STATISTICAL ANALYSIS OF GRAPHS WITH ABRUPT CHANGES

by

Türkan Hamzaoğlu

B.S. Computer Engineering, Boğaziçi University, 2009

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in
Boğaziçi University
2013

ACKNOWLEDGEMENTS

First of all, I would like to thank my advisor, Assoc. Prof. A. Taylan Cemgil, for pointing me in the right direction and sharing his valuable opinion during the course of this study. His courses about Bayesian probability theory and Monte Carlo methods were the main motivations for me to study this area.

I also would like to thank the members of the thesis jury, Assoc. Prof. Haluk Bingöl and Assoc. Prof. Muhittin Mungan for their valuable time and improving comments for making this dissertation better.

In addition, I thank to my family and friends for their support. I especially thank to members of PI-LAB for their nice sense of humour.

This research is supported by BAP unit for their financial support with the project (5723) Statistical Graph Analysis. Scientific and Technical Research Council of Turkey has also provided some support by means of 2210 scholarship.

ABSTRACT

STATISTICAL ANALYSIS OF GRAPHS WITH ABRUPT CHANGES

Graphs are powerful mathematical tools to express relationships between any kind of items in very diverse disciplines. In this work, we worked on stochastic block models and multiple change-point detection problem for graph time series, where number of change points is unknown. Stochastic block models is a branch of clustering algorithms for relational data. We studied bayesian approaches as expectation-maximization (EM), variational expectation-maximization, Monte Carlo methods as Gibbs sampling for analysis of stochastic block models. For time series analysis, we have studied Hidden Markov Models, applied well-known forward-backward algorithm to multiple change point analysis on network series. We have proposed an approximate inference algorithm that combines Monte Carlo approaches and hidden Markov models (forward filtering-backward sampling). In our model, we calculate the forward messages completely, sample a change point from those, calculate the backward message for the sampled changed point, update with the forward message and sample a change point for the previous time step. It continues in this way to the first time step, named backward-sampling. By this way, we have simplified the calculation cost. In addition, it is a motivation to use Monte Carlo methodologies in time series analysis where we can not take integrals easily in order to do exact inference. On experiments we have done on syntheic data, we have seen that our proposed approximate inference algorithm gives results in accordance with exact inference methodology, in detecting multiple change points and category assignments.

ÖZET

ANİ DEĞİŞİMLERİ OLAN ÇİZGELERİN İSTATİSTİKSEL NETWORK ANALİZİ

Çizgeler öğeler arasındaki ilişkileri ifade etmek için herhangi tür öğeler arasındaki ilişkileri ifade etmek için oldukça farklı disiplinlerde kullanılan kuvvetli matematiksel araçlardır. Bu çalışmada, rassal öbek çizgeleri ve rassal öbek çizge serilerinde sayısı bilinmeyen çoklu değişim noktası algılama problemi üzerinde çalıştık. Rassal öbek çizge modeli, ilişkisel veri kümeleme algoritmalarının bir dalıdır. Bu model üzerinde beklenti-en iyileme, değişimsel beklenti-en iyileme gibi Bayesçi metodları ve Gibbs örneklemesi gibi Monte Carlo metodlarını çalıştık. Zaman serisi analizinde, saklı Markov modelleri çalıştık, yaygın olarak kullanılan ileri-geri algoritmasını zaman serilerinde çoklu değişim noktası algılama problemine uyarladık. Monte Carlo yaklaşımlarını ve gizli Markov modellerini (ileri filtreleme-geri örnekleme) birleştiren yaklaşık çıkarım algoritması önerdik. Önerdiğimiz modelimizde, ileri yönlü iletilerin tamamı hesaplanır, son zaman dilimi için hesaplanan ileri yönlü iletilerden değişim noktası örneklenir, örneklenen değişim noktası için geri yönlü iletiler hesaplanır, ileri yönlü iletilerle güncellenerek, bir önceki zaman dilimi için değişim noktası örneklenir. lk zaman dilimine kadar devam eden bu yöntem geri yönde örnekleme olarak isimlendirilir.Bu yöntemle, hesaplama maliyetini düşürülmüş oldu. Ayrıca bu kullanım kolayca integral alamadığımız için tam çıkarım algoritmaları geliştiremediğimiz zaman serileri analizinde Monte Carlo metodolojilerinin kullanımı için örnek bir motivasyondur. Sentetik veri üzerinde yaptığımız deneylerde, önerdiğimiz yaklaşık çıkarımlama algoritmasının değişim noktalarını ve küme atamalarının algılanmasında tam çıkarımlama metodolojisiyle yeterince örtüşen sonuçlar verdiğini gözlemledik.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS/ABBREVIATIONS

SBM	Stochastic Block Model
ERG	Erdös-Réyni-Gilbert
MCMC	Monte Carlo Markov Chain
ERGM	exponential random graph model
EM	Expectation-Maximization
CRP	Chinese Restaurant Process
HMMs	Hidden Markov Models

1. INTRODUCTION

Consider the below graphs that represents the well-known monk data [1]. Turbulence was emerging inside American Catholicism in the late 1960's, and there was a major conflict in this particular monastery toward the end of Sampson's twelve-month study. In Figure 1.1 subfigures show each monk's most-liked three friends at three different times. Can you differentiate any friendship groups? What about changes in those groups by time? Can you make any further guesses with the help of matrices in Figure 1.2. These matrices represent observations, both positive and negative, on four social relations at four different times: Affect, Esteem, Influence, and Sanctioning. Respondents were to give their first, second, and third choices, first on the positive side (e.g., "List in order those three brothers whom you most esteemed"), then on the negative side (e.g., "List in order those three brothers whom you esteemed least") [2]. At the end of the fourth time step, monastery broke up. Beginning with Sampson, an interesting question was to discover the latent groups between monks that can explain the broke up.

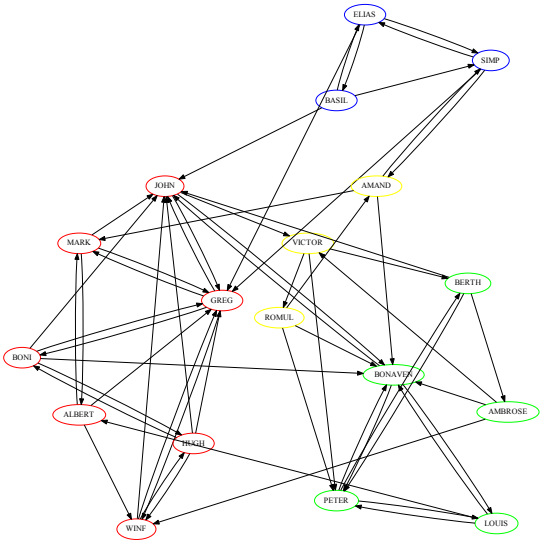
There are two main problems to be solved:

- identifying the subgroups in a network
- detecting abrupt changes that occur to that subgroups by time

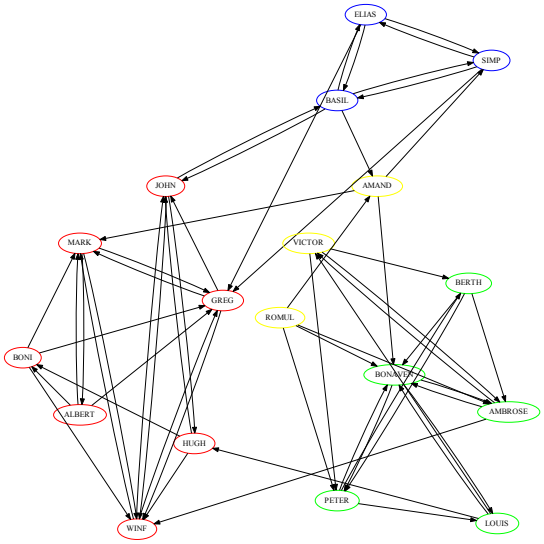
Both of these problems have attracted considerable attention throughout literature, separately.

Another real life problem that has attracted academic attention is the Enron scandal. By examining who-mailed-whom data, structural groups between the workers have been tried to find. In addition, any change in the communication pattern may also reveal some hints of the underlying secret process. [3–5]. Analyzing Enron dataset is very similar to working on terrorist cell identification in computer networks and such other problems [1]

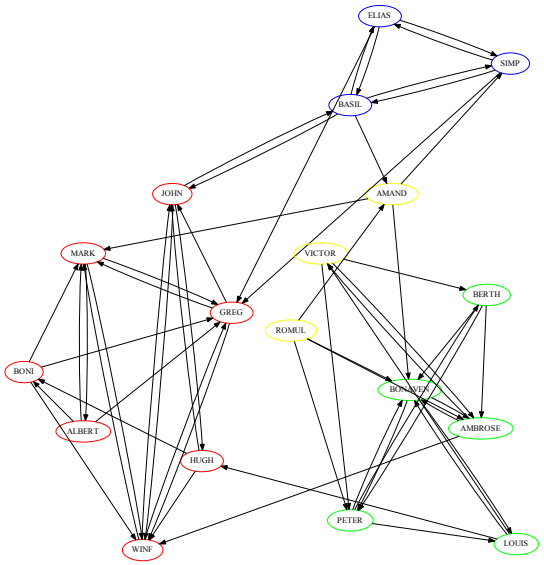
[1] http://www.isi.edu/ adibi/Enron/Enron.htm



(a) time 1



(b) time 2



(c) time 3

Figure 1.1. Like relationship between Monks: Different colors represent the group assignments of monks at each time step.

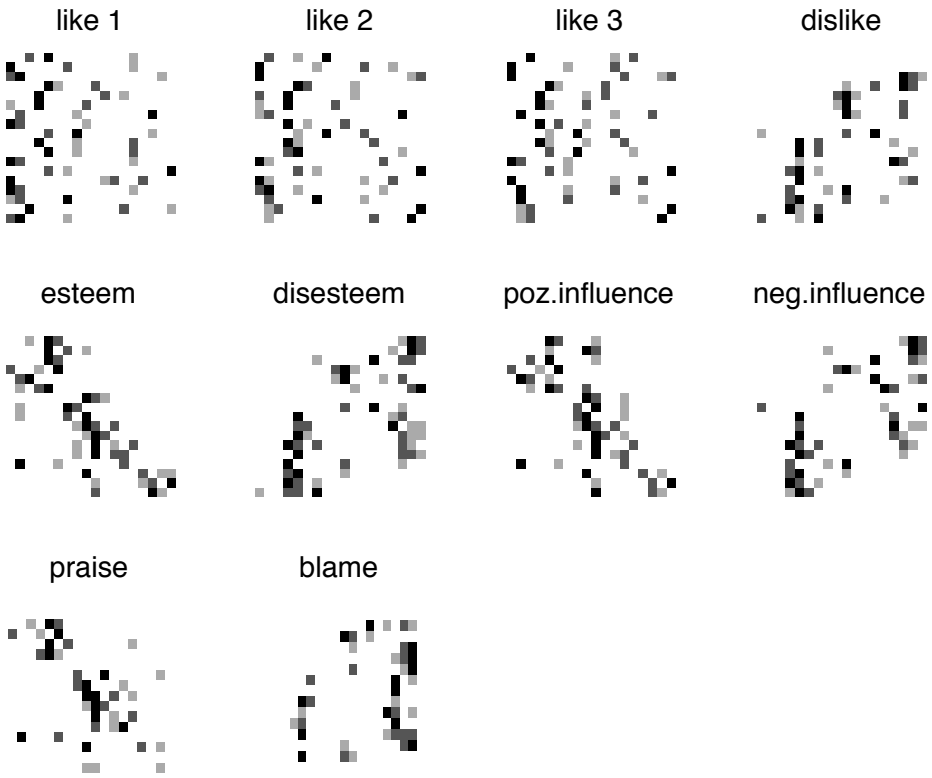


Figure 1.2. Further relations about Monks.

Examine the monthly aggregated adjacency matrices that represents who-mailed-whom information in Figure 1.3.

So far, graphs in examples show relation between same and one type of nodes: relation between monks and mailing relation between Enron workers. But this is not necessary. Nodes can be belong to multiple types. In order to better visualize the problem consider the below synthetic network in Figure 1.4: Assume there are N^u users and N^s services. Each service and user belongs to an underlying category. In the graphic, category assignments are indicated by different colors. Each set of nodes assigned to a category forms a subgroup or a block. Nodes belonging to two categories and edge between them forms a homogenous subgraph. Another assumption is that, edges between nodes depends on the categories that nodes belong to, in other words, category connection relations. In real world, networks are seen as only nodes and edges, category assignments can not be seen directly. We need to infer them somehow. Our first problem can be summarized as, finding the color assignments of nodes, and relation parameters between categories.

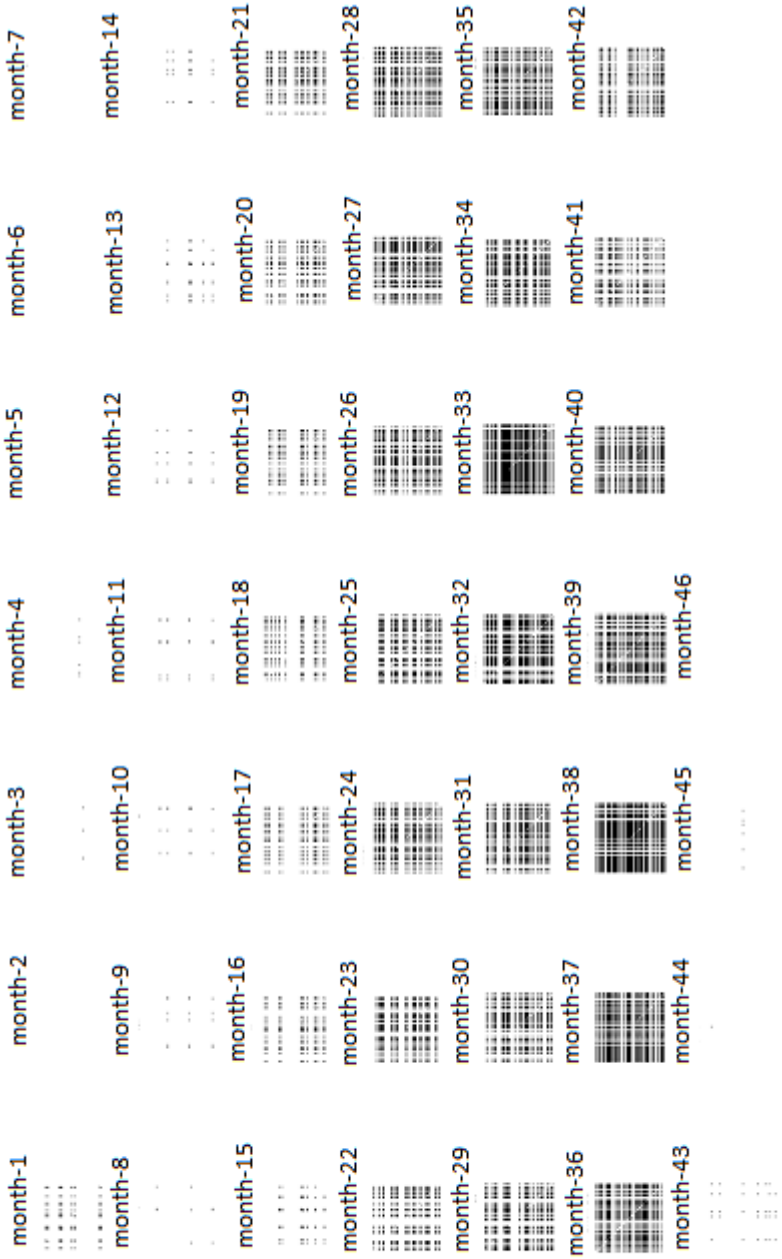


Figure 1.3. Enron monthly aggregated email data.

A subgroup in a network is referred as a *block* in network. Identifying blocks in networks or clustering nodes into blocks has many applications especially in social network analysis. Another description for the job is *modeling relational data*, there is at least one type of item in the network and groups are formed according to relations between them. Another explanatory examples are finding protein classifications according to protein-protein interaction network in computational biology [6], finding hidden groups communication networks, e.g., in detecting possible latent terrorist

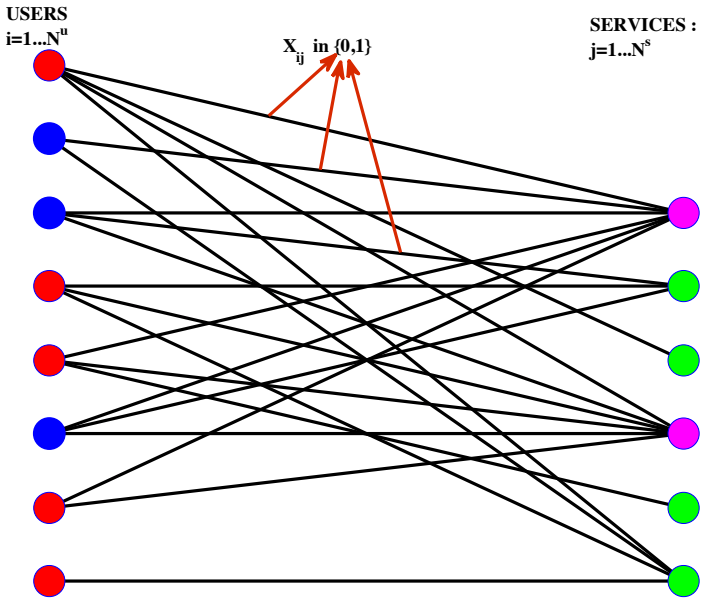


Figure 1.4. An example user and service network: Note the different color assignment of nodes.

cells [7].

Detecting change points in serial data also has exciting applications. For example, when drilling new petroleum well, changes in the well-log data give great information about the rock type and potential oil reservoir size [8]. Another well-known application area is detecting changes in financial areas such as stock markets [9, 10].

In order to visualize an example time series data of stochastic block model (SBM) series examine the below series of SBM graphs in Figure 1.5. Even if the category relations are indicated by gray scale matrices, it is hard to tell exactly when and how many change points are there in the series.

In this work, we will adopt Bayesian approach to deal with detecting underlying block structure in relational data and detecting multiple change points in the underlying structure.

The organization of this thesis as follows: In the rest of this section, our motivation for adopting the Bayesian approach and a literature survey on block models and

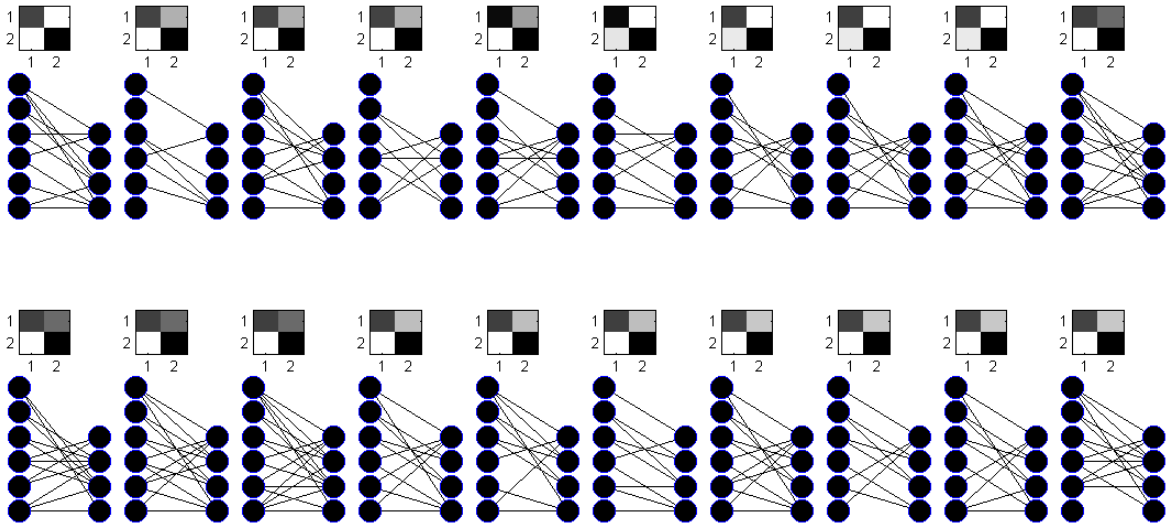


Figure 1.5. An example SBM time series: In real life in addition to category assignments, we do not know the category relations, either.

multiple change point models will be given. In the second section, statistical network models will be explained in three sections: Graphical Models as Mixture Models, Time Series Modeling with Hidden Markov Models and Multiple Change Point Models. In section three experiments will be done on both synthetic and real data sets. In the last section, conclusion will be given.

