, 2018, pp. 2343–2352.

[59] A. Mnih and R. R. Salakhutdinov, “Probabilistic matrix factorization,” in *Advances in neural information processing systems*, 2008, pp. 1257–1264.

[60] M. Mørup and M. N. Schmidt, “Bayesian community detection,” *Neural computation*, vol. 24, no. 9, pp. 2434–2456, 2012.

[61] M. Mørup, M. N. Schmidt, and L. K. Hansen, “Infinite multiple membership relational modeling for complex networks,” in *2011 IEEE International Workshop on Machine Learning for Signal Processing*, IEEE, 2011, pp. 1–6.

[62] M. E. Newman, “Clustering and preferential attachment in growing networks,” *Physical review E*, vol. 64, no. 2, p. 025 102, 2001.

[63] K. Palla, D. Knowles, and Z. Ghahramani, “An infinite latent attribute model for network data,” *arXiv preprint arXiv:1206.6416*, 2012.

[64] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2014, pp. 701–710.

[65] N. G. Polson, J. G. Scott, and J. Windle, “Bayesian inference for logistic models using pólya–gamma latent variables,” *Journal of the American statistical Association*, vol. 108, no. 504, pp. 1339–1349, 2013.

[66] I. Porteous, E. Bart, and M. Welling, “Multi-hdp: A non parametric bayesian model for tensor factorization.,” in *Aaai*, vol. 8, 2008, pp. 1487–1490.

[67] P. Rai, C. Hu, R. Henao, and L. Carin, “Large-scale bayesian multi-label learning via topic-based label embeddings,” in *Advances in Neural Information Processing Systems*, 2015, pp. 3222–3230.

[68] V. Rao and Y. W. Teh, "Fast mcmc sampling for markov jump processes and continuous time bayesian networks," *arXiv preprint arXiv:1202.3760*, 2012.

[69] F. Rosenblatt, "Principles of neurodynamics. perceptrons and the theory of brain mechanisms," Cornell Aeronautical Lab Inc Buffalo NY, Tech. Rep., 1961.

[70] N. Rosenfeld, O. Meshi, D. Tarlow, and A. Globerson, "Learning structured models with the auc loss and its generalizations," in *Artificial Intelligence and Statistics*, 2014, pp. 841–849.

[71] A. Schein, J. Paisley, D. M. Blei, and H. Wallach, "Bayesian poisson tensor factorization for inferring multilateral relations from sparse dyadic event counts," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2015, pp. 1045–1054.

[72] A. Schein, M. Zhou, D. M. Blei, and H. Wallach, "Bayesian poisson tucker decomposition for learning the structure of international relations," *arXiv preprint arXiv:1606.01855*, 2016.

[73] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *European Semantic Web Conference*, Springer, 2018, pp. 593–607.

[74] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[75] X. Tan, V. Rao, and J. Neville, "The indian buffet hawkes process to model evolving latent influences.," in *UAI*, 2018, pp. 795–804.

[76] M. Tarrés-Deulofeu, A. Godoy-Lorite, R. Guimerà, and M. Sales-Pardo, "Tensorial and bipartite block models for link prediction in layered networks and temporal networks," *Physical Review E*, vol. 99, no. 3, p. 032 307, 2019.

[77] Y. W. Teh, C. Blundell, and L. Elliott, "Modelling genetic variations using fragmentation-coagulation processes," in *Advances in neural information processing systems*, 2011, pp. 819–827.

[78] Y. W. Teh and V. Rao, "Gaussian process modulated renewal processes," in *Advances in Neural Information Processing Systems*, 2011, pp. 2474–2482.

[79] Y. W. Teh, D. Grür, and Z. Ghahramani, "Stick-breaking construction for the indian buffet process," in *Artificial Intelligence and Statistics*, 2007, pp. 556–563.

[80] C. Wang and D. M. Blei, “Variational inference for the nested chinese restaurant process,” in *Advances in Neural Information Processing Systems*, 2009, pp. 1990–1998.

[81] C. Wang, J. Paisley, and D. M. Blei, “Online variational inference for the hierarchical dirichlet process,” in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, in PMLR*, 2011.

[82] H. Wang, X. Shi, and D.-Y. Yeung, “Relational deep learning: A deep latent variable model for link prediction.,” 2017.

[83] E. P. Xing, W. Fu, L. Song, *et al.*, “A state-space mixed membership blockmodel for dynamic network tomography,” *The Annals of Applied Statistics*, vol. 4, no. 2, pp. 535–566, 2010.

[84] J. Yang, V. Rao, and J. Neville, “Decoupling homophily and reciprocity with latent space network models.,” in *UAI*, 2017.

[85] S. Yang and H. Koeppl, “A poisson gamma probabilistic model for latent node-group memberships in dynamic networks,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[86] ——, “Dependent relational gamma process models for longitudinal networks,” in *International Conference on Machine Learning*, 2018, pp. 5551–5560.

[87] K. Yao, T. Cohn, K. Vylomova, and K. Duh, “Depth-gated recurrent neural networks [j],” *arXiv preprint arXiv:1508.03790*, vol. 9, 2015.

[88] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, “Graph convolutional neural networks for web-scale recommender systems,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 974–983.

[89] M. Zaheer, A. Ahmed, and A. J. Smola, “Latent lstm allocation: Joint clustering and non-linear dynamic modeling of sequence data,” in *International Conference on Machine Learning*, 2017, pp. 3967–3976.