1.1. MOTIVATION

Finding subgroups in a network is a more general type of clustering problem. Clustering is a popular problem in machine learning field, which means identifying structure in an unlabeled data set by objectively organizing data into homogeneous groups where the within-group-object similarity is minimized and the between-group-object dissimilarity is maximized [11]. One approach in machine learning field is *supervised learning*. In supervised learning, a training set is given with the known values for a problem, cluster assignments in case of a clustering problem. Then the designed prediction methodology is tested with the test data again with the known answers. The closer the prediction is to real values, the better the prediction is.

Another common type of machine learning is *unsupervised learning* which is the

class of problems where one seeks to determine how the data is organized. In unsupervised learning, there is no training data. In clustering problems, trying to be solved by unsupervised learning, one needs to discover groups of input points (in our case nodes) where points are similar to each other in some way than to others outside the groups. For this, some kind of distance or similarity measure is needed.

As it is understood from the definitions of supervised and unsupervised learning, supervised learning is not applicable to *graph clustering*. Graph clustering is grouping the nodes together that are similar. *Assortative* and *disassortative* schemes are used to measure different kinds of similarity between nodes [12]. Then assignment of nodes to clusters can be deterministic, namely *hard clustering* or probabilistic i.e. *soft clustering*.

Another respect of the real life networks is the complexity and uncertainty. Therefore, it is natural to model them with probabilistic (random) models. So probability theory comes to scene. Probability theory allows us to make optimal predictions given all the information available to us, even though that information may be incomplete or ambiguous [13]. Probability theory has two branches: *frequentist approach* and *Bayesian approach*. In both approaches, likelihood of the given data under the parameters that are unknown plays a central role. But this is used different in the two approaches.

In frequentist probability, uncertainty in the problem is involved through noisy measurements and finite size of data set. Therefore, unknown parameters are assumed to be fixed and their values are determined by an *estimator*, e.g., maximum likelihood estimator. In contrast, in Bayesian approach, uncertainty is also involved by the parameters and they are assumed to come from an underlying probability distribution [13]. More precisely, *a priori* distribution of the parameter is involved to the likelihood to obtain *a posteriori* distribution of the parameter given the dataset. When a point estimator is needed, expected value of the parameter under the posterior distribution gives the optimal answer, in addition to giving a more informative framework about the behavior of the parameter.

Considering the existing approaches, Bayesian framework seems to be the most powerful and most generic approach. Because of this, we have chosen to apply Bayesian methodology to find subgroups in networks and describe their evolution through time. We will use *stochastic block models* as the underlying generative model for formation of clusters. In order to infer the time points at which changes occur in the existing clusters, we will use *hierarchical Markov models* for exact inference. Another approach for the inference of change points will be *partical Markov chain Monte Carlo* methodology. We will develop a *forward filtering backward sampling* algorithm to sample the change points. By using this algorithm, we will try to show that Monte Carlo sampling converge to posterior distribution obtained by exact inference with the advantage of less computational complexity.

1.2. LITERATURE SURVEY

Networks are used in many scientific fields. Well-known examples are social interactions [1, 14], biological applications such as protein-protein interaction networks [15–17], functional and co-expression gene similarity networks and gene regularity networks, analyzing of telecommunication networks and possible terrorist attacks, transportation networks, ecological networks such as food-webs, mobile phone networks and many others [18, 19].

Use of networks goes back to 1950's and the first mathematical model is the *Erdös-Réyni-Gilbert (ERG) random graph model* [20, 21]. According to this model, a static undirected network having N nodes and E edges is uniformly drawn from the possible $\binom{\binom{N}{2}}{E}$ graphs and it is denoted as $G(N, E)$ model. Another view point is that, every possible edge exists in graph with probability p, resulting in the expected number of edges $p\binom{N}{2}$. The possible simplest extension of the ERG model is done by customizing the edge probability between nodes according to some latent characteristics of the corresponding nodes. This is known as the *exchangeable graph model* studied by Airoldi et al. [22]. Another model used especially in the analysis of social networks is the p_1 *model* of Holland and Leinhardt [23, 24]. This model considers the dyadic pairings and describes the probability of a directed edge between two nodes in terms

of parameters such as expansiveness, popularity, mutuality in addition to a base edge propagation rate. All these parameters, except the base propagation rate, are node specific. Increasing randomness results in the p_2 *model* in which node specific parameters such as expansiveness and popularity are drawn from a underlying probability distribution [25]. As the number of the parameters increase Bayesian approach uses Monte Carlo Markov chain (MCMC) methods. A very popular graph model especially among physicists is the *exponential random graph model* (ERGM). Another interesting model is the *random graphs with fixed degree distributions*. In these models either the parameters of the degree distributions are fixed, or distributions that are conditional on some form of functions of degree distributions or degree sequences [26,27]. Diaconis et al. describes an importance sampling algorithm for generating random graphs with the given degree sequence [28].

A popular problem which has been focus of attention for at least 40 years is the *partitioning of nodes into homogenous groups*. The first example of this was encountered in social network analysis and it is known as *block-modeling* [29]. The first approaches for block-modeling were algorithmic search strategies that uses sociometrics such as cliques, status hierarchies, closures etc. [22]. Stochastic block-model approach, discover the block structures in a probabilistic way and using a statistical criterion like a likelihood function or observing a posteriori distribution. In stochastic block models, each node belongs to one or more sets and the edge probability between any two nodes depends on the blocks that the vertices belongs. One of the earliest examples of this model is proposed by Snijders and Nowicki in [30]. They work with undirected graphs and assume a node can belong to only one block. Therefore, they interpret assigning nodes to different blocks as a graph coloring problem.

Then, authors extend the given model for the directed graphs where relations can take values from a discrete finite set and number of blocks is unknown [31]. They also consider the case of missing data in the adjacency matrix. By using a prior Dirichlet density distribution for the unknown parameters, they follow a more generic Bayesian approach. They use Gibbs sampling to obtain the conditional distribution of the parameters and the block assignments.

Kemp et al. consider the same problem under the name *infinite block-model* for discovering latent classes in relational data [32]. They also consider the number of classes as unknown and allows the growth of number of classes by using Chinese restaurant process (CRP).

Hofman and Wiggins also describe the model as a mixture problem. But this time, they use a mixture of Erdös-Réyni graphs as the generating model. The percentages of group memberships come from a multinomial distribution, and they assume a Dirichlet prior for this multinomial parameters. Then, they assume in-group connections and between-group connections to be Bernoulli distributions with Beta priors. In order to infer the number of latent groups, they use the posterior distribution obtained by variational Bayes [33].

So far, relations in the networks were defined in only one type of items. If there are two type of items (dyadic data), problem is known as co-clustering or bi-clustering, meaning clustering of both items simultaneously. A good application area of this problem is the automated recommender systems, e.g., movie recommendation systems, market basket analysis, that uses collaborative filtering approaches.

Clustering of both items simultaneously is the basis of stochastic block-modeling. Hartigan et al. try to directly cluster the states that vote similarly (see Figure 1.6) and cluster the years that votes are similar, simultaneously, not independently. He defines a cluster as a submatrix of the original matrix and tries to find the submatrices whose sum of squares from the cluster averages are as small as possible. In order to achieve this, he uses a splitting scheme [34]. The found clusters are given in the Figure 1.7. This work is important since it is the first example in co-clustering.

Bayesian versions of co-clustering have been defined. A recent model defined by Shan and Banerjee, clusters two types of items simultaneously. According to their model, relations between items can take very different values other than binary values. In addition, items can belong to more than one group, meaning a mixed membership stochastic block model. This time, two Dirichlet priors are used for row and column

State	Year																	
	00	04	08	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68
Alabama (AA)	35	21	24	8	22	31	27	48	14	13	14	18	19	35	39	42	70	14
Arkansas (AS)	35	40	37	20	28	39	29	39	13	18	21	30	21	44	46	43	44	31
Delaware (DE)	54	54	52	33	50	56	58	65	51	43	45	45	50	52	55	49	39	45
Florida (FA)	19	21	22	8	18	31	28	57	25	24	26	30	34	55	57	52	48	41
Georgia (GA)	29	18	31	4	7	29	18	43	8	13	15	18	18	30	33	37	54	30
Kentucky (KY)	49	47	48	25	47	49	49	59	40	40	42	45	41	50	54	54	36	44
Louisiana (LA)	21	10	12	5	7	31	20	24	7	11	14	19	17	47	53	29	57	23
Maryland (MD)	52	49	49	24	45	55	45	57	36	37	41	48	49	55	60	46	35	42
Mississippi (MI)	10	5	7	2	5	14	8	18	4	3	4	6	3	40	24	25	87	14
Missouri (MO)	46	50	49	30	47	55	50	56	35	38	48	48	42	51	50	50	36	45
North Car. (NC)	45	40	46	12	42	43	55	29	29	27	26	33	33	46	49	48	44	40
South Car. (SC)	7	5	6	1	2	4	2	9	2	1	4	4	4	49	25	49	59	39
Tennessee (TE)	45	43	46	24	43	51	44	54	32	31	33	39	37	50	49	53	44	38
Texas (TS)	31	22	22	9	17	24	20	52	11	12	19	17	25	53	55	49	37	40
Virginia (VA)	44	37	38	17	32	38	33	54	30	29	32	37	41	56	55	52	46	43
West Virginia (WV)	54	55	53	21	49	55	49	58	44	39	43	45	42	48	47	54	32	40

Figure 1.6. Votes given by US states to Republicans in the 20th century.

State	Year																	
	12	36	32	40	44	48	16	04	68	08	24	00	20	28	56	60	52	64
SC	1	1	2	4	4	4	2	5	39	6	2	7	4	9	25	49	49	59
MI	2	3	4	4	6	3	5	5	14	7	8	10	14	18	24	25	40	87
GA	4	13	8	15	18	18	7	18	30	31	18	29	29	45	33	37	30	54
LA	5	11	7	14	19	17	7	10	23	12	20	21	31	24	53	29	47	57
AA	8	13	14	14	18	19	22	21	14	24	27	35	31	48	39	42	35	70
TS	9	12	11	19	17	25	17	22	40	22	20	31	24	52	55	49	53	37
FA	8	24	25	26	30	34	18	21	41	22	28	19	31	57	57	52	55	48
AS	20	18	13	21	30	21	28	40	31	37	29	35	35	39	46	43	44	44
VA	17	29	30	32	37	41	32	37	43	38	33	44	38	54	55	52	56	46
NC	12	27	29	26	33	33	42	40	40	46	40	45	43	55	49	48	46	44
TE	24	31	32	33	39	37	43	43	38	46	44	45	51	54	49	53	50	44
KY	25	40	40	42	45	41	47	47	44	48	49	49	49	59	54	54	50	36
MD	24	37	36	41	48	49	45	49	42	49	45	52	55	57	60	46	55	35
MO	30	38	35	48	48	42	47	50	45	49	50	46	55	56	50	50	51	36
WV	21	39	44	43	45	42	49	55	40	53	49	54	55	58	54	47	48	32
DE	33	43	51	45	45	50	50	54	45	52	58	54	56	65	55	49	52	39

Figure 1.7. Resulting clusters found by Hartigan.

items separately. In order to generate membership assignments, any discrete distribution may be used and for generation of relationship values any distribution from the exponential family can be used. As in the previous models, relationships between nodes depends on the groups of corresponding nodes [35].

Another extension to the basic stochastic block-model is considering the degree distributions of nodes. Karrer and Newman propose a degree-corrected stochastic block-model where multiple edges and self loops are allowed. According to their model, number of edges between any pair of nodes and self-loops is Poisson distributed with the parameter between class connectivity times the expected degree of the corresponding nodes [36].

In the collaborative filtering literature, matrix factorizations are another branch other than co-clustering for modeling of dyadic data. Jordan et al. integrated the latent matrix factorization and mixed membership modeling approaches and defined a Bayesian mixed membership matrix factorization model [37]. In spite of their good performance of dyadic data, due to the fact that people do not make the same preferences in different contexts, authors criticize the static nature of latent matrix factorization models. They found mixed membership models also restrictive, because of the pure reliance on the group wise interactions. According to their model, group assignments for both item types is drawn from a multinomial distribution with a Dirichlet prior. Latent factor vectors (related to the matrix factorization side) are drawn from Normal distributions with Normal-Wishart prior distributions. The relations between items are drawn from Normal distribution with a mean equal to latent factor vector productions plus the group dependent item-item biases. Authors aim at the prediction of unobserved data and for this purpose integrates out all the parameters and group assignments. Due to the intractability of the posterior distribution, they use Gibbs sampling to sample all the hyper-parameters, group assignments, item biases and item-item relations. Then, they calculate the Monte Carlo expectation of the relation predictions.

Olding et al. discuss open problems in network inference problems such as model elicitation, approximate inference and data sampling approaches [38]. One of the problems is model elicitation. Given a network, it is not clear how to select an appropriate underlying generative model. Another difficulty is the expansiveness of the exact inference computation due to the large size of the networks. As a solution, approximate inference schemes are defined. Authors underline the need for mechanisms that can evaluate the quality of the approximate solution. Most of the work in literature de-

pends on the interpretation of links between the nodes. Implications of unobserved links are not well-understood on inference methods. Unobserved links may occur due to the missing data or they may be the true result of the underlying structure.

2. BAYESIAN INFERENCE IN STATIC AND DYNAMIC NETWORKS

In this section, we will explain step by step the building blocks for Bayesian graph clustering on SBMs and detecting change points in time serial network data. We will begin with simpler ideas and show how they can be extended to deal with graphical data and time serial data. We will first study the mixture model and Bayesian inference methodology on the basic mixture model. Then we will revisit the Erdös-Réyni graph model and see how the stochastic block-model can be seen as a mixture of Erdös-Réyni graphs. We will develop Bayesian algorithms to infer the latent block structure. After that, we will study hidden Markov models (HMMs) on one dimensional sequential time series data, basically forward-backward algorithm. After that, we will develop an algorithm that combines HMMs and Monte Carlo methodology (forward filtering-backward sampling), and apply it to graphical time serial data. Although more complex variations of SBMs are given in literature, we wil stick to basic SBM where category assignments are hard and number of categories are known apriori. We do so in order to make our jump to time series analysis easier.

2.1. Graphical Models as Mixture Models

2.1.1. Bayesian Inference in General Mixture Models

Known probability distributions have some analytical properties that make our job easier. Unfortunately, in some cases it is not possible to describe a data set with a single distribution. For example, consider the Figure 2.1. Suppose, it shows the probability density function of a random variable x. Yellow points represents real probability densities. If we consider only the yellow points, we see that it is not possible to represent it with a single already known distribution. On the other hand, if we suppose three underlying Gaussian distributions whose linear combinations form the resulting distribution, we can describe the target distribution more precisely. In order to state which data point comes from which underlying distribution, we need a

latent variable. Each latent variable also comes from a probability distribution. The seen probability distribution is obtained by marginalizing the joint probability of latent variables and data points. By means of these latent variables, linear combinations of underlying distributions are obtained. As a summary, distributions formed by linear combinations of other known distributions are referred as *mixture models* and we can represent such complex distributions by mixtures of simpler distributions [13].

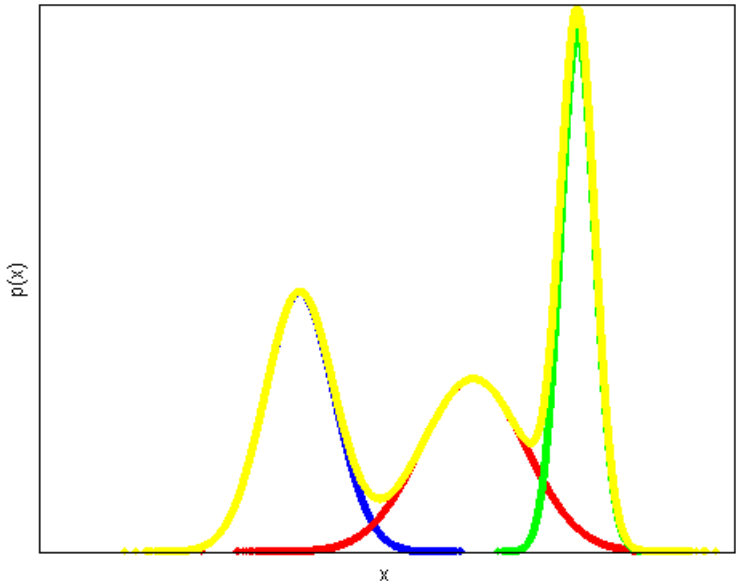


Figure 2.1. The distribution of the mixture of Gaussian variables.

In addition to forming complex probability distributions, mixture models are also used to cluster data. In a clustering problem, we have a data set $x_1, ..., x_N$ of random variable x. Our goal is to partition the data set in to K categories (or clusters). K may be given, or not. For simplicity, we will study the case in which it is given. For each variable x_n, we introduce a binary latent variable c_n of length K. $c_{nk} = 1$ if x_n belongs to category k, where $k = 1, ..., K$ and $\sum_k c_{nk} = 1$ meaning a variable belongs to one cluster only (hard clustering) . By this, we are coding the category assignment of variables by 1-of-K representation.

A general mixture model can be defined as follows:

$$
\begin{aligned}
&N : \text{Number of samples} \\
&x_{1:N} : \text{samples} \\
&K : \text{Number of categories} \\
&k \in \{1, \ldots, K\} : \text{index for categories} \\
&n \in \{1, \ldots, N\} : \text{index for samples} \\
&c_n \sim Multn(\pi) : \text{Category of sample n} \\
&\pi : \text{parameter set representing the proportion of each category} \\
&\quad \text{i.e. probability of a point belonging to each category} \\
&\beta_k : \text{parameter set for category k}
\end{aligned}
\tag{2.1}
$$

We can see the underlying generative model in the Figure 2.2.

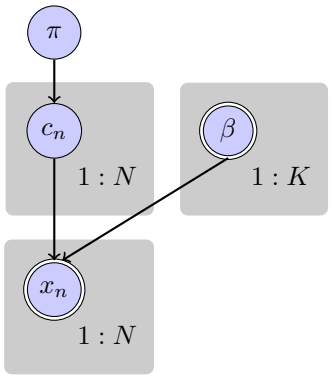


Figure 2.2. Generative model for 1D mixture model.