[90] H. Zhang, L. Qiu, L. Yi, and Y. Song, “Scalable multiplex network embedding.,” in *IJCAI*, vol. 18, 2018, pp. 3082–3088.

[91] S. Zhe and Y. Du, “Stochastic nonparametric event-tensor decomposition,” in *Advances in Neural Information Processing Systems*, 2018, pp. 6856–6866.

[92] M. Zhou, “Infinite edge partition models for overlapping community detection and link prediction,” in *Artificial intelligence and statistics*, 2015, pp. 1135–1143.

Appendix A

MCMC Inference for Fragmentation Coagulation Based Mixed Membership Stochastic Blockmodel

Fragmentation Coagulation Based Mixed Membership Stochastic Blockmodel (fcMMSB) is intractable for exact inference. Instead, a Gibbs sampling scheme is derived for posterior distribution with parameters $\boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{B}, \boldsymbol{Q}, \boldsymbol{g}$ and $\boldsymbol{\sigma}$. (Note: The shorthand "–" denotes all other variables in conditionals for sampling. The joint distribution with the above parameters can be expressed as:

$$
\begin{aligned}
\Pr(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{B}, \boldsymbol{Q}, \boldsymbol{g}, \boldsymbol{\sigma} | -) = & \prod_{i=1}^{N} \prod_{j=1}^{N} \prod_{t=0}^{T-1} \Pr(x_{ij}^{t} | z_i^t, z_j^t, Q_{g_{i\to j}^t}, \mathrm{B}_{g_{i\to j}^t g_{i\leftarrow j}^t}, \epsilon) \\
& \prod_{i=1}^{N} \prod_{t=0}^{T-1} \Pr(z_i^t | \zeta) \prod_{i=1}^{N} \prod_{t'=0}^{T-2} \Pr(z_i^{t'} | \eta) \\
& \prod_{i=1}^{N} \prod_{j=1}^{N} \prod_{t=0}^{T-1} \Pr(g_{i\to j}^t | \theta_i^t) \Pr(g_{i\leftarrow j}^t | \theta_j^t) \prod_{i=1}^{N} \prod_{t=0}^{T-1} \Pr(\theta_i^t | \alpha) \\
& \prod_{k=1}^{K} \Pr(Q_k | \mu_Q, \sigma_Q) \Pr(\mathrm{B}_{kk} | \mu_{\mathrm{B}}, \sigma_{\mathrm{B}}) \\
& \prod_{k=1}^{K} \prod_{l=k+1}^{K} \Pr(\mathrm{B}_{kl}, \mathrm{B}_{lk} | \mu_{kl}, \mu_{lk}, \boldsymbol{\sigma}_{kl}) \Pr(\boldsymbol{\sigma}_{kl} | \varphi, \psi)
\end{aligned}
$$

The full procedure of sampling is summarized in Algorithm 6.

A.1 Sampling $B_{lk}, B_{kl}(l \neq k)$

The polya-gamma distribution is used for data augmentation of the inference of $[B_{lk}, B_{kl}]$. The main result of polya-gamma data augmentation is introduced here. For more details, refer to [65].

We say a random variable ω follows a polya-gamma distribution, denoted as $\omega \sim \mathrm{PG}(b, c)$, if

$$\omega = \frac{1}{2\pi^2} \sum_{d=1}^{\infty} \frac{g_d}{(d - 1/2)^2 + c^2/(4\pi)^2}$$

where the variables g_d are independent and follow $\mathrm{Ga}(b, 1)$ gamma distribution where $b > 0$ and $c \in \mathcal{R}$. A fundamental theorem is described as following:

Theorem 4. *The binomial likelihood can be expressed as:*

$$\frac{(e^{\phi})^m}{(1 + e^{\phi})^n} = 2^{-n} e^{\kappa\phi} \int e^{-\omega\phi^2/2} p(\omega)\, d\omega$$

where $\omega \sim PG(n, 0)$, $\kappa = m - n/2$. Furthermore, having $\omega \sim PG(n, 0)$ with conditional distribution $p(w|\phi)$, $\frac{e^{-\omega\phi^2/2} p(\omega)}{\int e^{-\omega\phi^2/2} p(\omega) d\omega}$,

$$\omega|\phi \sim PG(n, \phi)$$

This theorem gives the intuition of how the inference of the binomial likelihood parametrized by the logistic transformation of variables can be simplified. Assuming that the prior of ϕ follows a Gaussian distribution $\mathcal{N}(\mu, \sigma)$ with binomial likelihood $\frac{(e^{\phi})^m}{(1+e^{\phi})^n}$, the posterior distribution of ϕ can be expressed as:

$$\begin{aligned} p(\phi|-) &\propto \frac{(e^{\phi})^m}{(1 + e^{\phi})^n} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{\phi-\mu}{\sigma})^2} \\ &\propto 2^{-n} e^{\kappa\phi} \int e^{-\omega\phi^2/2} p(\omega)\, d\omega\, e^{-\frac{1}{2}(\frac{\phi-\mu}{\sigma})^2} \\ &\propto e^{\kappa\phi} \int e^{-\omega\phi^2/2} p(\omega)\, d\omega\, e^{-\frac{1}{2}(\frac{\phi-\mu}{\sigma})^2} \end{aligned}$$