So probability of a sample point and a certain category assignment is

$$P(x_n, c_n) = P(c_n)P(x_n|\beta_{1:K}) \tag{2.2}$$

Thus, marginal probability of a sample point is

$$P(x_n|\beta_{1:K}) = \sum_{c_n} P(c_n)P(x_n|c_n, \beta_{1:K}) \tag{2.3}$$

Joint probability of all sample points is

$$P(x_{1:N}|\beta_{1:K}) = \prod_{n}^{N} \sum_{c_n} P(c_n)P(x_n|c_n, \beta_{1:K}) \tag{2.4}$$

Joint log probability of all sample points is

$$\begin{aligned} logP(x_{1:N}|\beta_{1:K}) &= log(\sum_{c_{1:N}} P(x_{1:N}, c_{1:N}|\beta_{1:K})) \\ logP(x_{1:N}|\beta_{1:K}) &= \sum_{n}^{N} log(\sum_{c_n} P(c_n)P(x_n|c_n, \beta_{1:K})) \end{aligned} \tag{2.5}$$

Now suppose, we do not know the parameter set —$\beta_{1:K}$, and we want to infer it. So our goal is

$$\underset{\beta_{1:K}}{\text{argmax}} \quad \mathcal{L}(x_{1:N}|\beta_{1:K}) = log(\sum_{c_{1:N}} P(x_{1:N}, c_{1:N}|\beta_{1:K})) \tag{2.6}$$

In order to evaluate above Equation 2.6 we need exponential number of calculations in terms of $c_{1:N}$. So we apply the following trick: *Divide and multiply by any distribution* $q(C), C = \{c_{1:N}\}$

$$\underset{\beta_{1:K}}{\text{argmax}} \quad \mathcal{L}(x_{1:N}|\beta_{1:K}) = log\big(\sum_{C} (P(x_{1:N}, c_{1:N}|\beta_{1:K}) \frac{q(C)}{q(C)})\big)$$

We know that *log* is an concave function and by Jensen's inequality

$$\begin{aligned} log\big(\sum_{C} (P(x_{1:N}, c_{1:N}|\beta_{1:K}) \frac{q(C)}{q(C)})\big) \geq \sum_{C} (log(P(X_{1:N}, C_{1:N}|\beta_{1:K})))q(C) - \sum_{C} log(q(C)) = \\ \langle log(P(x_{1:N}, c_{1:N}))\rangle_{q(C)} - \langle log(q(C))\rangle_{q(C)} \end{aligned} \tag{2.7}$$

Lets call

$$\mathcal{Q} = \langle log(P(x_{1:N}, c_{1:N}))\rangle_{q(C)} - \langle log(q(C))\rangle_{q(C)}$$

So if we maximize $\mathcal{Q}$ we can find a lower bound for $\mathcal{L}(x_{1:N}|\beta_{1:K})$ Therefore, we can define any distribution q(C) and take expectation.

In summary, theoretical EM algorithm can be formulated as follows:
E Step:

$$q(C) = P(C|X, \beta^{old}) \sim P(C, X, \beta^{old}) \frac{1}{P(X, \beta^{old})} \tag{2.8}$$

M Step:

$$\underset{\beta_{1:K}}{\text{argmax}} \quad \langle log(P(C, X, \beta^{old})) \rangle_{q(C)}$$

2.1.2. Stochastic Block Model as a Mixture Model

Before we dive into SBMs, we will study Erdös-Réyni-Gilbert graphs and Bayesian inference on them. By this, we will prepare a basis for Bayesian inference on SBMs.

2.1.2.1. Bayesian Inference On Erdös-Réyni-Gilbert Graphs. As stated in literature survey section, ERG graphs are first examples of random graph models. Basic probabilistic graphical model for an undirected ERG graph can be represented as in the below Figure 2.3

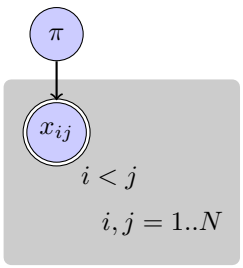


Figure 2.3. Graphical model for Erdös-Réyni-Gilbert graph.

According to this model, we can see that there are N nodes in the graph and the probability of an edge between nodes i and j, $i < j$ is Bernoulli distributed with the unknown parameter π.

Given the parameter π probability of an edge between any two nodes i and j, x_{ij} is

$$p(x_{ij}|\pi) = \pi^{x_{ij}} * (1-\pi)^{(1-x_{ij})}$$

Again given the parameter π, probabilities of edges are conditionally independent and probability of the whole undirected network X (no self-loops, no double links, no circles etc.) can be written as

$$p(X|\pi) = \prod_{j=1}^{N} \prod_{i}^{j} \pi^{x_{ij}} * (1-\pi)^{(1-x_{ij})}$$

If we take the logarithm of the above equation, we obtain

$$\log p(X|\pi) = \big(\sum_{j=1}^{N} \sum_{i}^{j} x_{ij}\big) * \log(\pi) + \big(\sum_{j=1}^{N} \sum_{i}^{j} (1-x_{ij})\big) * \log(1-\pi)$$

By the classical maximum likelihood approach, we can find the π value that maximizes the log-likelihood by taking the derivative and equating it to 0. Thus we obtain a point estimate of the parameter π,

$$\frac{\partial \log p(X|\pi)}{\partial \pi} = \frac{\sum_{j=1}^{N} \sum_{i}^{j} x_{ij}}{\pi} - \frac{\sum_{j=1}^{N} \sum_{i}^{j} (1-x_{ij})}{\log(1-\pi)} = 0$$

$$\pi = \frac{\sum_{j=1}^{N} \sum_{i}^{j} x_{ij}}{\frac{N(N-1)}{2}}$$

Let us now consider parameter estimation in Bayesian framework. In Bayesian view, *probability* is interpreted as *degree of belief* in a position. So probability distribution associates every possible state with a degree of belief. Belief about an unknown variable (in our Erdös-Réyni-Gilbert case, π) without any observed data (evidence) is represented by *prior* distribution. Every observed data updates the prior belief about the unknown variables resulting in *posterior* distribution. While using well-known distributions such as Gaussian, Bernoulli, gamma distributions for the evidence distributions, conjugate priors are used for the unknown parameter distributions in order to obtain well-known distributions as the posteriors. Posterior distributions represent the believes for each possible value of the unknown parameters after observing the data. High belief values mean high probability values. So higher the belief in the posterior value, higher the possibility of the corresponding parameter value.

Conjugate prior of the Bernoulli distribution is the Beta distribution. So let us assume prior distribution of the unknown parameter π is $Beta(\pi;\alpha,\beta)$. The posterior distribution of unknown π is obtained by

$$\begin{aligned}
p(\pi|X) =& \frac{p(X|\pi)p(\pi)}{p(X)} \\
=& \frac{p(X|\pi)p(\pi)}{\int p(X|\pi)p(\pi)\,d\pi} \\
\propto& p(X|\pi)p(\pi) \\
=& \Big(\prod_{j=1}^{N}\prod_{i}^{j}\pi^{x_{ij}}(1-\pi)^{(1-x_{ij})}\Big)\frac{\Gamma(\alpha+\Gamma)}{\Gamma(\alpha)\Gamma(\beta)}\pi^{\alpha-1}(1-\pi)^{\beta-1} \\
=& \frac{\Gamma(\alpha+\Gamma)}{\Gamma(\alpha)\Gamma(\beta)}\ \pi^{(\sum_{j=1}^{N}\sum_{i}^{j}x_{ij})+\alpha-1}(1-\pi)^{(\sum_{j=1}^{N}\sum_{i}^{j}(1-x_{ij}))+\beta-1}
\end{aligned}$$

If we multiply the latest equation with

$$\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}\frac{\Gamma\big((\sum_{j=1}^{N}\sum_{i}^{j}x_{ij})+\alpha-1\big)\Gamma\big((\sum_{j=1}^{N}\sum_{i}^{j}(1-x_{ij}))+\beta-1\big)}{\Gamma\big(\frac{N(N-1)}{2}+\alpha+\beta\big)}$$

we obtain the distribution $Beta(\pi; \alpha^{new}, \beta^{new})$ where $\alpha^{new} = (\sum_{j=1}^{N} \sum_{i}^{j} x_{ij}) + \alpha$ and $\beta^{new} = (\sum_{j=1}^{N} \sum_{i}^{j} (1 - x_{ij})) + \beta$.

Thus,

$$p(\pi|X) = Beta(\pi; \alpha^{new}, \beta^{new}) \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{\Gamma(\alpha^{new})\Gamma(\beta^{new})}{\Gamma(\alpha^{new} + \beta^{new})}$$

Since the prior parameters $\alpha, \beta, \alpha^{new}, \beta^{new}$ can be taken to be fixed, therefore

$$p(\pi|X) \propto Beta(\pi; \alpha^{new}, \beta^{new})$$

As a result, we obtain a Beta distribution scaled by some constants as the posterior distribution. This posterior gives believes for each possible value of the unknown parameter π.

2.1.2.2. Stochastic Block Model as a Mixture of Erdös-Réyni-Gilbert Graphs. The main aim behind the block models is finding clusters of vertices, also called communities or modules, in high dimensional data such as a network. In a network, nodes may belong to one or more different types. Consider for example a network that represents usage behavior of users for some services. Naturally, service categories and user categories differ from each other. And a service or a user belongs to a category. According to stochastic block models, interactions between a user and a service are determined by the belonged categories of the service and the user. And this category assignment is usually unknown, referred to as *latent structure*. The generative model for a stochastic block model can be better understood from the below probabillistic graphical model in Figure 2.4. First we will assume there is only one node type in the network. Each node belongs to an unknown category and connections between nodes are determined by this hidden categories. The underlying generative model can be written as

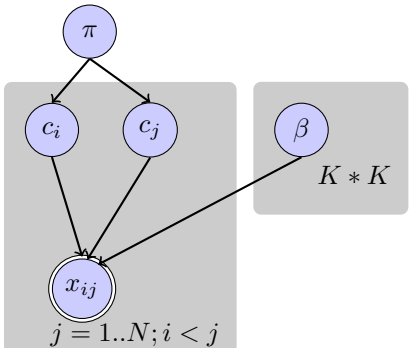


Figure 2.4. SBM for single item type network.

N : Number of nodes

K : Number of categories

$k \in \{1, \ldots, K\}$: index for categories

$p \in \{1, \ldots, N\}$: index for nodes

C_p : Category of node p

B_{kl} : Connection probability between a node from group k and

a node from a group l

$$Y_{pq} = \begin{cases} 1 & \text{if node } p \text{ is connected to node } q \\ 0 & \text{otherwise} \end{cases} \sim Bern(C_p B C_q^T) : \text{Adjacency matrix}$$

for connections between nodes

The joint distribution of adjacency matrix, category assignments and the parameter set can be obtained by

$$\begin{aligned} P(Y, \beta, C_{1:N}) &= p(C_{1:N})p(\beta)p(Y|\beta, C_{1:N}) \\ &= \prod_p^N \prod_k^K (\pi_k)^{C_{pk}} \prod_k^K \prod_l^K \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \beta_{kl}^{(\alpha-1)} (1-\beta_{kl})^{(\beta-1)} \\ &\quad \prod_p^N \prod_k^K \prod_q^N \prod_l^K (\beta_{kl}^{Y_{pq}} (1-\beta_{kl})^{(1-Y_{pq})})^{C_{pk}C_{ql}} \end{aligned}$$

$$\text{where } C_{pk} = \begin{cases} 1 \text{ if node } p \text{ belongs to category } k \\ 0 \text{ otherwise} \end{cases}$$

If latent category assignment ($C_{1:N}$) and interaction behavior between categories (β_{K*K}) are known, interaction between any two nodes is conditionally independent.

Therefore, nodes belonging to a certain two categories will form a homogenous subgraph whose edge connections come from a certain distribution. Thus, whole network will be mixture of little homogenous subgraphs.

If we assume category assignments are multinomial distribution with the parameter set π, β_{K*L} parameter set indicates the Bernoulli parameters that determine the connection (edge) probabilities between nodes from different categories.

$\prod_p^N \prod_k^K \prod_q^N \prod_l^K (\beta_{kl}^{Y_{pq}}(1-\beta_{kl})^{(1-Y_{pq})})^{C_{pk}C_{ql}}$ is an ERG graph for any fixed two categories (k and l,either same or not) and nodes belonging to them.
$\prod_p^N \prod_k^K (\pi_k)^{C_{pk}} \prod_k^K \prod_l^K \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \beta_{kl}^{(\alpha-1)}(1-\beta_{kl})^{(\beta-1)}$ behaves as mixing coefficient of the corresponding ERG subgraph. As complex distributions can be defined by mixtures of simpler distributions, stochastic block models can also be defined by mixtures of simpler graphs. We will study Bayesian parameter estimation on a SBM where there are two different node types and there is no connection between nodes of same types. In other words, we have bi-partite graphs. Each node type has its own categories. We will use Bayesian inference to infer both the unknown category assignments for each node, as well as the unknown connection parameters that determines connectivity between different node types according to their categories.

First of all let us look at the generative graph model in Figure 2.5. Assume that there are users and services. Each user belongs to a category and each service belongs to a service category. Service usage of users is determined by the category of the user and the corresponding service.

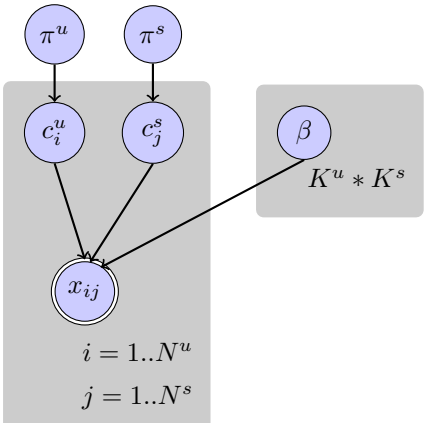


Figure 2.5. SBM for bi-partite network.

The generative model canbe written as follows:

N^u : Number of users

K^u : Number of user categories

N^s : Number of services

K^s : Number of service categories

$k \in \{1, \ldots, K^u\}$: index for user categories

$l \in \{1, \ldots, K^s\}$: index for service categories

$p \in \{1, \ldots, N^u\}$: index for users

$q \in \{1, \ldots, N^s\}$: index for services

$C_p^u \sim Multn(\pi^u)$: Category of user p

$C_q^s \sim Multn(\pi^s)$: Category of service q

$B_{kl} \sim Beta(\alpha, \beta)$: Connection probability between a user from group k and a service from a group l

$$Y_{pq} = \begin{cases} 1 & \text{if user } p \text{ uses service } q \\ 0 & \text{otherwise} \end{cases} \sim Bern(C_p^u B (C_q^s)^T)$$: Adjacency matrix for connections between users and services

(2.9)

Again we can write the full joint distribution of the observed graphs and unknown

parameters as follows:

$$
\begin{aligned}
p(Y, B, C^u_{1:N^u}, C^s_{1:N^s}) &= p(C^u_{1:N^u})p(C^s_{1:N^s})p(B)p(Y|B, C^u_{1:N^u}, C^s_{1:N^s}) \\
&= \prod_p^{N^u}\prod_k^{K^u}(\pi^u_k)^{C^u_{pk}}\prod_q^{N^s}\prod_l^{K^s}(\pi^s_l)^{C^s_{ql}}\prod_k^{K^u}\prod_l^{K^s}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}B_{kl}^{(\alpha-1)}(1-B_{kl})^{(\beta-1)} \\
&\quad \prod_p^{N^u}\prod_q^{N^s}\prod_k^{K^u}\prod_l^{K^s}(B_{kl}^{Y_{pq}}(1-B_{kl})^{(1-Y_{pq})})^{C^u_{pk}C^s_{ql}}
\end{aligned}
$$

$$
\begin{aligned}
\log p(Y, B, C^u_{1:N^u}, C^s_{1:N^s}) &= \log p(C^u_{1:N^u}) + \log p(C^s_{1:N^s}) + \log p(B) + \log p(Y|B, C^u_{1:N^u}, C^s_{1:N^s}) \\
&= \sum_p^{N^u}\sum_k^{K^u} C^u_{pk}\log(\pi^u_k) + \sum_q^{N^s}\sum_l^{K^s} C^s_{ql}\log(\pi^s_l) \\
&\quad + \sum_k^{K^u}\sum_l^{K^s}(\alpha-1)\log(B_{kl}) + (\beta-1)\log(1-B_{kl}) \\
&\quad + \log\Gamma(\alpha+\beta) - \log\Gamma(\alpha) - \log\Gamma(\beta) \\
&\quad + \sum_p^{N^u}\sum_q^{N^s}\sum_k^{K^u}\sum_l^{K^s} C^u_{pk}C^s_{ql}(Y_{pq}\log(B_{kl}) + (1-Y_{pq})\log(1-B_{kl}))
\end{aligned}
$$

We can write the log likelihood probability and lower bound of it as follows:

$$
\underset{B}{\text{argmax}} \quad \mathcal{L}(Y|B) = \log\big(\sum_{C^u}\sum_{C^s} P(Y, C^u, C^s|B)\big)
$$

By Jensen's inequality we know that

$$
\log\big(\sum_{C^u}\sum_{C^s}(P(Y, C^u, C^s|B)\frac{q(C^u, C^s)}{q(C^u, C^s)})\big) \geq \sum_{C^u}\sum_{C^s}\log(P(Y, C^u, C^s|B))\frac{q(C^u, C^s)}{q(C^u, C^s)}
$$

Lets call

$$
\mathcal{Q} = \langle\log(P(Y, B, C^u_{1:N^u}, C^s_{1:N^s}))\rangle_{q(C^u, C^s)} - \langle\log(q(C^u, C^s))\rangle_{q(C^u, C^s)}
$$

So we can find B by

$$\underset{B}{\operatorname{argmax}}\,\mathcal{Q} = \underset{B}{\operatorname{argmax}} \left(\begin{array}{l} \sum_{k}^{K^u}\sum_{l}^{K^s}(\alpha-1)\log(B_{kl}) + (\beta-1)\log(1-B_{kl}) \\ + \sum_{p}^{N^u}\sum_{q}^{N^s}\sum_{k}^{K^u}\sum_{l}^{K^s} C^u_{pk}C^s_{ql}(Y_{pq}\log(B_{kl}) + (1-Y_{pq})\log(1-B_{kl})) \end{array} \right)$$

$$\begin{aligned} \frac{\partial \mathcal{Q}}{\partial B_{kl}} &= \sum_{p}^{N^u}\sum_{q}^{N^s}\langle C^u_{pk}C^s_{ql}\rangle\left(\frac{Y_{pq}}{B_{kl}} - \frac{1-Y_{pq}}{1-B_{kl}}\right) + \frac{\alpha-1}{B_{kl}} - \frac{\beta-1}{1-B_{kl}} = 0 \\ &\Rightarrow B_{kl} = \frac{\sum_p^{N^u}\sum_q^{N^s}\langle C^u_{pk}C^s_{ql}\rangle Y_{pq} + \alpha - 1}{\sum_p^{N^u}\sum_q^{N^s}\langle C^u_{pk}C^s_{ql}\rangle + \alpha + \beta - 2} \end{aligned} \tag{2.10}$$

In EM we take $q(C^u_{1:N^u}, C^s_{1:N^s}) = P(C^u_{1:N^u}, C^s_{1:N^s}|Y, B)$

$$\begin{aligned} \log p(C^u_{1:N^u}, C^s_{1:N^s}|Y,B) =^+ &\sum_{p}^{N^u}\sum_{k}^{K^u} C^u_{pk}\log(\pi^u_k) + \sum_{q}^{N^s}\sum_{l}^{K^s} C^s_{ql}\log(\pi^s_l) \\ &+ \sum_{p}^{N^u}\sum_{q}^{N^s}\sum_{k}^{K^u}\sum_{l}^{K^s} C^u_{pk}C^s_{ql}(Y_{pq}\log(B_{kl}) + (1-Y_{pq})\log(1-B_{kl})) \end{aligned} \tag{2.11}$$

Since C^u_{pk} and C^s_{ql} are coupled we can assume $q(C^u_{1:N^u}, C^s_{1:N^s})$ to be factorized as $q(C^u_{1:N^u})q(C^s_{1:N^s})$. Because of this factorization, this is variational expectation-maximization algorithm.

$$logq(C^u_{pk}) =^+ C^u_{pk}log(\pi^u_k) + \sum_{q}^{N^s}\sum_{l}^{K^s}(C^u_{pk}C^s_{ql})[Y_{pq}log(B_{kl}) + (1-Y_{pq})log(1-B_{kl})] \tag{2.12}$$

And similarly,

$$\log q(C^s_{ql}) =^+ C^s_{ql}\log(\pi^s_l) + \sum_{p}^{N^u}\sum_{k}^{K^u}(C^u_{pk}C^s_{ql})[Y_{pq}\log(B_{kl}) + (1-Y_{pq})\log(1-B_{kl})] \tag{2.13}$$

2.2. Time Series Modeling with Hidden Markov Models

2.2.1. Sequential Data

Before dealing with multiple change points in sequential data, let us remember ordinary sequential data models. Sequential data is generated from a stationary sequential distribution. In other words, data evolves in time but the distribution from which it is generated remains the same.

In general Markov models, each observation depends on the observations before itself. Thus joint probability distribution of an observation sequence can be written as

$$p(x_1, ..., x_N) = \prod_{n=1}^{N} p(x_n|x_1, ..., x_{n-1}) \tag{2.14}$$

If it is assumed that an observation depends only on the most recent observation before itself and independent of the other previous ones, the model is a *first-order Markov chain*. The joint probability can be written as

$$p(x_1, ..., x_N) = p(x_1) \prod_{n=2}^{N} p(x_n|x_{n-1}) \tag{2.15}$$

Generalizing this, in n^{th}*-order Markov chain* an observation depends on the most recent n observations before itself. If we think of hierarchical models, dependence between observations is not enough. We should somehow include the latent variables in the models. In *hidden Markov models*, each observation in a time slice depends on the latent variable and dependence of latent variables of a time slice on previous time slices may be in various relations. The most basic relation is the first order Markov dependence between the latent variables. You can see the relation in Figure 2.6.

At this point, it is good to see the relation of SBMs observed in subsequent time

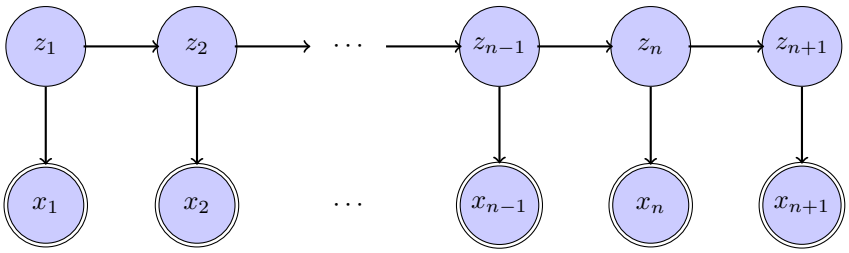


Figure 2.6. Hidden Markov Model: Double-bordered nodes represent the observations and single-bordered nodes represents the latent variables.

steps. In Figure 2.7, we can see the hierarchical Markov Model for SBM time series.

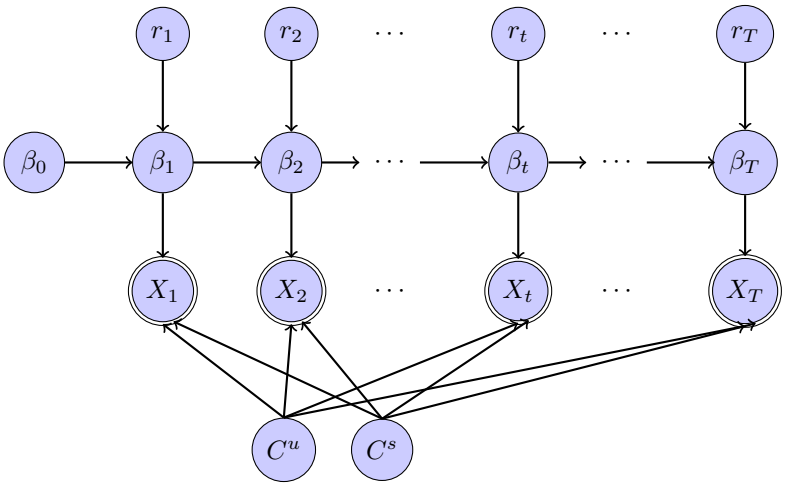


Figure 2.7. SBM time series: At each time step, category assignments are taken to be fixed but unknown. At each time step, there can be a change, and this results in the change of the connectivity matrix. If there is no change, connectivity matrix of the previous step is used. Dependence between consequtive β_t forms a first order markov model.

If we observe the relation between consecutive β matrices in Figure 2.7, we see the hidden Markov model here. In the following sections, we will first explain the widely used forward-backward algorithm, then give the mathematical model generating the SBM time series. After that, we will derive the forward-backward algorithm for this model.

2.2.2. Forward-Backward Algorithm

Forward-Backward algorithm is used to make inference about the latent variables in a hidden Markov model(HMM). Given the graphical model, in order to infer about the hidden variables $z_{1:N}$, we need to calculate $p(z_n|x_1, ..., x_N)$. Making use of conditional independence properties obtained by d-separation rules, we can easily write the following

$$\begin{aligned} p(z_n|x_{1:N}) =& \frac{p(x_{1:N}|z_n)p(z_n)}{p(x_{1:N})} \\ =& \frac{p(x_{1:n}|z_n)p(x_{n+1:N}|z_n)p(z_n)}{p(x_{1:N})} \\ =& \frac{p(x_{1:n}, z_n)p(x_{n+1:N}|z_n)}{p(x_{1:N})} \end{aligned} \tag{2.16}$$

If we define

$$\begin{aligned} \alpha(z_n) &= p(x_{1:n}, z_n) \\ \beta(z_n) &= p(x_{n+1:N}|z_n) \end{aligned} \tag{2.17}$$

we obtain

$$p(z_n|x_{1:N}) = \frac{\alpha(z_n)\beta(z_n)}{p(x_{1:N})} \tag{2.18}$$

Now we will show the recursion relations to calculate $\alpha(z_n)$ and $\beta(z_n)$ easily.

$$\alpha(z_n) = p(x_{1:n}, z_n) \tag{2.19}$$

$$= p(x_{1:n}|z_n)p(z_n) \tag{2.20}$$

$$= p(x_n|z_n)p(x_{1:n-1}|z_n)p(z_n) \tag{2.21}$$

$$= p(x_n|z_n)p(x_{1:n-1}, z_n) \tag{2.22}$$

$$= p(x_n|z_n) \sum_{z_{n-1}} p(x_{1:n-1}, z_{n-1}, z_n) \tag{2.23}$$

$$= p(x_n|z_n) \sum_{z_{n-1}} p(x_{1:n-1}, z_n|z_{n-1}) p(z_{n-1}) \tag{2.24}$$

$$= p(x_n|z_n) \sum_{z_{n-1}} p(x_{1:n-1}|z_{n-1}) p(z_n|z_{n-1}) p(z_{n-1}) \tag{2.25}$$

$$= p(x_n|z_n) \sum_{z_{n-1}} p(x_{1:n-1}, z_{n-1}) p(z_n|z_{n-1}) \tag{2.26}$$

$$= p(x_n|z_n) \sum_{z_{n-1}} \alpha(z_{n-1}) p(z_n|z_{n-1}) \tag{2.27}$$

If we define $\alpha_{n|n-1}(z_n) = p(z_n, x_{1:n-1})$ and redefine $\alpha_n(z_n) = \alpha_{n|n}(z_n)$ we obtain the relation:

$$\alpha_{n|n}(z_n) = p(x_n|z_n) \alpha_{n|n-1}(z_n) \tag{2.28}$$

And similarly, we obtain the following recursion equation for $\alpha_{n|n-1}$:

$$\begin{aligned}
\alpha_{n|n-1}(z_n) &= p(z_n, x_{1:n-1}) \\
&= \sum_{z_{n-1}} p(z_n, z_{n-1}, x_{1:n-1}) \\
&= \sum_{z_{n-1}} p(x_{1:n-1}, z_n|z_{n-1}) p(z_{n-1} \\
&\text{by d-separation condition} \\
&= \sum_{z_{n-1}} p(x_{1:n-1}|z_{n-1}) p(z_n|z_{n-1}) p(z_{n-1} \\
&= \sum_{z_{n-1}} p(x_{1:n-1}, z_{n-1}) p(z_n|z_{n-1}) \\
&= \sum_{z_{n-1}} \alpha_{n-1|n-1}(z_{n-1}) p(z_n|z_{n-1})
\end{aligned} \tag{2.29}$$

$$\begin{aligned}
\beta(z_n) &= p(x_{n+1:N}|z_n) \\
&= \sum_{z_{n+1}} p(x_{n+1:N}, z_{n+1}|z_n) \\
&= \sum_{z_{n+1}} p(x_{n+1:N}|z_{n+1}, z_n)p(z_{n+1}|z_n) \\
&= \sum_{z_{n+1}} p(x_{n+1:N}|z_{n+1})p(z_{n+1}|z_n) \\
&= \sum_{z_{n+1}} \beta(z_{n+1})p(x_{n+1}|z_{n+1})p(z_{n+1}|z_n)
\end{aligned} \tag{2.30}$$

We can also divide the β_n messages in to submessages in the following way: Let us define $\beta_{n|n+1}(z_n) = p(x_{n+1:N}|z_n)$ and redefine $\beta(z_n) = \beta_{n|n}(z_n)$. Following this, we obtain the relation:

$$\begin{aligned}
\beta_{n|n+1}(z_n) &= p(x_{n+1:N}|z_n) \\
&= \sum_{z_{n+1}} p(x_{n+1:N}, z_{N+1}|z_n) \\
&= \sum_{z_{n+1}} p(x_{n+1:N}|z_{N+1}, z_n)p(z_{n+1}|z_n) \\
&= \sum_{z_{n+1}} p(x_{n+1:N}|z_{N+1})p(z_{n+1}|z_n) \\
&= \sum_{z_{n+1}} \beta_{n+1|n+1}(z_{n+1})p(z_{n+1}|z_n)
\end{aligned} \tag{2.31}$$

Similarly, we find the recursion equation for $\beta_{n|n}$:

$$\begin{aligned}
\beta_{n|n}(z_n) &= p(x_{n:N}|z_n) \\
&= \sum_{z_{n+1}} p(x_n, x_{n+1:N}, z_{N+1}|z_n) \\
&\quad \text{by d-separation condition} \\
&= \sum_{z_{n+1}} p(x_n|z_n)p(x_{n+1:N}, z_{N+1}|z_n) \\
&= p(x_n|z_n) \sum_{z_{n+1}} p(x_{n+1:N}, z_{N+1}|z_n) \\
&= p(x_n|z_n)\beta_{n|n+1}(z_n)
\end{aligned} \tag{2.32}$$

By forward-backward algorithm, we can obtain the posterior distribution $p(z_n|x_{1:N})$. We have done the derivations for discrete hidden state variables. If hidden states are also continues, we replace summations with integrations.

2.3. Multiple Change Point Models

Changepoints are abrupt variations in the generative parameters of a data sequence [39]. Data can be one dimensional or multi-dimensional. Usually number and place of changepoint is not known, and main aim is to detect changepoints. The generative mathematical model behind our SBM time series data (seen in Figure 2.7) can be summarized as below:

N^u : Number of users

K^u : Number of user categories

N^s : Number of services

K^s : Number of service categories

$k \in \{1, \ldots, K^u\}$: index for user categories

$l \in \{1, \ldots, K^s\}$: index for service categories

$p \in \{1, \ldots, N^u\}$: index for users

$q \in \{1, \ldots, N^s\}$: index for services

$C_p^u \sim Multn(\pi^u)$: Category of user p

$C_q^s \sim Multn(\pi^s)$: Category of service q

$r_t \sim Be(p)$: Indicates a change point at time t

$B_t \sim [r_t = 0]\delta(B_t = B_{t-1}) + [r_t = 1] \prod_k^{K^u} \prod_l^{K^s} Beta(\alpha, b)$: If there is a change point at time t, communication matrix is drawn randomly from the prior distribution.

$C_p^u \sim Multn(\pi^u)$: Category of a node from the first item type

$C_q^s \sim Multn(\pi^s)$: Categories of a node from the second item type

(2.33)

Conditional probability of observed adjacency matrix at a time is calculated as in the previous section:

$$Y_t|B_t, C^u, C^s = \prod_p^{N^u} \prod_q^{N^s} \prod_k^{K^u} \prod_l^{K^s} (B_{t,kl}^{Y_{t,pq}} (1 - B_{t,kl})^{(1-Y_{t,pq})})^{C^u_{pk} C^s_{ql}} \tag{2.34}$$

Clearly, our model is an example of a hierarchical change point model. Main aim is to make prediction about the change points $r_{1:T}$. Thus, we should be able to find the conditional probability of change points at each time point.

$$p(r_t|Y_{1:T}) = \sum_p^{N^u} \sum_q^{N^s} \sum_k^{K^u} \sum_l^{K^s} p(r_t|Y_{1:T}, C^u_{p=1:N^u, k=1:K^u}, C^s_{q=1:N^s, l=1:K^s}) \tag{2.35}$$

Considering the dependence among the B matrices of consequent time steps, there is also a hidden Markov Model in our change point model for SBMs. Thus, we should somehow derive the forward-backward algorithm for our SBM time series model and use that to infer change points.

By use of conditional independence and d-separation rules, reobserving the Figure 2.7, at each time step, observed SBM consists of conditionally independent Erdös-Réyni graphs given the change point and category assignments of nodes.

$p(Y_t|B_t, C^u, C^s)$ can be also rewritten as:

$$p(Y_t|B_t, C^u, C^s) = \prod_p^{N^u} \prod_q^{N^s} (\sum_k^{K^u} \sum_l^{K^s} C^u_{pk} B_{t,kl} C^s_{ql})^{Y_{t,pq}} (1 - \sum_k^{K^u} \sum_l^{K^s} C^u_{pk} B_{t,kl} C^s_{ql})^{(1-Y_{t,pq})} \tag{2.36}$$

Say number of edges between nodes of user category k and service category l is $n_{k,l}$ and maximum possible edges between categories k and l is $n^{max}_{k,l}$ (in fact equal to the product of nodes in category k and number of nodes in category l).

Thus, we obtain :

$$p(Y_t|B_t, C^u, C^s) = \prod_k^{K^u} \prod_l^{K^s} B_{t,kl}^{n_{k,l}} (1 - B_{kl})^{n_{k,l}^{max} - n_{k,l}} \tag{2.37}$$

which is equal to product of Erdös-Réyni graphs.

Since category assignments are assumed not to change and every change point occurrence results in the regeneration of the connectivity matrix B_t from the prior distribution, it reduces to $K^u \times K^s$ Erdös-Réyni change point models, i.e. Erdös-Réyni time series. Thus, we will derive the forward-backward algorithm for Erdös-Réyni graph time series.

2.3.1. Forward-Backward Algorithm for Erdös-Réyni Graph Change Point Model

In this section, we will first derive the forward-backward algorithm for a single Erdös-Réyni change point model. Then in the next section, we will combine Gibbs sampling and forward-backward algorithm in the EM algorithm to integrate out the unknown category assignments.

Multiple change point model for Erdös-Réyni graph is as follows:

N : Number of nodes

$n \in \{1, \ldots, N\}$: index for nodes

$r_t \sim Be(p)$: Indicates a change point at time t

$p_t \sim [r_t = 0]\delta(B_t = p_{t-1}) + [r_t = 1]Beta(\alpha, b)$

:If there is a change point at time t, edge probability is drawn randomly

from the prior Beta distribution.

$Y_t|p_t = \prod_{n=1}^{N} \prod_{q=1}^{n} p_t^{Y_{t,nq}} (1 - p_t)^{(1-Y_{t,nq})}$: conditional probability of an observed matrix at time t

(2.38)

We will define the forward and backward messages for computing the marginals of form $p(r_t, p_t, Y_{1:T})$. Let us define α and β messages as in the previous section. But this time, in addition to discrete hidden variables r_t we have also contionous hidden variables p_t:

$$\begin{aligned}
\alpha_{0|0}(r_0, p_0) &= p(p_0) \\
t &= 1..T \\
\alpha_{t|t-1}(r_t, p_t) &= p(p_t, r_t, Y_{1:t-1}) \\
\alpha_{t|t}(r_t, p_t) &= p(p_t, r_t, Y_{1:t}) \\
\\
\beta_{T|T}(r_T, p_T) &= p(Y_T | r_T, p_T) \\
t &= T-1, .., 1 \\
\beta_{t|t+1}(r_t, p_t) &= p(Y_t + 1 : T | r_t, p_t) \\
\beta_{t|t}(r_t, p_t) &= p(Y_t : T | r_t, p_t)
\end{aligned} \tag{2.39}$$

Similar to Equations 2.28, 2.29, 2.31 and 2.32 we obtain the recursion equations for α and β messages as follows:

$$\alpha_{t|t}(r_t, p_t) = p(Y_t | r_t, p_t) \alpha_{t|t-1}(r_t, p_t) \tag{2.40}$$

And similarly, we obtain the following recursion equation for $\alpha_{t|t-1}$:

$$\begin{aligned}
\alpha_{t|t-1}(r_t, p_t) &= p(r_t, p_t, Y_{1:t-1}) \\
&= \sum_{r_{t-1}, p_{t-1}} \alpha_{t-1|t-1}(r_{t-1}, p_{t-1}) p(r_t, p_t | r_{t-1}, p_{t-1})
\end{aligned} \tag{2.41}$$

$$\begin{aligned}
\beta_{t|t+1}(r_t, p_t) &= p(Y_{t+1:T} | r_t, p_t) \\
&= \sum_{r_{t+1}, p_{t+1}} \beta_{t+1|t+1}(r_{t+1}, p_{t+1}) p(r_{t+1}, p_{t+1} | r_t, p_t)
\end{aligned} \tag{2.42}$$

Similarly, we find the recursion equation for $\beta_{t|t}$:

$$\begin{aligned}\beta_{t|t}(r_t, p_t) &= p(Y_{t:Y}|r_t, p_t) \\ &= p(Y_t|r_t, p_t)\beta_{t|t+1}(r_t, p_t)\end{aligned} \tag{2.43}$$

In order to obtain the posterior marginals of hidden variables (r_t, p_t), let us use the α and β messages:

$$\begin{aligned}p(r_t, p_t|Y_{1:T}) &\sim p(r_t, p_t, Y_{1:T}) \\ &\text{by conditional dependence} \\ &= p(r_t, p_t, Y_{1:t-1})p(Y_{t:T}|r_t, p_t, Y_{1:t-1}) \\ &\text{by d-separation} \\ &= p(r_t, p_t, Y_{1:t-1})p(Y_{t:T}|r_t, p_t) \\ &\text{by using definitions of } \alpha \text{ and } \beta \text{ messages} \\ &= \alpha_{t|t-1}(r_t, p_t)\beta_{t|t}(r_t, p_t) \\ &= \alpha_{t|t}(r_t, p_t)\beta_{t|t+1}(r_t, p_t)\end{aligned} \tag{2.44}$$

Since we are interested in inferring about the change points at each time step (r_t), we integrate out the graph parameters (p_t).

$$p(r_t|Y_{1:T}) = \int_{p_t} \alpha_{t|t-1}(r_t, p_t)\beta_{t|t}(r_t, p_t) \tag{2.45}$$

Now we should find the closed form distributions for $p(r_t|Y_{1:T})$.