If the ω is induced as a auxiliary variable following $\mathrm{PG}(n, \phi)$, the posterior distribution of ϕ can be simplified as:

$$
\begin{aligned}
p(\phi|-) &\propto e^{\kappa\phi}e^{-\omega\phi^2/2}e^{-\frac{1}{2}(\frac{\phi-\mu}{\sigma})^2} \\
&\propto e^{-\frac{1}{2}\frac{(1+w\sigma^2)\phi^2-2(\mu+\kappa\sigma^2)\phi}{\sigma^2}} \\
&\propto e^{-\frac{1}{2}\frac{\phi^2-2\frac{\mu+\kappa\sigma^2}{1+w\sigma^2}\phi}{\sigma^2(1+w\sigma^2)}} \\
&\propto e^{-\frac{1}{2}\frac{(\phi-\frac{\mu+\kappa\sigma^2}{1+w\sigma^2})^2}{\sigma^2(1+w\sigma^2)}}
\end{aligned}
$$

which is a Gaussian distribution with $\mathcal{N}(\frac{\mu+\kappa\sigma^2}{1+w\sigma^2}, (\sigma^2(1+w\sigma^2))^{\frac{1}{2}})$. Therefore, the true posterior of ϕ can be derived by updating ϕ and ω alternately.

Now turn to the inference of B_{lk}, B_{kl}. For simplicity, B_{lk}, B_{kl} are concatenated, denoted as $\hat{\boldsymbol{B}}$ while μ_{kl}, μ_{lk} as $\boldsymbol{\mu}_{kl}$ Besides, the observation x_{ij}^t related to $\hat{\boldsymbol{B}}$ is reparamterized by a vector $\boldsymbol{x}_{ij}^t$, expressed as: $[x_{ij,0}^t, x_{ij,1}^t]$, where

$$
\begin{aligned}
x_{ij,0}^t &= \mathbb{I}[x_{ij}^t = 1]\mathbb{I}[g_{i\to j}^t = k]\mathbb{I}[g_{i\leftarrow j}^t = l] \\
x_{ij,1}^t &= \mathbb{I}[x_{ij}^t = 1]\mathbb{I}[g_{i\to j}^t = l]\mathbb{I}[g_{i\leftarrow j}^t = k]
\end{aligned}
$$

Now the posterior distribution of $\hat{\boldsymbol{B}}$ can be described as:

$$
\begin{aligned}
\Pr(\hat{\boldsymbol{B}}|-) \propto & \prod_{i=1}^{N}\prod_{j=1}^{N}\prod_{t=0}^{T-1} \Pr(x_{ij}^t|z_i^t, z_j^t, Q_{g_{i\to j}^t}, \mathrm{B}_{g_{i\to j}^t g_{i\leftarrow j}^t}, \epsilon)\Pr(\hat{\boldsymbol{B}}|\boldsymbol{\mu}_{kl}, \boldsymbol{\sigma}_{kl}) \\
= & \prod_{i=1}^{N}\prod_{j=1}^{N}\prod_{t=0}^{T-1} \Pr(x_{ij}^t|\hat{\boldsymbol{B}})^{\mathbb{I}[g_{i\to j}^t=k]\mathbb{I}[g_{i\leftarrow j}^t=l]+\mathbb{I}[g_{i\to j}^t=l]\mathbb{I}[g_{i\leftarrow j}^t=k]} \\
& \Pr(\hat{\boldsymbol{B}}|\boldsymbol{\mu}_{kl}, \boldsymbol{\sigma}_{kl}) \\
\propto & \prod_{i,j,t} \frac{e^{\boldsymbol{x}_{ij}^{t}{}^{\intercal}\hat{\boldsymbol{B}}}}{1+e^{\boldsymbol{x}_{ij}^{t}{}^{\intercal}\hat{\boldsymbol{B}}}} e^{-\frac{1}{2}(\hat{\boldsymbol{B}}-\boldsymbol{\mu}_{kl})^{\intercal}\boldsymbol{\sigma}_{kl}{}^{-1}(\hat{\boldsymbol{B}}-\boldsymbol{\mu}_{kl})} \\
= & \frac{[e^{B_{kl}}]^{n_{kl}^1}}{[1+e^{B_{kl}}]^{n_{kl}}} \frac{[e^{B_{lk}}]^{n_{lk}^1}}{[1+e^{B_{lk}}]^{n_{lk}}} e^{-\frac{1}{2}(\hat{\boldsymbol{B}}-\boldsymbol{\mu}_{kl})^{\intercal}\boldsymbol{\sigma}_{kl}{}^{-1}(\hat{\boldsymbol{B}}-\boldsymbol{\mu}_{kl})} \\
= & \frac{[e^{\mathbf{v}_0^{\intercal}\hat{\boldsymbol{B}}}]^{n_{kl}^1}}{[1+e^{\mathbf{v}_0^{\intercal}\hat{\boldsymbol{B}}}]^{n_{kl}}} \frac{[e^{\mathbf{v}_1^{\intercal}\hat{\boldsymbol{B}}}]^{n_{lk}^1}}{[1+e^{\mathbf{v}_1^{\intercal}\hat{\boldsymbol{B}}}]^{n_{lk}}} e^{-\frac{1}{2}(\hat{\boldsymbol{B}}-\boldsymbol{\mu}_{kl})^{\intercal}\boldsymbol{\sigma}_{kl}{}^{-1}(\hat{\boldsymbol{B}}-\boldsymbol{\mu}_{kl})} \\
\propto & e^{\kappa_0\mathbf{v}_0^{\intercal}\hat{\boldsymbol{B}}} e^{-\omega_0(\mathbf{v}_0^{\intercal}\hat{\boldsymbol{B}})^2/2} e^{\kappa_1\mathbf{v}_1^{\intercal}\hat{\boldsymbol{B}}} e^{-\omega_1(\mathbf{v}_1^{\intercal}\hat{\boldsymbol{B}})^2/2} e^{-\frac{1}{2}(\hat{\boldsymbol{B}}-\boldsymbol{\mu}_{kl})^{\intercal}\boldsymbol{\sigma}_{kl}{}^{-1}(\hat{\boldsymbol{B}}-\boldsymbol{\mu}_{kl})} \\
\propto & e^{-\frac{\omega_0}{2}(\mathbf{v}_0^{\intercal}\hat{\boldsymbol{B}}-\kappa_0/\omega_0)^2} e^{-\frac{\omega_1}{2}(\mathbf{v}_1^{\intercal}\hat{\boldsymbol{B}}-\kappa_1/\omega_1)^2} e^{-\frac{1}{2}(\hat{\boldsymbol{B}}-\boldsymbol{\mu}_{kl})^{\intercal}\boldsymbol{\sigma}_{kl}{}^{-1}(\hat{\boldsymbol{B}}-\boldsymbol{\mu}_{kl})} \\
= & e^{-\frac{1}{2}(\mathbf{v}\hat{\boldsymbol{B}}-\hat{\boldsymbol{\kappa}})^{\intercal}\hat{\boldsymbol{\sigma}}^{-1}(\mathbf{v}\hat{\boldsymbol{B}}-\hat{\boldsymbol{\kappa}})} e^{-\frac{1}{2}(\hat{\boldsymbol{B}}-\boldsymbol{\mu}_{kl})^{\intercal}\boldsymbol{\sigma}_{kl}{}^{-1}(\hat{\boldsymbol{B}}-\boldsymbol{\mu}_{kl})} \\
= & e^{-\frac{1}{2}(\hat{\boldsymbol{B}}-\hat{\boldsymbol{\kappa}})^{\intercal}\hat{\boldsymbol{\sigma}}^{-1}(\hat{\boldsymbol{B}}-\hat{\boldsymbol{\kappa}})} e^{-\frac{1}{2}(\hat{\boldsymbol{B}}-\boldsymbol{\mu}_{kl})^{\intercal}\boldsymbol{\sigma}_{kl}{}^{-1}(\hat{\boldsymbol{B}}-\boldsymbol{\mu}_{kl})} \\
\propto & e^{-\frac{1}{2}(\hat{\boldsymbol{B}}^{\intercal}\hat{\boldsymbol{\sigma}}^{-1}\hat{\boldsymbol{B}}-\hat{\boldsymbol{B}}^{\intercal}\hat{\boldsymbol{\sigma}}^{-1}\hat{\boldsymbol{\kappa}}-\hat{\boldsymbol{\kappa}}^{\intercal}\hat{\boldsymbol{\sigma}}^{-1}\hat{\boldsymbol{B}}+\hat{\boldsymbol{B}}^{\intercal}\boldsymbol{\sigma}_{kl}{}^{-1}\hat{\boldsymbol{B}}-\hat{\boldsymbol{B}}^{\intercal}\boldsymbol{\sigma}_{kl}{}^{-1}\boldsymbol{\mu}_{kl}-\boldsymbol{\mu}_{kl}\boldsymbol{\sigma}_{kl}{}^{-1}\hat{\boldsymbol{B}})} \\
= & e^{-\frac{1}{2}[\hat{\boldsymbol{B}}^{\intercal}(\hat{\boldsymbol{\sigma}}^{-1}+\boldsymbol{\sigma}_{kl}{}^{-1})\hat{\boldsymbol{B}}-\hat{\boldsymbol{B}}^{\intercal}(\hat{\boldsymbol{\sigma}}^{-1}\hat{\boldsymbol{\kappa}}+\boldsymbol{\sigma}_{kl}{}^{-1}\boldsymbol{\mu}_{kl})-(\hat{\boldsymbol{\kappa}}^{\intercal}\hat{\boldsymbol{\sigma}}^{-1}+\boldsymbol{\mu}_{kl}\boldsymbol{\sigma}_{kl}{}^{-1})\hat{\boldsymbol{B}}]} \\
\propto & e^{-\frac{1}{2}(\hat{\boldsymbol{B}}-\boldsymbol{\mu}^*)^{\intercal}\boldsymbol{\sigma}^{*-1}(\hat{\boldsymbol{B}}-\boldsymbol{\mu}^*)}
\end{aligned}
$$