2.3.1.1. Derivation of Forward Messages. Let us start with $\alpha_{0|0}(p_0) = p(Y_0, p_0)$. In the first step r_t is not taken into account, because if it is a change point or not, it is

the beginning condition for us.

$$\begin{aligned}\alpha_{0|0}(p_0) &= p(p_0, Y_0) \\ &= p(Y_0|p_0)p(p_0) \\ &= \prod_{p=1}^{N}\prod_{q=1}^{p}(p_0^{Y_{0,pq}}(1-p_0)^{1-Y_{0,pq}})\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}p_0^{\alpha-1}(1-p_0)^{\beta-1}\end{aligned}$$

We take the logarithm

$$\begin{aligned}&= ((\sum_{p=1}^{N}\sum_{q<p}Y_{0,pq}) + \alpha - 1)log(p_0) + ((\sum_{p=1}^{N}\sum_{q<p}1 - Y_{0,pq}) + \beta - 1)log(1-p_0) \\ &+ log(\Gamma(\alpha+\beta)) - log(\Gamma(\alpha)) - log(\Gamma(\beta))\end{aligned} \tag{2.46}$$

If we multiply both sides with

$$\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}\frac{\Gamma(\alpha+\beta+\frac{N(N+1)}{2})}{\Gamma(\sum_{p=1}^{N}\sum_{q<p}Y_{0,pq})+\alpha)\Gamma(\sum_{p=1}^{N}\sum_{q<p}1-Y_{0,pq})+\beta)} \tag{2.47}$$

Let us define

$$\begin{aligned}n_0 &= \sum_{p=1}^{N}\sum_{q<p}Y_{0,pq} \\ n_0^{'} &= \sum_{p=1}^{N}\sum_{q<p}(1-Y_{0,pq}) \\ \alpha_0^{new} &= n_0 + \alpha \\ \beta_0^{new} &= n_0^{'} + \beta \\ c_0 &= \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}\frac{\alpha_0^{new}+\beta_0^{new}}{\Gamma(\alpha_0^{new})\Gamma(\beta_0^{new})}\end{aligned}$$

Then we obtain the following equation

$$\alpha_{0|0}(p_0) = \frac{1}{c_0} * Beta(p_0; \alpha_0^{new}, \beta_0^{new}) \tag{2.48}$$

This is a Beta potential, which can be represented by $(\alpha_0^{new}, \beta_0^{new}, c_0)$ where c_0 is the normalization constant. Let us now derive the equations for $\alpha_{1|0}(p_1, r_1)$. For $r_1 = 1$, by the definition of our change point model, time step t_1 becomes independent from time step t_0.

$$\begin{aligned}
\alpha_{1|0}(p_1, r_1 = 1) &= p(p_1, r_1 = 1, Y_0) \\
&= p(p_1, r_1 = 1)p(Y_0) \\
&= p(p_1, r_1 = 1) \int dp_0 p(Y_0|p_0)p(p_0) \\
&= p(p_1, r_1 = 1) \int dp_0 \alpha_{0|0}(p_0) \\
&= l_1 p_1^{(a-1)}(1 - p_1)^{(b-1)} * c_0 \\
&\text{This is another Beta potential of the form} \\
&\sim (a, b, l_1 * c_0)
\end{aligned} \tag{2.49}$$

For $r_1 = 0$, by the definition of our change point model, p_1 is equal to p_0.

$$\begin{aligned}
\alpha_{1|0}(p_1, r_1 = 0) &= p(p_1, r_1 = 0, Y_0) \\
&= \int dp_0 p(p_1, r_1 = 0|p_0)\alpha_{0|0}(p_0) \\
&= \int dp_0 (1 - l_1)\delta(p_1 - p_0)\alpha_{0|0}(p_0) \\
&\text{This can be represented by the Beta potential} \\
&\sim (a, b, (1 - l_1) * c_0)
\end{aligned} \tag{2.50}$$

From now on, let us use Beta potential notation for the $\alpha_{t|t-1}$ messages. For change point situations, i.e. for $r_t = 1$ due to the independence of subsequent time steps, it behaves as the beginning of a new time series. You can see the path in Figure 2.8. Note the absence of arrows to the c nodes in Figure 2.8. This is to emphasize the start of a new independent series (also no category dependence between time steps in ER graphs).

$$\begin{aligned}
&r_t = 1 \\
&\alpha^0_{t|t-1} \sim (a^0_{t|t-1}, b^0_{t|t-1}, c^0_{t|t-1})
\end{aligned} \tag{2.51}$$

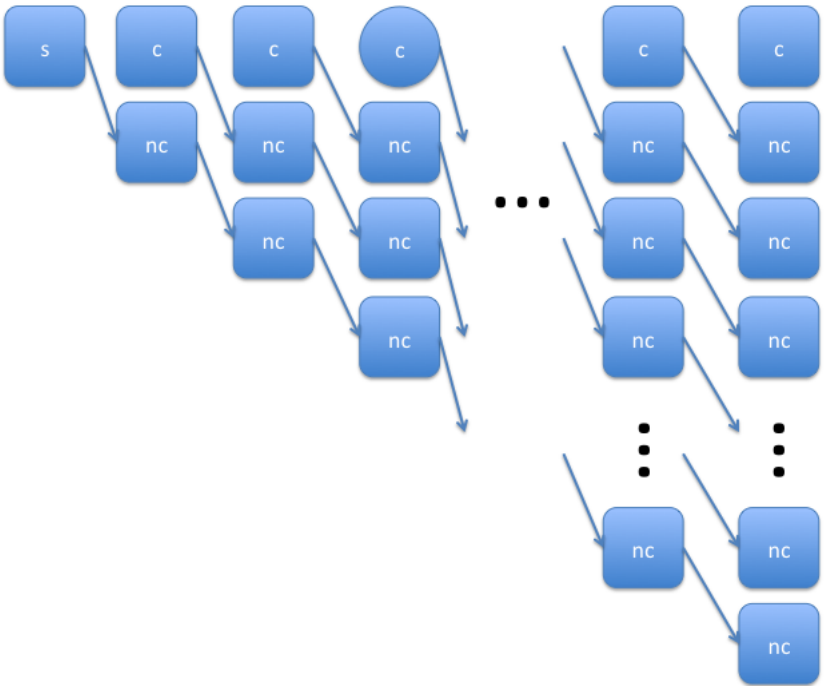


Figure 2.8. Possible change point paths for a give time series data. c: change point, nc: no change point, s:start. Node indicated with a circle has the paths: $s-c-c-c$, $s-nc-c-c$, $s-nc-nc-c$, $s-c-nc-c$.

For $r_t = 0$ there are t different previous possible paths from the beginning $t = 0$. For each possible path we define

$$\begin{aligned} &\text{for } \tau = 1..t \\ &\alpha^{\tau}_{t|t-1} \sim (a^{\tau}_{t|t-1}, b^{\tau}_{t|t-1}, c^{\tau}_{t|t-1}) \end{aligned} \tag{2.52}$$

Now let us derive the closed form equations for $\alpha^{\tau=0:t}_{t|t}$ messages.

$$\begin{aligned} \alpha_{1|1}(p_1, r_1) = p(Y_0 : 1, p_1, r_1) &= p(Y_1|r_1, p_1)\alpha_{1|0}(r_1, p_1) \\ &= p_1^{n_1}(1-p_1)^{n_1^{'}}(a_{1|0}, b_{1|0}, c_{1_0}) \end{aligned} \tag{2.53}$$

This equation can be written as multiplication of two Beta potentials.

$$\begin{aligned}\alpha_{1|1}(p_1, r_1) &= p_1^{n_1}(1-p_1)^{n_1^{'}}(a_{1|0}, b_{1|0}, c_{1_0}) \\ &= (\nu_1, \mu_1, \rho_1)(a_{1|0}, b_{1|0}, c_{1_0}) \\ &\qquad\qquad \text{where} \\ &\qquad\qquad \nu_t = n_t + 1 \\ &\qquad\qquad \mu_t = n_1^{'} + 1 \\ &\qquad\qquad \rho_t = \frac{\Gamma(\nu_t + \mu_t)}{\Gamma(\nu_t)\Gamma(\mu_t)}\end{aligned} \tag{2.54}$$

In addition, product of two beta potentials also result in a beta potential. For example assume two beta potentials (a_1, b_1, c_1) and (a_2, b_2, c_2) Then,

$$\begin{aligned}a_{new} &= a_1 + a_2 - 1 \\ b_{new} &= b_1 + b_2 - 1 \\ c_{new} &= c_1 c_2 \frac{\Gamma(a_{new})\Gamma(b_{new})}{\Gamma(a_{new} + b_{new})} \frac{\Gamma(a_1 + b_1)\Gamma(a_2 + b_2)}{\Gamma(a_1)\Gamma(a_2)\Gamma(b_1)\Gamma(b_2)}\end{aligned} \tag{2.55}$$

Let us derive the $\alpha_{1|1}(r_1, p_1)$ for the paths $r_1 = 1$ and $r_1 = 0$ in other terms, $\alpha_{1|1}^{\tau=0}(r_1 = 1, p_1)$ and $\alpha_{1|1}^{\tau=1}(r_1 = 0, p_1)$.

$$\begin{aligned}\alpha_{1|1}^0 &= (\nu_1, \mu_1, \rho_1)(a_{1|0}^0, b_{1|0}^0, c_{1|0}^0) \text{ for } r_1 = 1 \\ \alpha_{1|1}^1 &= (\nu_1, \mu_1, \rho_1)(a_{1|0}^1, b_{1|0}^1, c_{1|0}^1) \text{ for } r_1 = 0\end{aligned} \tag{2.56}$$

As stated before, these are also beta potentials. Before generalizing the derivations for all t, let us derive also for $t = 2$. For $t = 2$ Now, the paths can be states as below:

$$\begin{aligned}&r_1 = 1, r_2 = 1, \text{or} \\ &r_1 = 0, r_2 = 1, \text{or} \\ &r_1 = 1, r_2 = 0, \text{or} \\ &r_1 = 0, r_2 = 0\end{aligned} \tag{2.57}$$

For $r_2 = 1$,

$$\begin{aligned}
\alpha_{2|1}(r_2 = 0, p_2) = \alpha^0_{2|1} &= p(r_2 = 1, p_2, Y_1) \\
&= \sum_{r_1} \int dp_1 \alpha_{1|1}(r_1, p_1) p(r_2, p_2 | r_1, p_1) \\
&= \sum_{r_1} \int dp_1 \alpha_{1|1}(r_1, p_1) p(r_2, p_2) \\
&= l_1 Beta(p_2; a, b, 1) \int dp_1 (\alpha^0_{1|1} + \alpha^1_{1|1}) \\
&\quad \text{Since integration of beta potentials results in the normalization} \\
&\quad \text{constants,} \\
&= l_1 Beta(p_2; a, b, 1)(c^0_{1|1} + c^1_{1|1}) \\
&= Beta(a, b, l_1(c^0_{1|1} + c^1_{1|1}))
\end{aligned} \tag{2.58}$$

For $r_2 = 0$, we have two $\alpha_{2|1}$ messages for the two different routes. This is due to the dependence between the p_2 and p_1.

$$\begin{aligned}
\alpha_{2|1}(r_2 = 0, p_2) for r_1 = 0 = \alpha^1_{2|1} &= p(r_2 = 0, p_2, Y_1 | r_1 = 1) \\
&= \int dp_1 \alpha^0_{1|1} p(r_2, p_2) \\
&= (1 - l_1)(a^0_{1|1}, b^0_{1|1}, c^0_{1|1}) \\
&= (a^0_{1|1}, b^0_{1|1}, c^0_{1|1}(1 - l_1))
\end{aligned} \tag{2.59}$$

$$\begin{aligned}
\alpha_{2|1}(r_2 = 0, p_2) \text{ for } r_1 = 1 = \alpha^2_{2|1} &= p(r_2 = 0, p_2, Y_1 | r_1 = 0) \\
&= \int dp_1 \alpha^1_{1|1} p(r_2, p_2) \\
&= (1 - l_1)(a^1_{1|1}, b^1_{1|1}, c^1_{1|1}) \\
&= (a^1_{1|1}, b^1_{1|1}, c^1_{1|1}(1 - l_1))
\end{aligned}$$

$\alpha_{2|2}(p_2, r_2)$ messages are derived easily as the product of beta potentials:

$$\begin{aligned}\alpha_{2|2}(r_2, p_2) &= p(r_2, p_2, Y_{1:2}) \\ &= \alpha_{2|1}(r_2, p_2) p(Y_2|p_2, r_2) \\ &\text{As derived for } \alpha_{1|1}, p(Y_2|p_2, r_2) \text{ is a beta potential of the form } (\nu_2, \mu_2, \rho_2) \\ &\text{for } \tau = 0 : t \\ &= (a^\tau_{2|1}, b^\tau_{2|1}, c^\tau_{2|1})(\nu_2, \mu_2, \rho_2)\end{aligned} \tag{2.60}$$

Generalizing for $t > 0$,

$$\begin{aligned}\alpha_{t|t}(r_t, p_t) &= p(r_t, p_t, Y_{1:t}) = \alpha_{t|t-1}(r_t, p_t) p(Y_t|p_t, r_t) \\ &\text{for } \tau = 0 : t \\ &= (a^\tau_{t|t-1}, b^\tau_{t|t-1}, c^\tau_{t|t-1})(\nu_t, \mu_t, \rho_t)\end{aligned} \tag{2.61}$$

$$\begin{aligned}\alpha_{t|t-1}(r_t, p_t) &= p(r_t, p_t, Y_{1:t-1}) \\ &= \int dp_{t-1} \sum_{r_{t-1}} p(p_t, r_t|p_{t-1}) \alpha_{t-1|t-1}(p_{t-1}, r_{t-1}) \\ &= \alpha^0_{t|t-1}(r_t = 1, p_t) = (a, b, l_1 \sum_{\tau=0}^{t-1} c^\tau_{t-1|t-1}) \\ &for\ \tau = 1 : t \\ &\alpha^\tau_{t|t-1}(r_t = 0, p_t) = (a^{\tau-1}_{t-1|t-1}, b^{\tau-1}_{t-1|t-1}, (1 - l_1) c^{\tau-1}_{t-1|t-1})\end{aligned} \tag{2.62}$$

2.3.1.2. Derivation of Backward Messages. Let us now derive the backward messages for our multiple change point model on SBMs.

$$\begin{aligned}\beta_{T|T+1}(p_T, r_T) &= p(Y_{T+1:T}|p_T, r_T) = 1 \\ &\text{This can be represented by Beta potential } (1, 1, 1) \\ &(\hat{a}_{T|T+1}, \hat{b}_{T|T+1}, \hat{c}_{T|T+1}) = (1, 1, 1)\end{aligned} \tag{2.63}$$

$$
\begin{aligned}
\beta_{T|T}(p_T, r_T) &= p(Y_{T:T}, p_T, r_T) \\
&= p(Y_t|r_T, p_T)\beta_{T|T+1}(p_T, Y_T) \\
&\text{Joint probability of Erdös-Réyni graph given } r_T \text{ and } p_T \\
&\text{is a Beta potential } (\nu_T, \mu_T, \rho_T) \text{ as derived before} \\
&\text{Since } \beta_{T|T+1}(r_T = 1, p_T) = \beta_{T|T+1}(r_T = 0, p_T) \\
&\beta_{T|T}(r_T = 0, p_T) = \beta_{T|T}(r_T = 1, p_T) \\
&= (\nu_T, \mu_T, \rho_T)(1, 1, 1) \\
&= (\hat{a}_{T|T}, \hat{b}_{T|T}, \hat{c}_{T|T}) \\
&\text{call this } \beta^0_{T|T}(r_T, p_T) \text{ to make recursion easy}
\end{aligned} \tag{2.64}
$$

Let us now continue for $T-1$

$$
\begin{aligned}
\beta_{T-1|T}(p_{T-1}, r_{T-1}) &= \int dp_T \sum_{r_T} p(r_T, p_T|p_{T-1})\beta_{T|T}(p_T, r_T) \\
&= \int dp_T p(r_T = 1) Beta(P_T; \alpha, \beta)\beta_{T|T}(p_T, r_T = 1) \\
&+ \int dp_T p(r_T = 0)\delta(p_T - p_{T-1})\beta_{T|T}(p_T, r_T = 0)
\end{aligned} \tag{2.65}
$$

Now, we can begin to calculate β messages for different paths similar to α messages. As seen from the above equation value of r_{T-1} does not effect $\beta_{T-1|T}$ messages. Here paths represents the consecutive time step r_T:

$$
\begin{aligned}
\beta^0_{T-1|T}(p_{T-1}, r_{T-1}) &= \int dp_T \; p(r_T = 1) Beta(P_T; \alpha, \beta)\beta_{T|T}(p_T, r_T) \\
&= \int dp_T \; p_1(\alpha, \beta, 1)(\hat{a}_{T|T}, \hat{b}_{T|T}, \hat{c}_{T|T}) \\
&= (1, 1, p_1 \; c[(\alpha, \beta, 1)(\hat{a}_{T|T}, \hat{b}_{T|T}, \hat{c}_{T|T})])
\end{aligned} \tag{2.66}
$$

We define $c[(a, b, c)] = c$ for a Beta potential.

$$
\begin{aligned}
\beta^1_{T-1|T}(p_{T-1}, r_{T-1}) &= \int dp_T \; p(r_T = 0)\delta(p_T - p_{T-1})\beta_{T|T}(p_T, r_T) \\
&= \int dp_T \; p_0 \; \delta(p_T - p_{T-1})(\hat{a}_{T|T}, \hat{b}_{T|T}, \hat{c}_{T|T}) \\
&= (\hat{a}_{T|T}, \hat{b}_{T|T}, p_0 \; \hat{c}_{T|T})
\end{aligned} \tag{2.67}
$$

$$
\begin{aligned}
\beta_{T-1|T-1}(p_{T-1}, r_{T-1}) &= p(Y_{T-1}|p_{T-1})\beta_{T-1|T}(p_{T-1}, r_{T-1}) \\
\beta^0_{T-1|T-1}(p_{T-1}, r_{T-1}) &= (\nu_{T-1}, \mu_{T-1}, \rho_{T-1})\beta^0_{T-1|T}(p_{T-1}, r_{T-1}) \\
&= (\nu_{T-1}, \mu_{T-1}, p_1\ c[(\alpha, \beta, 1)(\hat{a}_{T|T}, \hat{b}_{T|T}, \hat{c}_{T|T})]) \\
\beta^1_{T-1|T-1}(p_{T-1}, r_{T-1}) &= (\nu_{T-1}, \mu_{T-1}, \rho_{T-1})\beta^1_{T-1|T}(p_{T-1}, r_{T-1}) \\
&= (\nu_{T-1}, \mu_{T-1}, \rho_{T-1})(\hat{a}_{T|T}, \hat{b}_{T|T}, p_0\ \hat{c}_{T|T}) \\
&Here\ p_{T-1} = p_T \\
&= (\hat{a}^0_{T|T}, \hat{b}^0_{T|T}, p_0\ \hat{c}^0_{T|T})
\end{aligned}
\tag{2.68}
$$

After derivation of forward and backward messages, we see that equation $p(r_t|Y_{1:T}) = \int_{p_t} \alpha_{t|t-1}(r_t, p_t)\beta_{t|t}(r_t, p_t)$ is in fact integration of multiplication of two beta potentials. Multiplication of beta potentials is another beta potential. Thus, posterior distribution is also proportional to a beta potential.

2.3.2. Forward Filtering- Backward Sampling Algorithm

In forward-backward algorithm, α and β messages are calculated for each possible value of each hidden variable at each time step. In order to reduce computational complexity we can use Monte Carlo sampling. As shown in the previous section, $\alpha_t|t(p_t, r_t)$ messages are the filtering densities $p(p_t, r_t, Y_{1:t})$. If we apply Bayes rule to $p(p_T, r_T|Y_{1:T})$ we observe the following:

$$
\begin{aligned}
p(p_T, r_T|Y_{1:T}) &= \frac{p_T, r_T, Y_{1:T}}{p(Y_{1:T})} \\
&\text{Since } Y_{1:T} \text{ is observed, } p(Y_{1:T}) \text{ is constant} \\
&\sim p(p_T, r_T, Y_{1:T}) = \alpha_{T|T}(p_T, r_T)
\end{aligned}
\tag{2.69}
$$

Then we can sample the change point r_T at time step T. Then we can calculate the backward message for the samples r_T. Then we can continue with $T-1$ and calculate $\beta_{T-1|T-1}$ messages for $r_{T-1} = 0$ and $r_{T-1} = 1$ by using the sampled r_T value. Then we can calculate the smoothed probabilities for r_{T-1} values and sample it from these probabilities. If enough samples are taken, distribution formed by the samples converges to exact inference probabilities. (This is shown in experiments section.)