Therefore, the posterior distribution of $\hat{\boldsymbol{B}}$ follows $\mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\sigma}^*)$ where

$$
\begin{aligned}
\boldsymbol{\mu}^* &= \boldsymbol{\sigma}^*(\boldsymbol{\kappa}+\boldsymbol{\sigma}_{kl}\boldsymbol{\mu}_{kl}) \\
\boldsymbol{\sigma}^* &= (\hat{\boldsymbol{\sigma}}^{-1}+\boldsymbol{\sigma}_{kl}{}^{-1})^{-1}
\end{aligned}
$$

where $\mathbf{v}_0 = [1\ 0]^{\intercal}, \mathbf{v}_1 = [0\ 1]^{\intercal}$ and $\mathbf{v}$ is a two dimensional identity matrix. $\hat{\boldsymbol{\kappa}} = [\kappa_0/\omega_0\ \kappa_1/\omega_1]^{\intercal}$, $\boldsymbol{\kappa} = [\kappa_0\ \kappa_1]^{\intercal}$, $\kappa_0 = n_{kl}^1 - n_{kl}/2$, $\kappa_1 = n_{lk}^1 - n_{lk}/2$, and $\hat{\boldsymbol{\sigma}}^{-1} = \mathrm{diag}(\omega_0, \omega_1)$. ω_0 and ω_1 follow $\mathrm{PG}(n_{kl}, B_{kl})$ and $\mathrm{PG}(n_{lk}, B_{lk})$ respectively.

$$
\begin{aligned}
n_{lk} &= \sum_{t,i,j} \mathbb{I}[g_{i\to j}^t = l]\cdot\mathbb{I}[g_{i\leftarrow j}^t = k]\cdot\mathbb{I}[z_i^t = z_j^t] \\
n_{lk}^1 &= \sum_{t,i,j} \mathbb{I}[g_{i\to j}^t = l]\cdot\mathbb{I}[g_{i\leftarrow j}^t = k]\cdot\mathbb{I}[z_i^t = z_j^t]\cdot\mathbb{I}[x_{ij}^t = 1]
\end{aligned}
$$

A.2 Sampling B_{kk} and Q_k

The sampling scheme of B_{kk} and Q_k is similar with $\hat{\boldsymbol{B}}$. For simplicity, B_{kk}, Q_k are also concatenated, denoted as $\hat{\boldsymbol{Q}}$ while μ_B, μ_Q as $\boldsymbol{\mu}_Q$ and σ_B, σ_Q as $\boldsymbol{\sigma}_Q$ where $\boldsymbol{\sigma}_Q = \mathrm{diag}(\sigma_B, \sigma_Q)$. Now the posterior distribution of $\hat{\boldsymbol{Q}}$ can be described as:

$$\begin{aligned}
\Pr(\hat{\boldsymbol{Q}}|-) \propto & \prod_{i=1}^{N}\prod_{j=1}^{N}\prod_{t=0}^{T-1} \Pr(x_{ij}^t | z_i^t, z_j^t, Q_{g_{i\to j}^t}, \mathrm{B}_{g_{i\to j}^t g_{i\leftarrow j}^t}, \epsilon) \Pr(\hat{\boldsymbol{Q}}|\boldsymbol{\mu}_{kl}, \boldsymbol{\sigma}_{kl}) \\
= & \prod_{i=1}^{N}\prod_{j=1}^{N}\prod_{t=0}^{T-1} \Pr(x_{ij}^t|\hat{\boldsymbol{Q}})^{\mathbb{I}[g_{i\to j}^t = k]\mathbb{I}[g_{i\leftarrow j}^t = k]} \Pr(\hat{\boldsymbol{Q}}|\boldsymbol{\mu}_Q, \boldsymbol{\sigma}_Q) \\
\propto & e^{-\frac{1}{2}(\overline{\mathbf{v}}\hat{\boldsymbol{Q}} - \hat{\boldsymbol{\kappa}})^{\intercal} \hat{\boldsymbol{\sigma}_Q}^{-1} (\overline{\mathbf{v}}\hat{\boldsymbol{Q}} - \hat{\boldsymbol{\kappa}})} e^{-\frac{1}{2}(\hat{\boldsymbol{Q}} - \boldsymbol{\mu}_{kl})^{\intercal} \boldsymbol{\sigma}_Q^{-1} (\hat{\boldsymbol{Q}} - \boldsymbol{\mu}_Q)} \\
\propto & e^{-\frac{1}{2}(\hat{\boldsymbol{Q}} - \boldsymbol{\mu}_Q^*)^{\intercal} \boldsymbol{\sigma}_Q^{*\,-1} (\hat{\boldsymbol{Q}} - \boldsymbol{\mu}_Q^*)}
\end{aligned}$$

So the posterior distribution of $\hat{\boldsymbol{Q}}$ follows $\mathcal{N}(\boldsymbol{\mu}_Q^*, \boldsymbol{\sigma}_Q^*)$:

$$\begin{aligned}
\boldsymbol{\mu}_Q^* &= \boldsymbol{\sigma}_Q^* (\overline{\boldsymbol{v}}^{\intercal} \boldsymbol{\kappa}_Q \overline{\boldsymbol{v}} + \boldsymbol{\sigma}_Q \boldsymbol{\mu}_Q) \\
\boldsymbol{\sigma}^* &= \left(\overline{\boldsymbol{v}}^{\intercal} \hat{\boldsymbol{\sigma}}_Q^{-1} \overline{\boldsymbol{v}} + \boldsymbol{\sigma}_Q{}^{-1}\right)^{-1}
\end{aligned}$$