Let us now derive the backward sampling equations: In backward sampling, beginning from T, we sample $r_{t+1:T}$. For any time step t, by backward sampling our aim is to calculate the posterior distribution $p(r_t|Y_{1:T}, r_{t+1:T})$. We can calculate that probability in the following way:

$$\begin{aligned} p(r_{t-1}|A_{1:T}, r_{t:T}) &= \int dp_t dp_{t-1} p(p_{t-1}, p_t, r_{t-1}|A_{1:T}, r_{t:T}) \\ &= \int dp_t dp_{t-1} p(p_{t-1}, r_{t-1}|p_t, r_{t:T}, A_{1:T}) p(p_t|A_{1:T}, r_{t:T}) \end{aligned} \tag{2.70}$$

Let us work on these probabilities separately in order to obtain more closed expressions.

$$\begin{aligned} p(p_t|Y_{1:T}, r_{t:T}) &= \frac{p(Y_{1:T}, r_{t:T}, p_t)}{p(Y_{1:T}, r_{t:T})} \\ &\text{Since } Y_{1:T}, r_{t:T} \text{ are constants} \\ &\sim p(Y_{1:T}, r_{t:T}, p_t) \\ &= p(Y_{t:T}, r_{t:T}, p_t) p(Y_{1:T-1}, r_{t:T}, p_t) \\ &= \frac{p(Y_{t:T}, r_t, p_t)}{p(r_{t+1:T})} p(Y_{1:t-1}, r_t, p_t) \\ &\text{by definition this is equal to} \\ &= \beta_{t|t}(r_t, p_t)\ \alpha_{t|t-1}(r_t, p_t) \end{aligned} \tag{2.71}$$

Let us now work on the second part

$$\begin{aligned} p(p_{t-1}, r_{t-1}|p_t, r_{t:T}, Y_{1:T}) &= p(p_{t-1}, r_{t-1}|p_t, r_{t:T}, Y_{1:t-1}) \\ &= \frac{p(p_t, r_t|p_{t-1}, r_{t-1}, Y_{1:t-1}) p(p_{t-1}, r_{t-1}|Y_{1:t-1})}{p(p_t, r_t|Y_{1:t-1})} \\ &= \frac{p(p_t, r_t|p_{t-1}, r_{t-1}) p(p_{t-1}, r_{t-1}|Y_{1:t-1})}{p(p_t, r_t|Y_{1:t-1})} \\ &\text{multiply both sides with } p(Y_{1:t-1}) \\ &= \frac{p(p_t, r_t|p_{t-1}, r_{t-1}) p(p_{t-1}, r_{t-1}|Y_{1:t-1}) p(Y_{1:t-1})}{p(p_t, r_t|Y_{1:t-1}) p(Y_{1:t-1})} \end{aligned}$$

we obtain

$$= \frac{p(p_t, r_t|p_{t-1}, r_{t-1})p(p_{t-1}, r_{t-1}, Y_{1:t-1})}{p(p_t, r_t, Y_{1:t-1})} = \frac{p(p_t, r_t|p_{t-1}, r_{t-1})\alpha_{t-1|t-1}}{\alpha_{t|t-1}(r_t, p_t)} \tag{2.72}$$

Now we can return to $p(r_{t-1}|A_{1:T}, r_{t:T})$:

$$\begin{aligned} p(r_{t-1}|Y_{1:T}, r_{t:T}) &= \int dp_t dp_{t-1} p(p_{t-1}, r_{t-1}|p_t, r_{t:T}, Y_{1:T}) p(p_t|Y_{1:T}, r_{t:T}) \\ &= \int dp_t dp_{t-1} \frac{p(p_t, r_t|p_{t-1}, r_{t-1})\alpha_{t-1|t-1}(r_{t-1}, p_{t-1}}{\alpha_{t|t-1}(r_{t-1}, p_{t-1})} \ \beta_{t|t}(r_t, p_t) \ \alpha_{t|t-1}(r_t, p_t) \\ &= \int dp_t dp_{t-1} p(p_t, r_t|p_{t-1}, r_{t-1})\alpha_{t-1|t-1}(r_{t-1}, p_{t-1}) \ \beta_{t|t}(r_t, p_t) \end{aligned} \tag{2.73}$$

$$\begin{aligned} &\text{If } r_{t-1} = 1, \ p(r_{t-1} = 1|Y_{1:T}, r_{t:T}) = \\ &\qquad \int\int dp_t \ dp_{t-1} \ l_1 Beta(p_t; \alpha, \beta)(a_{t-1|t-1}, b_{t-1|t-1}, c_{t-1|t-1})(\hat{a}_{t|t}, \hat{b}_{t|t}, \hat{c}_{t|t}) \\ &\text{If } r_{t-1} = 0, \ p(r_{t-1} = 0|Y_{1:T}, r_{t:T}) = \\ &\qquad \int\int dp_t \ dp_{t-1} (1 - l_1)(a_{t-1|t-1}, b_{t-1|t-1}, c_{t-1|t-1})(\hat{a}_{t|t}, \hat{b}_{t|t}, \hat{c}_{t|t}) \end{aligned} \tag{2.74}$$

Integration of these potentials result in the summation of normalization constants of Beta potentials. Then we can draw random change points proportional to these posterior probabilities.

2.3.3. Inference with Gibbs Sampling

So far, while making inference about change points $r_{1:T}$, we assumed that category assignments are known and does not change. By this assumption, we assumed SBM is divided into correct ERG graphs. And then for each ERG time series, we make inference for change points separately. In this section, we will explain how to handle

all the unknown parameters together.

Let us rewrite our objective function:

$$\begin{aligned} p(r_{1:T}|Y_{1:T}) &= \sum_{C^u}\sum_{C^s}\int d\beta_{1:T}\ p(r_{1:T},\beta_{1:T},C^u,C^s|Y_{1:T}) \\ &= \sum_{C^u}\sum_{C^s}\int d\beta_{1:T}\ p(r_{1:T},\beta_{1:T}|C^u,C^s,Y_{1:T})\ p(C^u,C^s|Y_{1:T}) \end{aligned} \tag{2.75}$$

In order to integrate out category assignments (C^u and C^s), we will integrate forward filtering-backward sampling algorithm with Gibbs sampling together. We can sample $r_{1:T}$ by forward filtering-backward sampling algorithm as in the previous section. Taking our graphical model into account we can summarize collapsed Gibb's sampler merged with forward filtering-backward sampling as follows (Algorithm in Figures 2.9 and 2.10):

for all $p \in 1,...,N^u$ **do**
  sample $C^u_p \sim Multn(\pi^u)$
  for all $q \in 1,...,N^s$ **do**
    sample $C^s_q \sim Multn(\pi^s)$
    for all $t \in 1,...,T$ **do**
      sample $r_t \sim Be(p)$
      {Gibbs Sampling}
      for all $i \in 1,...,epoch$ **do**
        for all $t \in 1,...,T$ **do**
          Inner Gibb's Sampling()
        end for
      end for
    end for
  end for
end for

Figure 2.9. Forward Filtering–Backward Sampling.

Procedure: **Inner Gibb's Sampling()**

for all $k \in 1, .., K^u$ **do**

  for all $l \in 1, .., K^s$ **do**

    $A^i_{t,kl}$ =Subgraph formed by nodes belonging to k and l categories.

    for all $A^i_{t,kl}$ **do** {Forward Filtering}

      calculate beta potentials of messages: $\alpha^i_{t,kl}(r_{t,kl})$

      $\beta_{T|T+1} = 1$ {Backward Sampling}

      for all $t \in T, T-1.., 1$ **do**

        s^i_t: $\alpha^i_{t|t}(r_t, kl) * \beta^i_{t|t+1}(r_{t,kl})$

        Sample change point for time slice t: $r^i_{t,kl} \sim s^i_t$

        For sampled $r_{t,kl}$, calculate potential $\beta^i_{t-1|t}(r^i_{t-1,kl})$

      end for

    end for

  end for

end for

for all $p \in 1, .., N^u$ **do**

  Sample $C^{u(i)}_p \sim p(C^{u(i)}_p | Y_{1:T}, C^{u(i)}_{p^-}, C^{s(i-1)}, r^{(}_{1:T} i))$

end for

for all $q \in 1, ..., N^s$ **do**

  Sample $C^{s(i)}_q \sim p(C^{s(i)}_q | Y_{1:T}, C^{s(i)}_{q^-}, C^{u(i)}, r^{(}_{1:T} i))$

end for

for all $t \in 1, ..., T$ **do**

  for all $k \in 1, ..., K^u$ **do**

    for all $l \in 1, ..., K^s$ **do**

      calculate prob $r_t(k, l)$

      $p(r_{t,kl} = 1) = \frac{1}{EPOCH} \sum_i^{epoch} [r_{t,kl} = 1]$

      $p(r_{t,kl} = 0) = \frac{1}{EPOCH} \sum_i^{epoch} [r_{t,kl} = 0]$

    end for

  end for

end for

Figure 2.10. Gibbs Sampling Sampling For epoch i and time step t.

3. EXPERIMENTS

3.1. On Synthetic Data

3.1.1. Change Points on Erdös-Réyni Graph Time Series

In order to compare forward-backward algorithm with the proposed backward-sampling approach, we begin with relatively easy case: Detection of change points in ER time series. As mentioned before, change point at a time step means, change in the edge probability.

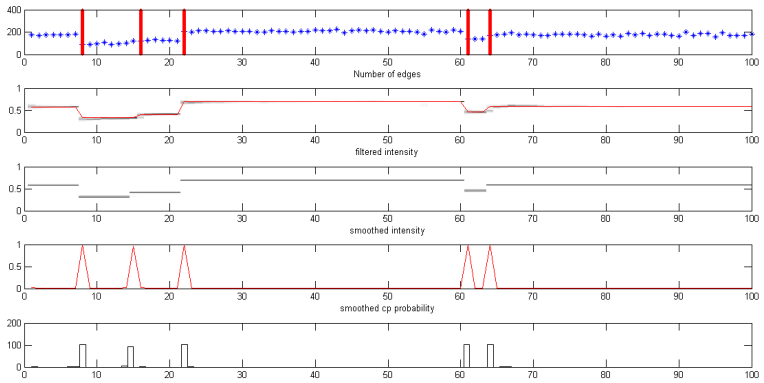
We will compare the posterior probability of change points obtained by forward-backward algorithm and forward filtering-backward sampling algorithm.

We have generated synthetic data for the following scenario:

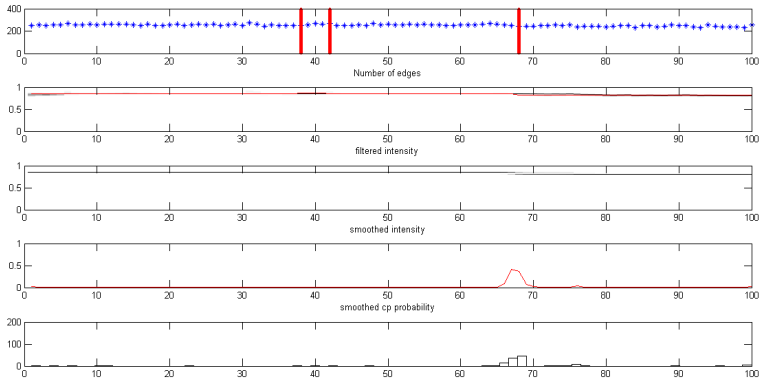
$$
\begin{aligned}
T &= 100 \text{ number of time steps} \\
p &= beta(\alpha, \beta) \text{ edge probability} \\
\alpha &= 1,\ \beta = 1 \text{ uninformative beta distribution parameters} \\
N^u &= 20 \text{ number of nodes of type 1} \\
N^s &= 15 \text{ number of nodes of type 2} \\
p_1 &= 0.02 \text{ change point probability} \\
epoch &= 200
\end{aligned}
$$

The underlying generative model is as in the Equation 2.37. Results of forward-backward algorithm and forward filtering-backward sampling algorithm is as follows for three different runs:

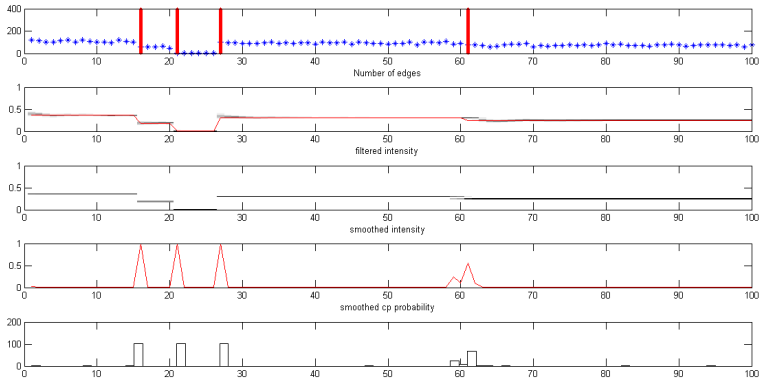
In each experiment shown in Figure 3.1, first subgraphs show number of edges for each time step. In addition, red bars in first subgraphs show the real change point



(a) Experiment 1



(b) Experiment 2



(c) Experiment 3

Figure 3.1. Synthetic Data Experiments on ER Change Point Model.

places. Second subgraphs show the filtered values for inferred edge probabilities (p). Third subgraphs in each experiment show the smoothed values obtained by use of backward (beta) messages for inferred edge probabilities. Forth subgraphs shows the

posterior distribution for the change points obtained by forward-backward algorithm. Finally, last subgraphs shows the histogram of change points that are obtained by forward filtering - backward sampling algorithm.

When we observe the results, we see that although exact inference with forward-backward algorithm infers the change points with high probability, histogram obtained with sampling algorithm shows smaller probabilities. Thus, number of epochs used in sampling should be investigated. Another thing we observe is that, change point probability for very small changes are infered with small probabilities also with forward-backward algorithm. This is to do with total edge number (sample size) and central limit theorem (CLT). Thus we should observe the relation between infered change point probability, edge probability and number of nodes.

Each ER graph is a Bernoulli sample described by number of trials equal to maximum possible edge number, say M, and parameter p. So each ER graph's sample mean can be calculated as follows:

$$\hat{p} = \frac{\sum_{i=1}^{M} x_i}{M}$$

In addition, sampling variance can be calculated as

$$\begin{aligned} var(\hat{p}) &= \frac{\sum_{i=1}^{M}(x_i - \hat{p})^2}{M(M-1)} \\ &= \frac{M\hat{p}(1-\hat{p})^2 + (M - M\hat{p})p^2}{M(M-1)} \\ &= \frac{(M\hat{p}(1-\hat{p}))(\hat{p} + (1-\hat{p}))}{M(M-1)} \\ &= \frac{\hat{p}(1-\hat{p})}{M-1} \end{aligned}$$

According to this equation, we see that sampling variance, thus standard error which is squareroot of variance, decreases by increasing sample size. Thus, small changes in edge probabilities can be better detected for bigger node numbers, assuming same sample mean $\hat{p}$. In Figure 3.2, this relation can be observed. In order to obtain the

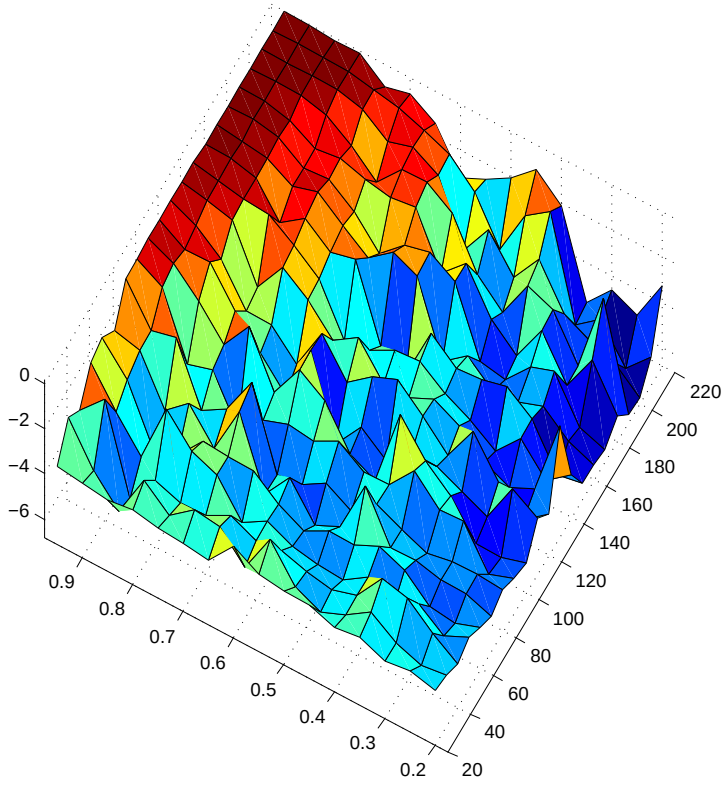
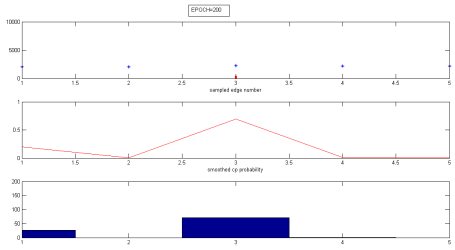


Figure 3.2. N and P relation: In order to obtain this graph we have generated a ER time series of length 5 and at time step 3 we have assigned a change point. For every node number-edge probability combination we have taken 5 runs and print the average of them.

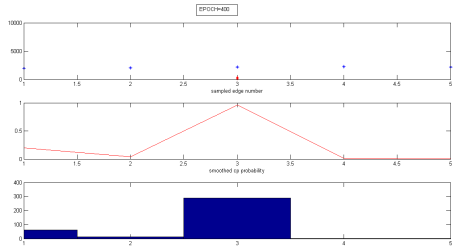
graph in 3.2 we have generated a ER time series of length 5 and at time step 3 we have assigned a change point. We took the change point probability from 0.2 to 0.98 untill time step 2 and afterward we increased change point probability by 0.02 and fixed. We have take runs for total node numbers beginning from 10+10 to 100+100 (since bipartite ER graph) increased by 10. Thus, maximum possible edge numbers differ from 100 to 10000. For every node number-edge probability combination we have taken 5 runs and print the average of them.

For forward filtering-backward sampling algorithm, epoch size is the corresponding sample size. Thus, similarly increasing epoch size also increases sample size. As the number of sample point increases estimated change points approximate better to the values inferred with exact inference algorithm. This can be observed in Figure 3.3.