where $\boldsymbol{\kappa}_Q = [\overline{\kappa_0}\ \overline{\kappa_1}]^{\intercal}$, $\overline{\kappa_0} = m_k^1 - m_k/2$, $\overline{\kappa_1} = \overline{m}_k^1 - \overline{m}_k/2$, and $\hat{\boldsymbol{\sigma}}^{-1} = \mathrm{diag}(\overline{\omega}_0, \overline{\omega}_1)$. $\overline{\omega}_0$ and $\overline{\omega}_1$ follow $\mathrm{PG}(m_k, B_{kk})$ and $\mathrm{PG}(\overline{m}_k, B_{kk} + Q_k)$ respectively. $\overline{\boldsymbol{v}}$ is expressed as:

$$\overline{\boldsymbol{v}} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

Here $m_k, m_k^1, \overline{m}_k$ and $\overline{m}_k^1$ are expressed as:

$$\begin{aligned}
m_k &= \sum_{i,j,t} \mathbb{I}[g_{i\to j}^t = k] \cdot \mathbb{I}[g_{i\leftarrow j}^t = k] \cdot \mathbb{I}[z_i^t = z_j^t] \\
m_k^1 &= \sum_{i,j,t} \mathbb{I}[g_{i\to j}^t = k] \cdot \mathbb{I}[g_{i\leftarrow j}^t = k] \cdot \mathbb{I}[z_i^t = z_j^t] \cdot \mathbb{I}[x_{ij}^t = 1] \\
\overline{m}_k &= \sum_{i,j,t} \mathbb{I}[g_{i\to j}^t = k] \cdot \mathbb{I}[g_{i\leftarrow j}^t = k] \cdot \mathbb{I}[z_i^t \neq z_j^t] \\
\overline{m}_k^1 &= \sum_{i,j,t} \mathbb{I}[g_{i\to j}^t = k] \cdot \mathbb{I}[g_{i\leftarrow j}^t = k] \cdot \mathbb{I}[z_i^t \neq z_j^t] \cdot \mathbb{I}[x_{ij}^t = 1]
\end{aligned}$$

A.3 Sampling $g^t_{i\to j}$

Collapsed Gibbs sampling is used on $g^t_{i\to j}$ by marginalizing over θ^t_i. Here we assume that $g_{i\leftarrow j} = l$. The posterior distribution of $g^t_{i\to j}$ can be expressed as:

$$\begin{aligned}
\Pr(g^t_{i\to j} = k|-) \propto & \Pr(x^t_{ij}|z^t_i, z^t_j, Q_{g^t_{i\to j}}, \mathrm{B}_{g^t_{i\to j} g^t_{i\leftarrow j}}, \epsilon)\Pr(g^t_{i\to j} = k|\boldsymbol{g}_{i,\neg[i\to j]}) \\
= & \frac{[e^{y^t_{ij}}]^{\mathbb{I}[x^t_{ij}=1]}}{1+e^{y^t_{ij}}} \int \Pr(g^t_{i\to j} = k|\theta^t_i)\Pr(\theta^t_i|\boldsymbol{g}^t_{i,\neg[i\to j]}, \alpha)\, d\theta^t_i
\end{aligned}$$

where $\boldsymbol{g}^t_{i,\neg[i\to j]}$ represents and y^t_{ij} is expressed as:

$$\begin{aligned}
y^t_{ij} = & [\mathbb{I}[z^t_i = z^t_j]\mathbf{B}_{lk} + \mathbb{I}[l = k]\mathbb{I}[z^t_i \neq z^t_j](\mathbf{B}_{lk} + Q_k) + \mathbb{I}[l \neq k]\mathbb{I}[z^t_i \neq z^t_j]\epsilon \\
& \mathbb{I}[g^t_{i\to j} = k]\mathbb{I}[g^t_{i\leftarrow j} = l]
\end{aligned}$$

The second part in the integrity can be calculated as:

$$\begin{aligned}
\Pr(\theta^t_i|\boldsymbol{g}^t_{i,\neg[i\to j]}, \alpha) \propto & \Pr(\boldsymbol{g}^t_{i,\neg[i\to j]}|\theta^t_i)\Pr(\theta^t_i|\alpha) \\
\propto & \prod_{k=1}^{K} [\theta^t_{i,k}]^{n^k_{i,\neg[i\to j]}(t)} [\theta^t_{i,k}]^{\alpha_k - 1} \\
= & \prod_{k=1}^{K} [\theta_{i,k}]^{n^k_{i,\neg[i\to j]} + \alpha_k - 1}
\end{aligned}$$

Substitute the above equation into the posterior distribution of $g^t_{i\to j}$, then

$$\begin{aligned}
\Pr(\theta^t_i|\boldsymbol{g}^t_{i,\neg[i\to j]}, \alpha) \propto & \frac{[e^{y^t_{ij}}]^{\mathbb{I}[x^t_{ij}=1]}}{1+e^{y^t_{ij}}} \int \theta_{i,k} \prod_{k=1}^{K} [\theta_{i,k}]^{n^k_{i,\neg[i\to j]}(t) + \alpha_k - 1}\, d\theta_i \\
= & \frac{[e^{y^t_{ij}}]^{\mathbb{I}[x^t_{ij}=1]}}{1+e^{y^t_{ij}}} \frac{n^k_{i,\neg[i\to j]}(t) + \alpha_k}{\sum_k n^k_{i,\neg[i\to j]}(t) + \alpha_k}
\end{aligned}$$

where $n^k_{i,\neg[i\to j]}(t) = \sum_{l,l\neq j} \mathbb{I}[g^t_{i\to l} = k] + \sum_l \mathbb{I}[g^t_{i\leftarrow l} = k]$.