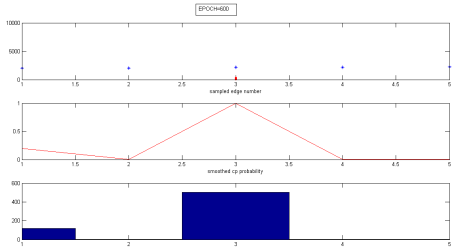
In order to obtain the effect of increasing epoch on inferred change point probability



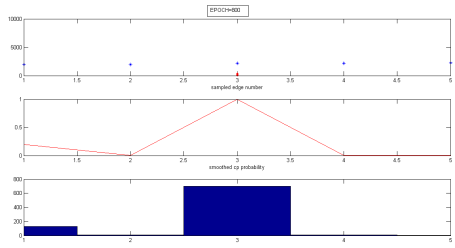
(a) Experiment 1



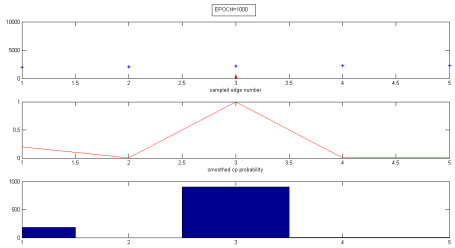
(b) Experiment 2



(c) Experiment 3



(d) Experiment 4



(e) Experiment 5

Figure 3.3. Effect of increasing epoch on inferred change point probability.

in 3.3, we have taken ER graph series of length 5. For time steps 1 and 2 edge probability p is equal to 0.2. At time step 3, we assumed a change point and change p from 0.2 to 0.22. We have taken different runs with epoch size 200, 400, 600, 800 and 1000. As seen in the subplots, increasing epoch size improves the result of forward filtering-backward sampling algorithm.

Considering epoch size and number of nodes, we have taken runs with the following scenarios:

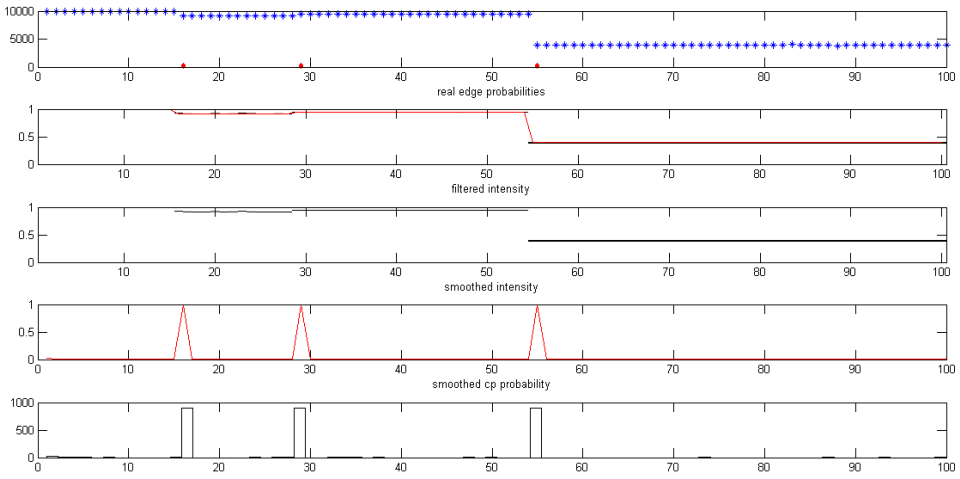
$$
\begin{aligned}
T &= 100 \text{ number of time steps} \\
p &= beta(\alpha, \beta) \text{ edge probability} \\
\alpha &= 6 \ \beta = 1 \text{ in order to observe small changes in high edge probabilities} \\
\alpha &= 1 \ \beta = 6 \text{ in order to observe small changes in low edge probabilities} \\
N^u &= 100 \text{ number of nodes of type 1} \\
N^s &= 100 \text{ number of nodes of type 2} \\
p_1 &= 0.02 \text{ change point probability} \\
epoch &= 1000
\end{aligned}
$$

Observing the change point histograms and the posterior distributions in Figure 3.4, we can conclude that proposed forward filtering-backward sampling algorithm gives results in accordance with exact inference algorithm forward-backward.

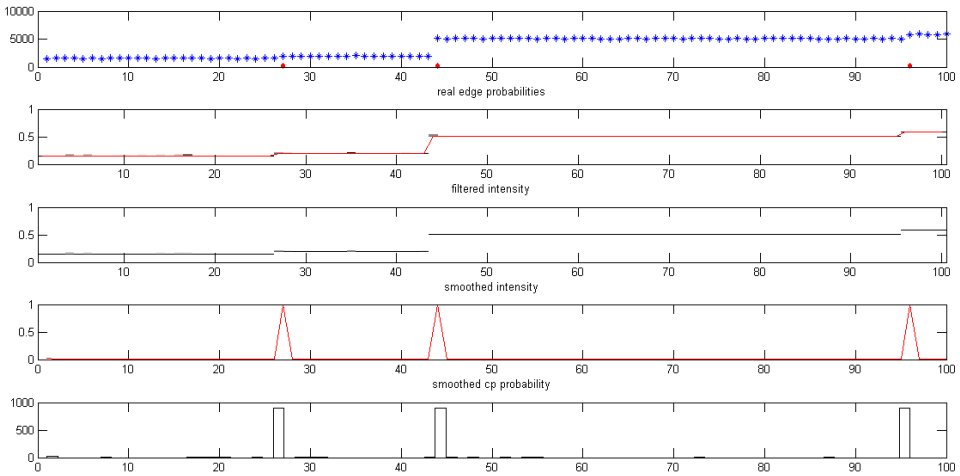
3.1.2. Change Point Detection on SBM Time Series

In this section, we have generated synthetic SBM time series with change points. Considering the node number and epoch size effects, we have generated a scenario as follows:

$$
\begin{aligned}
T &= 200 \text{ number of time steps} \\
B &= beta(\alpha, \beta) \text{ edge probability} \\
\alpha &= 1 \ \beta = 1 \text{ uninformative beta distribution parameters} \\
N^u &= 200 \text{ number of nodes of type 1} \\
N^s &= 200 \text{ number of nodes of type 2} \\
K^u &= 2 \text{ number of categories of node type 1} \\
K^s &= 2 \text{ number of categories of node type 2} \\
p_1 &= 0.04 \text{ change point probability}
\end{aligned}
\tag{3.1}
$$



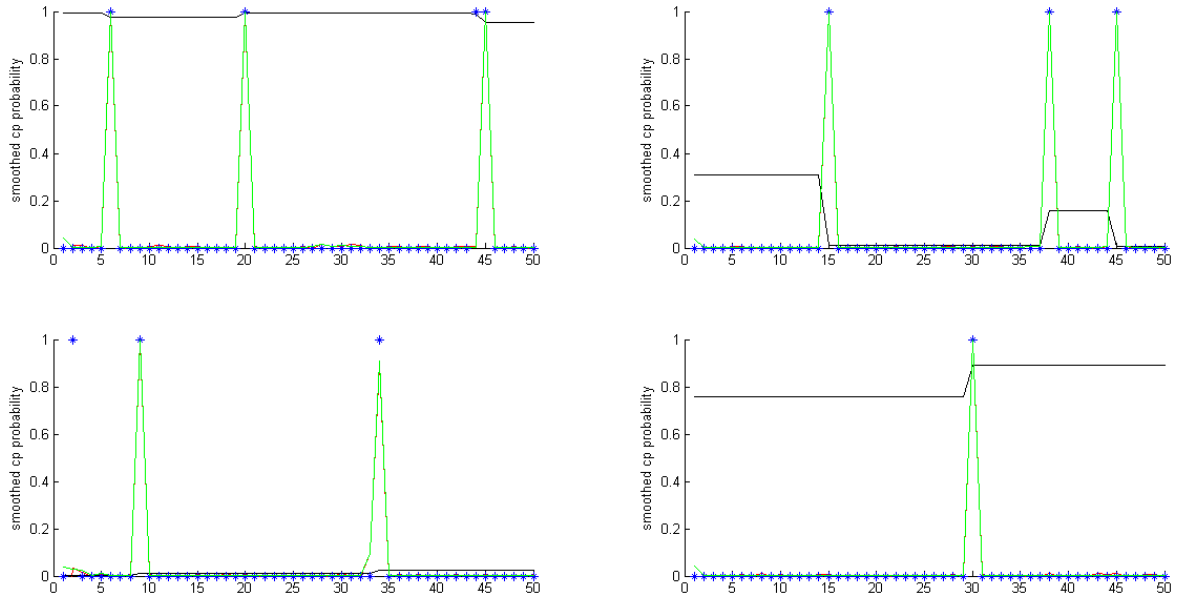
(a) Inference for small changes in high connectivity



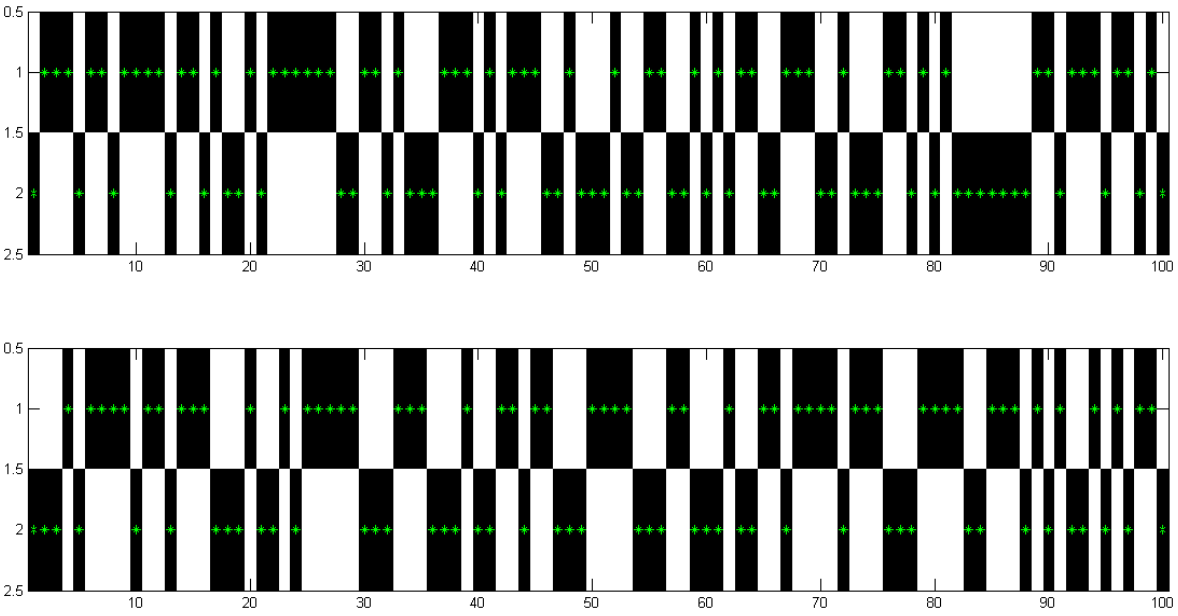
(b) Inference for small changes in low connectivity

Figure 3.4. Synthetic Data Experiments on ER Change Point Model.

In Figure 3.5, inference results for category assignments, and change points obtained by forward filtering - backward smoothing and forward filtering-backward sampling schemes can be seen.



(a) Change point inference



(b) Change point inference

Figure 3.5. Synthetic Data Experiments on SBM Change Point Model: Green line represents the posterior probability obtained by samples of forward filtering-backward sampling. (Nearly unseen) red line represents the posterior probability obtained by exact inference algorithm. In this run, results of two algorithm coincide rather well.

3.2. On Real Data

3.2.1. ENRON Data

We have tried to apply our proposed algorithm on Enron dataset[2] .

[2]http://cis.jhu.edu/ parky/Enron/enron.html

We have parsed the data into different time segments. While doing so, we have aggregated the edges on a single adjacency matrix. Figure 3.6 is formed in that way. Then we have taken runs for different number of categories and different number of ag-

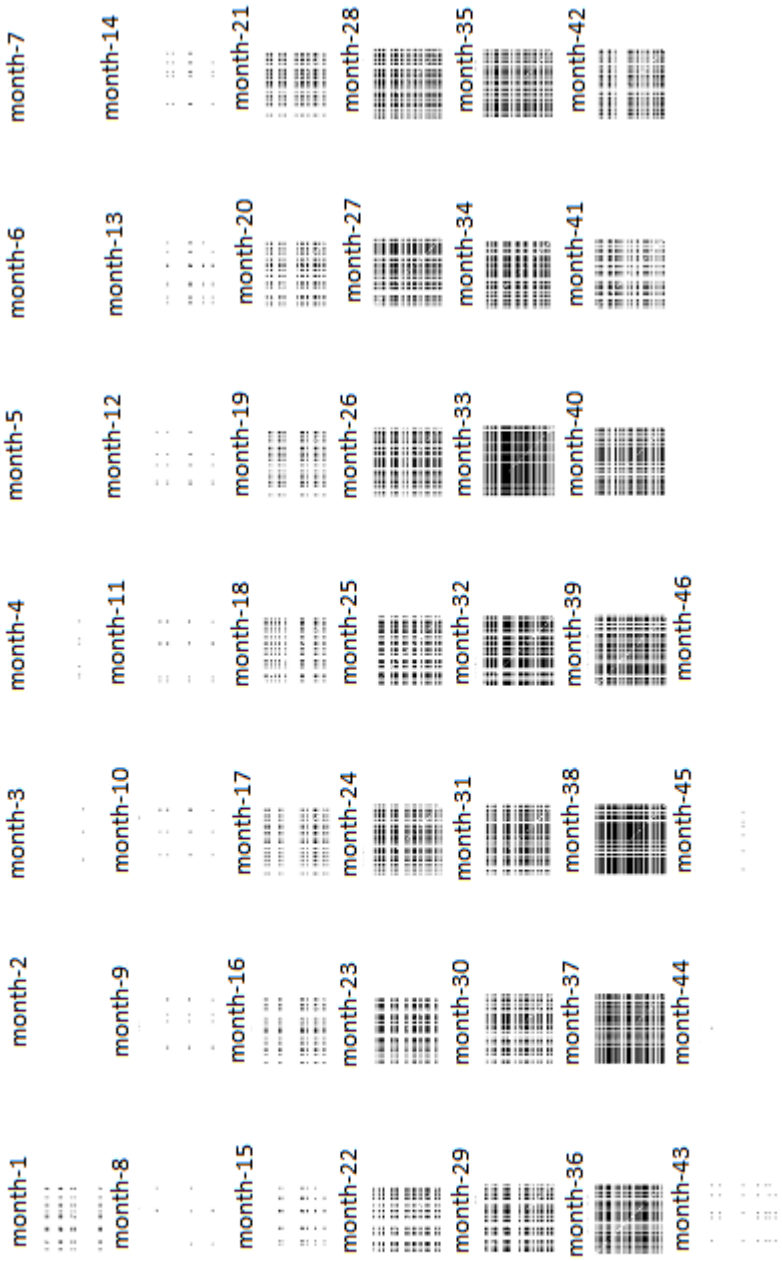


Figure 3.6. Enron monthly aggregated email data: Multiple mails between workers are counted as one or represented by a single edge.

gregation days, for example monthly, weakly, fifteen days for 2 categories, 4 categories. While aggregating time, we did not sum the number of edges, we counted once every edge. Below is the run results for different configurations: Figures 3.7, 3.8, 3.9, 3.10,

3.11 and 3.12.

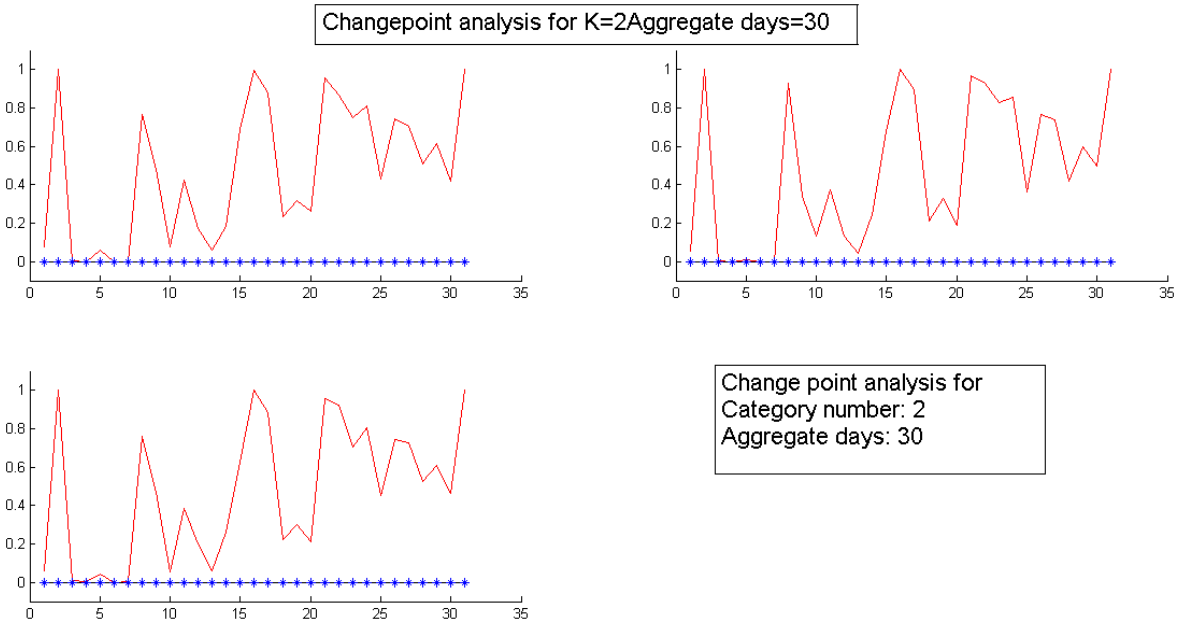


Figure 3.7. Change point analysis for aggregate day number=30 and assumed number of categories=2.

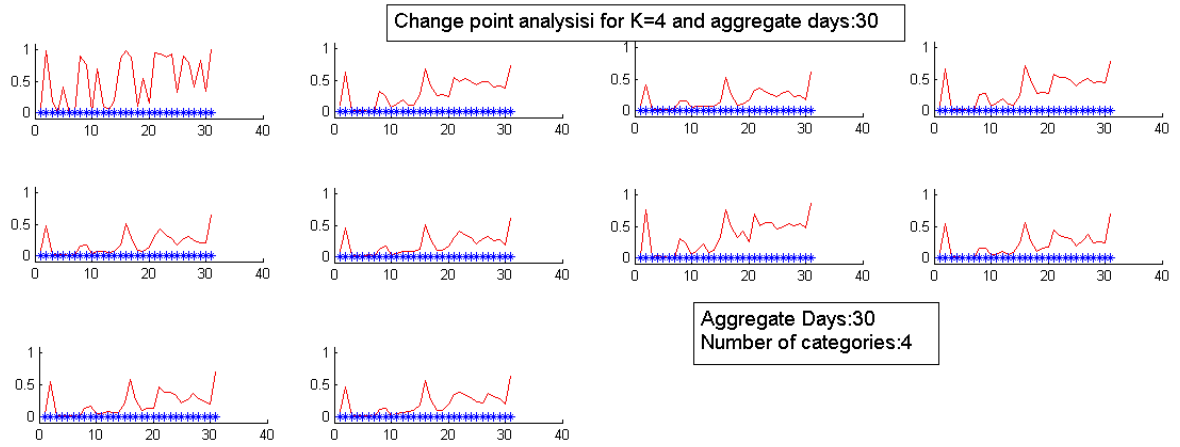


Figure 3.8. Change point analysis for aggregate day number=30 and assumed number of categories=4.

When we observe the change point probabilities and the adjacency matrices of Enron data, we see that results are in accordance with the changes in the adjacency matrix series. Considering taken runs with different configurations, although we assume different number of categories, all the graphs shows the same pattern for change point inference. This means, under our assumed model, ENRON data cannot be separated

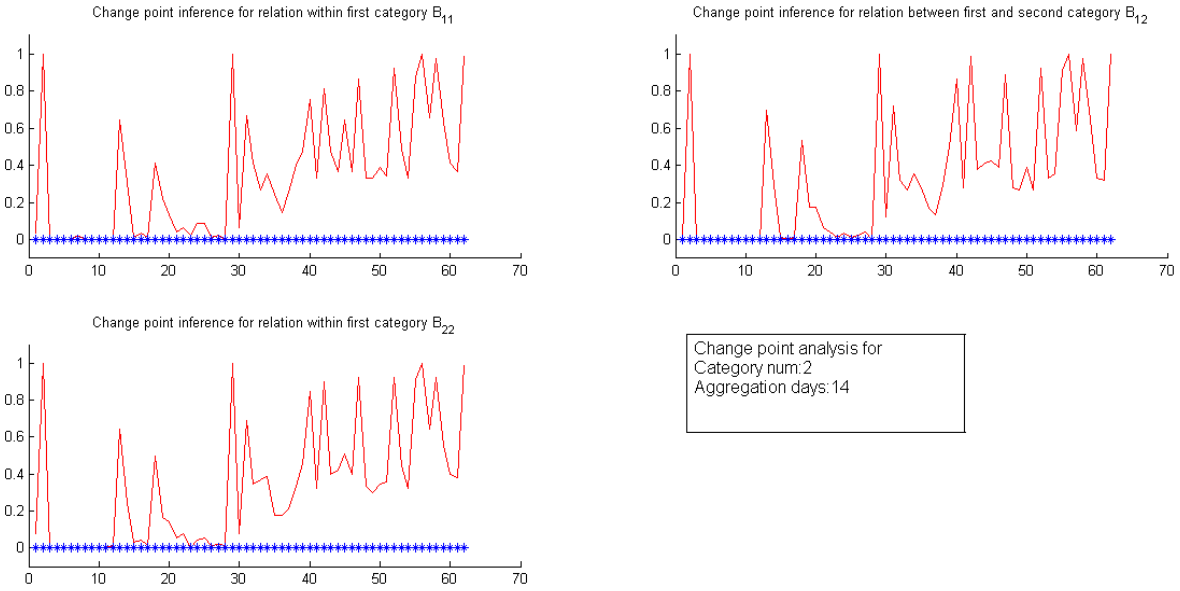


Figure 3.9. Change point analysis for aggregate day number=14 and assumed number of categories=4.

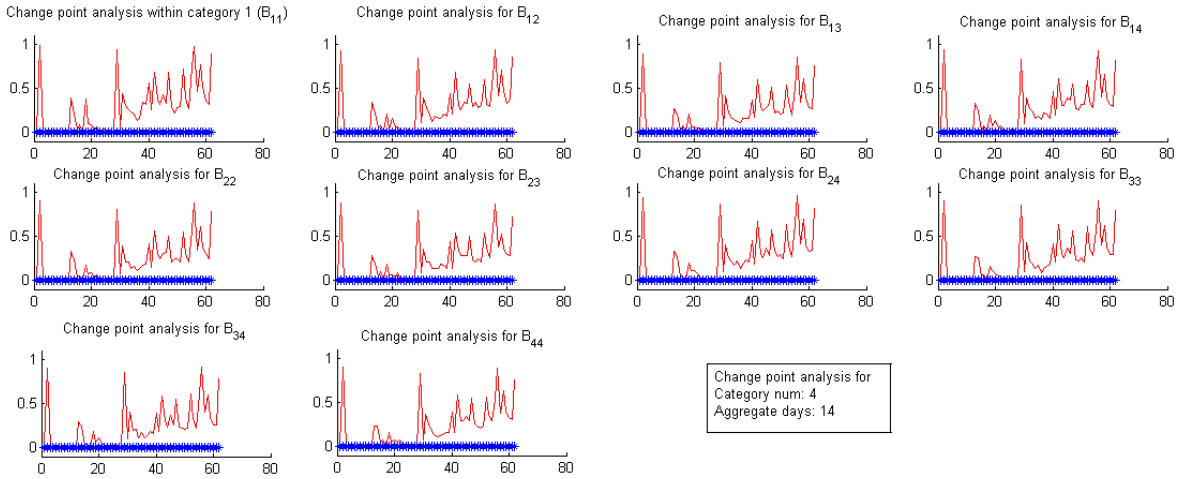


Figure 3.10. Change point analysis for aggregate day number=14 and assumed number of categories=4.

to blocks, there is only single category.

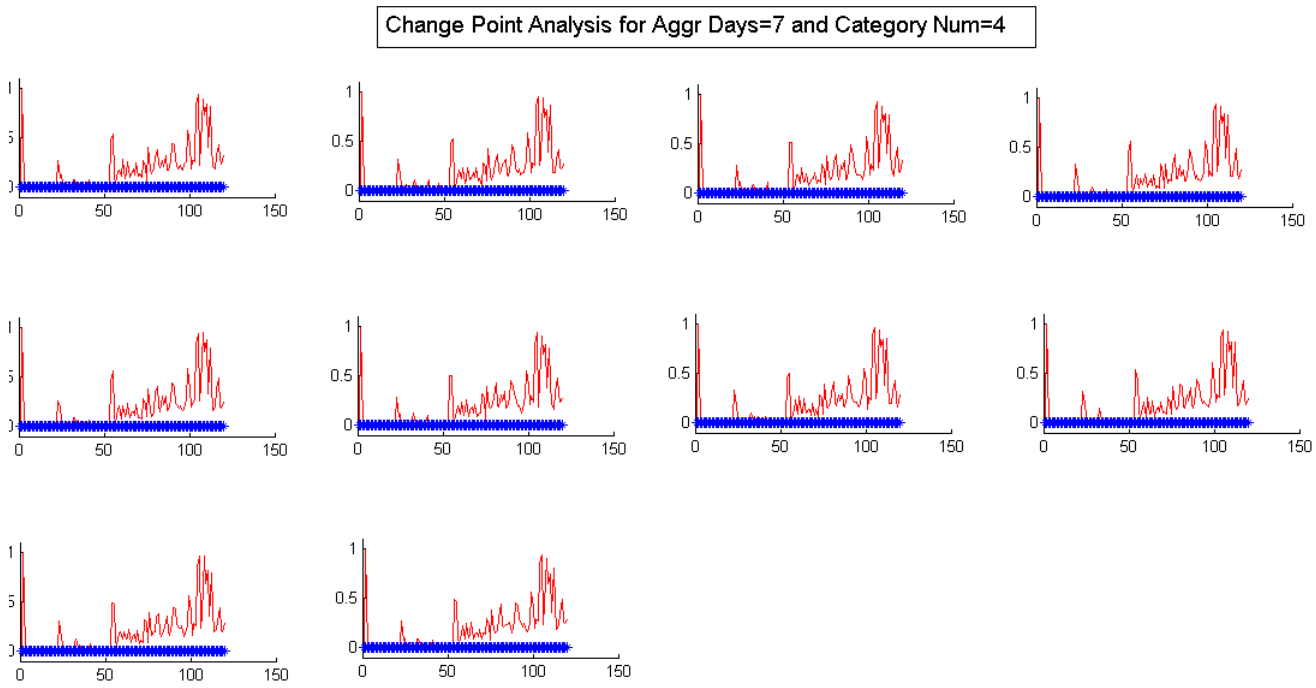


Figure 3.11. Change point analysis for aggregate day number=7 and assumed number of categories=4.

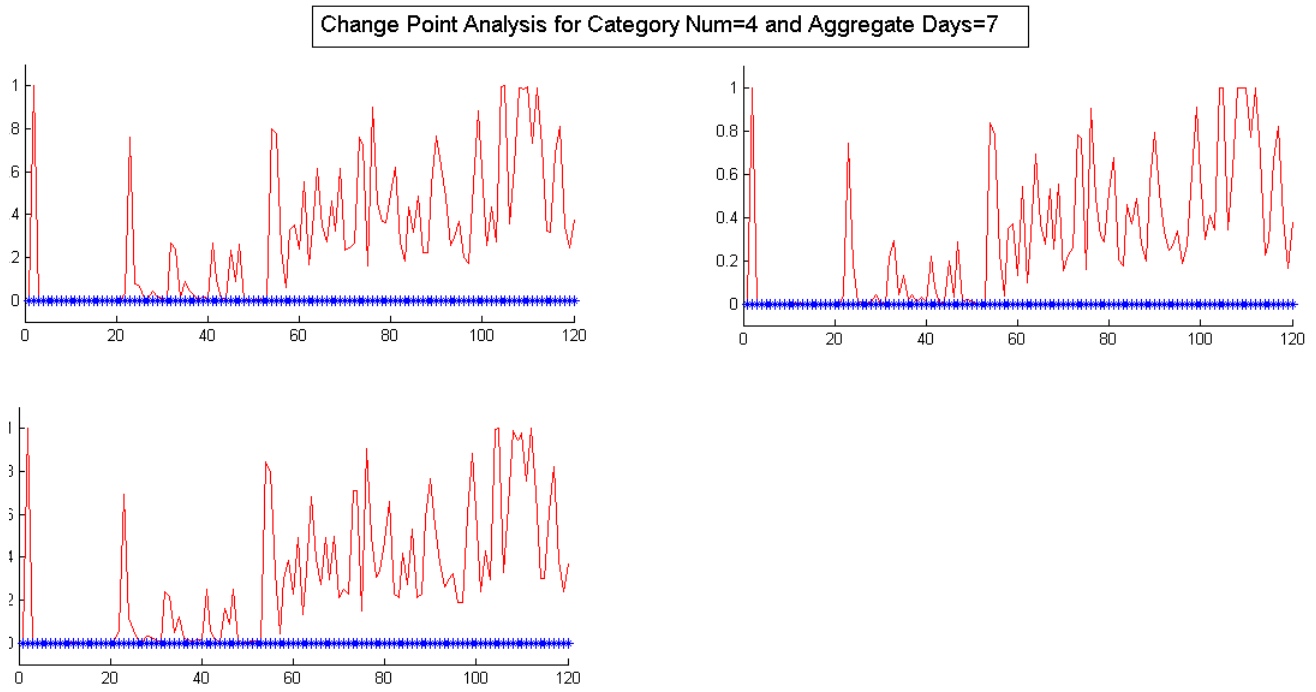


Figure 3.12. Change point analysis for aggregate day number=7 and assumed number of categories=2.

4. CONCLUSIONS AND FUTURE WORK

In this thesis work, we have studied Bayesian inference on SBMs and multiple change point detection in time series of SBMs. We have developed an approximate inference algorithm for change point detection. Our motivation was that analyzing network data is becoming more and more common recently, but analyzing time series of network data is a more rear issue.

For inference in SBM, we have applied variational EM and Gibbs sampling algorithms. We have observed that both approaches gives similar acceptable results. For multiple change point detection, firstly we have applied well-known HMM method, forward-backward algorithm. Then, we have proposed an approximate inference algorithm, by combining Markov Chain Monte Carlo methods. We have calculated forward potentials as usual in the forward-backward algorithm, but backward messages were calculated only for the change points that were drawn, not for the all possible path states. We have seen by the synthetic data tests that our proposed approximate inference algorithm gives acceptable results compared with the exact inference methodology, in addition it costs less CPU time.

In our inference models, we have assumed a fixed number of categories. Our primary next step will be removing this assumption by Dirichlet processes. In fact, many examples are done for SBMs by Dirichlet models for static networks. Our contribution will mainly be on combining time series data with Dirichlet models.

Our current model for change point detection on SBM time series works as if it processes batch data. Another extension can be, modifying the algorithm for online detection of change points.

REFERENCES

1. Sampson, S., *Crisis in A Cloister*, Ph.D. Thesis, Cornell University, 1969.

2. Breiger, R. L., S. A. Boorman and P. Arabie, "An Algorithm For Clustering Relational Data With Applications To Social Network Analysis And Comparison With Multidimensional Scaling", *Journal of Mathematical Psychology*, Vol. 12, pp. 328 – 383, 1975.

3. Grindrod, P., M. C. Parsons, D. J. Higham and E. Estrada, "Communicability Across Evolving Networks", *Physical Review E*, Vol. 83, pp. 46120 – 46124, 2011.

4. Chapanond, A., M. Krishnamoorthy and B "Graph Theoretic and Spectral Analysis of Enron Email Data", *Computational & Mathematical Organization Theory*, Vol. 11, pp. 265–281, 2005.

5. Keila, P. S. and D. B. Skillicorn, "Structure in the Enron Email Dataset", *Computational and Mathematical Organization Theory*, Vol. 11, pp. 183–199, 2005.

6. Airoldi, E. M., D. M. Blei, S. E. Fienberg, E. P. Xing and T. Jaakkola, "Mixed Membership Stochastic Block Models For Relational Data With Application to Protein-Protein Interactions", *In Proceedings of the International Biometrics Society Annual Meeting*, 2006.

7. Dombroski, M. J. and K. M. Carley, "NETEST: Estimating a Terrorist Network's Structure", *Computational & Mathematical Organization Theory*, Vol. 8, pp. 235–241, 2002.

8. Fearnhead, P. and P. Clifford, "On-line Inference For Hidden Markov Models via Particle Filters", *Journal Of The Royal Statistical Society Series B*, Vol. 65, pp. 887–899, 2003.

9. Dias, A. and P. Embrechts, *Change-Point Analysis for Dependence Structures in Finance and Insurance*, Tech. rep., Department of Mathematics, ETH Zurich, 2002.

10. Loschi, R. H., P. L. Iglesias, R. Arellano-Valle and F. R. B. Cruz, "Full Predictivistic Modeling of Stock Market Data: Application to Change Point Problems", *European Journal of Operational Research*, Vol. 180, pp. 282–291, 2007.

11. Liao, W., "Clustering of Time Series Data A Survey", *Pattern Recognition*, Vol. 38, pp. 1857–1874, 2005.

12. Newman, M. E. J., "Mixing Patterns in Networks", *Physical Review E*, Vol. 67, pp. 26126–26139, 2003.

13. Bishop, C. M., *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

14. Milgram, S., "The Small World Problem", *Psychology Today*, Vol. 1, pp. 60–67, 1967.

15. Gavin, A.-C., M. Bosche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, J. M. Rick, A.-M. Michon, C.-M. Cruciat, M. Remor, C. Hofert, M. Schelder, M. Brajenovic, H. Ruffner, A. Merino, K. Klein, M. Hudak, D. Dickson, T. Rudi, V. Gnau, A. Bauch, S. Bastuck, B. Huhse, C. Leutwein, M.-A. Heurtier, R. R. Copley, A. Edelmann, E. Querfurth, V. Rybin, G. Drewes, M. Raida, T. Bouwmeester, P. Bork, B. Seraphin, B. Kuster, G. Neubauer and G. Superti-Furga, "Functional Organization of The Yeast Proteome by Systematic Analysis of Protein Complexes", *Nature*, Vol. 415, pp. 141–147, 2002.

16. Reguly, T., A. Breitkreutz, L. Boucher, B.-J. J. Breitkreutz, G. C. Hon, C. L. Myers, A. Parsons, H. Friesen, R. Oughtred, A. Tong, C. Stark, Y. Ho, D. Botstein, B. Andrews, C. Boone, O. Troyanskya, T. Ideker, K. Dolinski, N. N. Batada and M. Tyers, "Comprehensive Curation and Analysis of Global Interaction Networks in Saccharomyces Cerevisiae", *Journal of biology*, Vol. 5, pp. 11+, 2006.

17. Yu, H., P. Braun, M. A. Yıldırım, I. Lemmens, K. Venkatesan, J. Sahalie, T. Hirozane-Kishikawa, F. Gebreab, N. Li, N. Simonis, T. Hao, J.-F. Rual, A. Dricot, A. Vazquez, R. R. Murray, C. Simon, L. Tardivo, S. Tam, N. Svrzikapa, C. Fan, A.-S. de Smet, A. Motyl, M. E. Hudson, J. Park, X. Xin, M. E. Cusick, T. Moore, C. Boone, M. Snyder, F. P. Roth, A.-L. Barabási, J. Tavernier, D. E. Hill and M. Vidal, "High-Quality Binary Protein Interaction Map of the Yeast Interactome Network", *Science*, Vol. 322, pp. 104–110, 2008.

18. Friedman, N., "Inferring Cellular Networks Using Probabilistic Graphical Models", *Science*, Vol. 303, pp. 799–805, 2004.

19. Walczak, A. M., A. Mugler and C. H. Wiggins, "A Stochastic Spectral Analysis of Transcriptional Regulatory Cascades", *PNAS, Proceedings of the National Academy of Sciences*, Vol. 106, p. 6529, 2009.

20. Gilbert, E. N., "Random Graphs", *The Annals of Mathematical Statistics*, Vol. 30, pp. 1141–1144, 1959.

21. Erdös, P. and A. Rényi, "On Random Graphs, I", *Publicationes Mathematicae (Debrecen)*, Vol. 6, pp. 290–297, 1959.

22. Goldenberg, A., A. X. Zheng, S. E. Fienberg and E. M. Airoldi, "A Survey of Statistical Network Models", *Foundation and Trends in Machine Learning*, Vol. 2, pp. 129–233, 2010.

23. Holland, P. W. and S. Leinhardt, "A Dynamic Model for Social Networks", *The Journal of Mathematical Sociology*, Vol. 5, pp. 5–20, 1977.

24. Holland, P. W. and S. Leinhardt, "An Exponential Family of Probability Distributions for Directed Graphs", *Journal of the American Statistical Association*, Vol. 76, pp. 33–50, 1981.

25. van Duijn, M. A. J., T. A. B. Snijders and B. J. H. Zijlstra, "P2: A Random Effects

Model With Covariates For Directed Graphs", *Statistica Neerlandica*, Vol. 58, pp. 234–254, 2004.

26. Chung, F., L. Lu and V. Vu, "Spectra of Random Graphs with Given Expected Degrees", *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 100, pp. 6313–6318, 2003.

27. Newman, M. E. J., D. J. Watts and S. H. Strogatz, "Random Graph Models of Social Networks", *Proceedings of the National Academy of Sciences*, Vol. 99, pp. 2566–2572, 2007.

28. Blitzstein, J. and P. Diaconis, "A Sequential Importance Sampling Algorithm for Generating Random Graphs with Prescribed Degrees", *Internet Mathematics*, Vol. 6, pp. 489–522, 2011.

29. Harary, F., R. Z., Norman and D. Cartwright, *Structural Models: An Introduction to the Theory of Directed Graphs*, John Wiley & Sons, New York, NY, USA, 1965.

30. Snijders, T. A. and K. Nowicki, "Estimation and Prediction for Stochastic Blockmodels for Graphs with Latent Block Structure", *Journal of Classification*, Vol. 14, pp. 75–100, 1997.

31. Nowicki, K. and T. A. B. Snijders, "Estimation and Prediction for Stochastic Blockstructures", *Journal of the American Statistical Association*, Vol. 96, pp. 1077–1087, 2001.

32. Kemp, C., T. L. Griffiths and J. B. Tenenbaum, "Discovering Latent Classes in Relational Data", *MIT Artificial Intelligence Memos*, p. 258701, 2004.

33. Hofman, J. M. and C. H. Wiggins, "Bayesian Approach to Network Modularity", *Physical Review Letters*, Vol. 100, p. 258701, 2008.

34. Hartigan, J. A., "Direct Clustering of a Data Matrix", *Journal of the American Statistical Association*, Vol. 67, pp. 123–129, 1972.

35. Shan, H. and A. Banerjee, "Bayesian Co-clustering", *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pp. 530–539, IEEE Computer Society, Washington, DC, USA, 2008.

36. Karrer, B. and M. E. J. Newman, "Stochastic Blockmodels and Community Structure in Networks", *Physical Review E*, Vol. 83, pp. 016107–016117, 2011.

37. Mackey, L., D. Weiss and M. Jordan, "Mixed Membership Matrix Factorization", *Proceedings of the 27 th International Conference on Machine Learning*, Haifa, Israel, 2010.

38. Olding, B. P. and J. W. Patrick, "Inference for Graphs and Networks: Extending Classical Tools to Modern Data", p. 6, 2009.

39. Adams, R. P. and D. J. C. MacKay, *Bayesian Online Changepoint Detection*, Tech. rep., University of Cambridge, Cambridge, UK, 2007.