A.4 Sampling $\mathbf{z}$

The prior of latent communities sequence $\mathbf{z}$ is:

$$\Pr_{\text{prior}}(\mathbf{z}|\eta, \zeta) = \prod_{i=1}^{N} \prod_{t=0}^{T-1} \Pr(z^t_i|\eta) \prod_{i=1}^{N} \prod_{t'=0}^{T-2} \Pr(z^{t'}_i|\zeta)$$

and the posterior of z_i^t for entity i at time t can be described as:

$$\begin{aligned}\Pr(z_i^t|-) &\propto \Pr(\mathbf{x}_{i\cdot}^t, \mathbf{x}_{\cdot i}^t|-)\Pr(z_i^t|\eta, \boldsymbol{z}_{\neg i}^t, \boldsymbol{z}^{t-1'})\Pr(z_i^{t'}|\boldsymbol{z}^t, \zeta) \\ &= \prod_{j,t} \frac{[e^{y_{ij}^t}]^{\mathbb{I}[x_{ij}^t=1]}}{1+e^{y_{ij}^t}} \cdot \frac{[e^{y_{ji}^t}]^{\mathbb{I}[x_{ji}^t=1]}}{1+e^{y_{ji}^t}} \cdot \Pr(z_i^t|\zeta, \boldsymbol{z}_{\neg i}^t, \boldsymbol{z}^{t-1'})\Pr(z_i^{t'}|\boldsymbol{z}^t, \eta)\end{aligned}$$

where y_{ij}^t follows the previous definition from previous section. Here $\mathbf{x}_{i\cdot}^t = \{x_{ij}^t | j \in \{1, ..., N\}\}$ and $\mathbf{x}_{\cdot i}^t = \{x_{ji}^t | j \in \{1, ..., N\}\}$. $\Pr(z_i^t|\eta, \boldsymbol{z}_{\neg i}^t, \boldsymbol{z}^{t-1'})$ is the distribution of fragmentation process for entity i at time t. Here we assume that entity i belong to community q at time $t - 1'$. $\Pr(z_i^t|\eta, \boldsymbol{z}_{\neg i}^t, \boldsymbol{z}^{t-1'})$ can be described as:

$$\Pr(z_i^t = h|\zeta, \boldsymbol{z}_{\neg i}^t, \boldsymbol{z}^{t'}) = \begin{cases} |\chi_h^t|/(|\chi_q^{t-1'}| + \zeta - 1) & \text{if } \chi_q^{t-1'} \neq \emptyset, \chi_h^t \neq \emptyset \\ \zeta/(|\chi_q^{t-1'}| + \zeta - 1) & \text{if } \chi_q^{t-1'} \neq \emptyset, \chi_h^t = \emptyset \\ 1 & \text{if } \chi_q^{t-1'} = \emptyset, \chi_h^t = \emptyset \\ 0 & \text{otherwise} \end{cases}$$

χ_h^t represents the entities belonging to community h at time t and $|\chi_h^t|$ denotes the number of entities in community h. $\Pr(z_i^{t+1'}|\boldsymbol{z}^t, \eta)$ is the distribution of coagulation process for entity i at time t'. Here we assume that entity i belong to community h at time t. $\Pr(z_i^{t'}|\boldsymbol{z}^t, \eta)$ can be described as:

$$\Pr(z_i^{t'} = e|\boldsymbol{z}^t, \eta) = \begin{cases} 1 & \text{if } \chi_e^{t'} \neq \emptyset, \chi_h^t \neq \emptyset \\ |\Lambda_e^{t'}|/(|\nu^t| + \eta - 1) & \text{if } \chi_e^{t'} \neq \emptyset, \chi_h^t = \emptyset \\ \eta/(|\nu^t| + \eta - 1) & \text{if } \chi_e^{t'} = \emptyset, \chi_h^t = \emptyset \\ 0 & \text{otherwise} \end{cases}$$

where ν^t represents the set of communities at time t, $|\nu^t|$ is the number of communities and Λ_e^t represents the communities at t which belong to the community set with index e at time t', denoted as $\Lambda_e^{t'} = \{\chi_v^t | \chi_v^t \subseteq \chi_e^{t'}\}$.

A.5 Sampling σ_{kl}

The inverse Wishart distribution is chosen as the prior of $\boldsymbol{\sigma}_{kl}$ that $\boldsymbol{\sigma}_{kl} \sim \mathcal{IW}(\upsilon, \varrho)$. The density distribution of inverse Wishart distribution is expressed as following:

$$\Pr(\boldsymbol{\sigma}_{kl}|\upsilon,\varrho)=\frac{|\varrho|^{\upsilon/2}}{2^{\upsilon p/2}\Gamma_p(\frac{\upsilon}{2})}|\boldsymbol{\sigma}_{kl}|^{-(\upsilon+p+1)/2}e^{-\frac{1}{2}\mathrm{tr}(\varrho\boldsymbol{\sigma}_{kl}{}^{-1})}$$

As the prior and likelihood of σ_{kl} are a conjugate pair, we give the posterior of σ_{kl} directly.

$$\begin{aligned}\Pr(\boldsymbol{\sigma}_{kl}|-)\ &\propto\Pr(\hat{\boldsymbol{B}}|\boldsymbol{\mu}_{kl},\boldsymbol{\sigma}_{kl})\Pr(\boldsymbol{\sigma}_{kl}|\upsilon,\varrho)\\&\propto|\boldsymbol{\sigma}_{kl}|^{-\frac{1}{2}}e^{-\frac{1}{2}(\hat{\boldsymbol{B}}-\boldsymbol{\mu}_{kl})^{\intercal}\boldsymbol{\sigma}_{kl}{}^{-1}(\hat{\boldsymbol{B}}-\boldsymbol{\mu}_{kl})}|\boldsymbol{\sigma}_{kl}|^{-(\upsilon+p+1)/2}e^{-\frac{1}{2}\mathrm{tr}(\varrho\boldsymbol{\sigma}_{kl}{}^{-1})}\\&=|\boldsymbol{\sigma}_{kl}|^{-(\upsilon+p+2)/2}e^{-\frac{1}{2}\mathrm{tr}((\hat{\boldsymbol{B}}-\mu_{kl})(\hat{\boldsymbol{B}}-\mu_{kl})^{\intercal}\boldsymbol{\sigma}_{kl}{}^{-1})}e^{-\frac{1}{2}\mathrm{tr}(\varrho\boldsymbol{\sigma}_{kl}{}^{-1})}\\&=|\boldsymbol{\sigma}_{kl}|^{-(\upsilon+p+2)/2}e^{-\frac{1}{2}\mathrm{tr}[((\hat{\boldsymbol{B}}-\mu_{kl})(\hat{\boldsymbol{B}}-\mu_{kl})^{\intercal}+\varrho)\boldsymbol{\sigma}_{kl}{}^{-1}]}\end{aligned}$$

so the posterior distribution of follows

$$\boldsymbol{\sigma}_{kl}|-\sim\mathcal{IW}(1+\upsilon,\varrho+(\hat{\boldsymbol{B}}-\mu_{kl})(\hat{\boldsymbol{B}}-\mu_{kl})^{\intercal})$$

A.6 Prediction

In the previous sections, we derived the samples at each iteration. We would like to use these samples to estimate the unobserved relations. Our prediction target at iteration s, $\hat{x}_{ij}^{t[s]}$, is expressed as:

$$\hat{x}_{ij}^{t[s]}=\hat{\theta}_i^{\mathsf{t}\intercal}\cdot\bar{\mathbf{B}}\cdot\hat{\theta}_j^t$$

where the superscript of $\hat{\theta}_i^{\mathsf{t}\intercal}$ is the transpose of the vector. Here each dimension of θ_i^t and $n_k^i(t)$ are expressed as:

$$\hat{\theta}_i^{t,k}=\frac{n_k^i(t)+\alpha_k}{\sum_k n_k^i(t)+\alpha_k}$$

$$n_k^i(t)=\sum_j\mathbb{I}[g_{i\to j}^t=k]+\sum_j\mathbb{I}[g_{i\leftarrow j}^t=k]$$

Each entry $\bar{\mathbf{B}}_{lk}$ of $\bar{\mathbf{B}}$ is $\frac{1}{1+\exp{(-\bar{\mathbf{Y}}_{lk})}}$ where $\bar{\mathbf{Y}}_{lk}$ is described as:

$$\bar{\mathbf{Y}}_{lk}=\mathbb{I}[z_i^t=z_j^t]\mathbf{B}_{lk}+\mathbb{I}[l=k]\mathbb{I}[z_i^t\neq z_j^t](\mathbf{B}_{lk}+Q_k)+\mathbb{I}[l\neq k]\mathbb{I}[z_i^t\neq z_j^t]\epsilon$$

Algorithm 6 Sampling Algorithm for fcMMSB

```
Input: observations $\boldsymbol{x}$, iterations $D, U$
Initialize the number of groups $K$, fragmentation parameter $\zeta$, coagulation parameter $\eta$, parameter of inverse Wishart distribution $(\upsilon, \varrho)$, $\boldsymbol{\mu}_{kl}, \mu_B, \sigma_B, \mu_Q, \sigma_Q, \alpha$
for $d = 1, ..., D$ do
    for $k = 1, \ldots, K$ do
        for $l = 1 + k, \ldots, K$ do
            for $u = 1, \ldots, U$ do
                sample $B_{kl}, B_{lk}$
                sample $\omega_0, \omega_1$
            end for
            sample $\boldsymbol{\sigma}_{kl}$
        end for
    end for
    for $k = 1, \ldots, K$ do
        for $u = 1, \ldots, U$ do
            sample $B_{kk}, Q_k$
            sample $\overline{\omega}_0, \overline{\omega}_1$
        end for
    end for
    for $i = 1, \ldots, N$ do
        for $j = 1, \ldots, N$ do
            for $t = 0, \ldots, t - 1$ do
                sample $g^t_{i \to j}$
                sample $g^t_{i \leftarrow j}$
            end for
        end for
    end for
    for $i = 1, \ldots, N$ do
        for $t = 0, 0', \ldots, t - 2, t - 2', t - 1$ do
            sample $z^t_i$
        end for
    end for
end for
return $\boldsymbol{z}, \boldsymbol{B}, \boldsymbol{Q}, \boldsymbol{g}, \boldsymbol{\sigma}$
```