MODEL SELECTION FOR RELATIONAL DATA FACTORIZATION MODELS

by

Çağlar Hızlı

B.S., Electrical and Electronics Engineering, Boğaziçi University, 2010

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 Computer Engineering
Boğaziçi University

2019

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to Prof. Taylan Cemgil for his guidance, enthusiastic encouragement and useful critiques of this research work. I feel privileged by having the chance to work under his supervision. Also, I would like to thank Dr. Serap Kırbız for her guidance and encouragement as a co-supervisor. Besides, I wish to thank Ass. Prof. Tınaz Ekim and Dr. Emre Uğur for their support and guidance during my graduate studies.

A productive research process is only possible with a fruitful research environment. I am grateful for my friends in PILAB Sigma for providing such a great environment. I would like to thank Gökhan Çapan, Burak Kurutmaz, Mine Öğretir, Özge Bozal, Sıla Güler, Merve Ünlü, Hakan Kalaycı, Melih Barsbey, Serhan Daniş, Burak Suyunu, İlker Gündoğdu and Caner Türkmen for their friendship, great academic discussions, and joyful coffee breaks.

I would also like to extend my thanks to my family. I am grateful for their love, guidance and support which provided me this opportunity.

Finally, I want to thank TÜBİTAK for the support. It was a pleasure to work in their research project.

ABSTRACT

MODEL SELECTION FOR RELATIONAL DATA FACTORIZATION MODELS

For relational data factorization, generative models provide a statistically principled approach that allows for extending the factorization task in the probabilistic framework of Bayesian statistics. The most well-known example of such models is the stochastic blockmodel which is a mixture of Bernoullis defined for relational data. In this work, we propose the model BAM-MMSB which replicates the generative process of the mixed-membership stochastic block model (MMSB) within the generic allocation framework of Bayesian allocation model (BAM). In contrast to traditional blockmodels, BAM-MMSB considers the observations as Poisson counts generated by a base Poisson process and marked according to the generative process of MMSB.

A considerable amount of algorithms have been proposed to factorize relational data. However, model selection for this task is still an open problem. In the sequel, we estimate the optimal number of communities for BAM-MMSB by computing the variational approximations of the marginal likelihood for each model order. Although we only perform the model order selection task in our work, we believe that the generic allocation perspective of BAM promises a generalized model selection solution where we not only select the model order but also choose the best factorization.

We describe the proposed model and derive the inference algorithms. Next, we display the experimental setup where we represent relational data as Poisson counts of the allocation model. Later, we assess our variational inference algorithm in terms of interpretability of the model output and block recovery and model selection performance by the experiments on synthetic and real-world datasets.

ÖZET

İLİŞKİSEL VERİ AYRIŞTIRILMASINDA MODEL SEÇİMİ

İlişkisel veri ayrıştırılmasında üretken modeller ilkeli bir yaklaşım sunar, ve ayrıştırma görevini Bayesçi istatistiğin olasılıksal çerçevesi içinde genişletmeye izin verir. Bu modellerin en bilinen örneği, ilişkisel veri üzerinde tanımlı bir Bernoulli karışım modeli olan raslantısal öbek modelidir. Bu çalışmada, karışık üyelikli raslantısal öbek modelinin (MMSB) üretken sürecini Bayesçi atama modelinin (BAM) genel atama çerçevesinde yeniden oluşturan BAM-MMSB modelini öneriyoruz. Geleneksel öbek modellerin aksine, BAM-MMSB gözlemleri temel bir Poisson süreci tarafından üretilen ve MMSB'nin üreteç modeline göre işaretlenmiş Poisson sayımları olarak kabul eder.

İlişkisel verileri ayrıştırmak için kayda değer miktarda algoritma önerilmiştir. Ancak, bu modeller için model seçimi hala açık bir problemdir. Çalışmanın devamında, her model boyutu için marjinal olabilirliğin varyasyonel yaklaşımlarını hesaplayarak, BAM-MMSB modelinde eniyi topluluk sayısını tahmin ediyoruz. Çalışmamızda sadece model boyutu seçimi görevini yerine getirmemize rağmen, BAM'ın genel atama perspektifinin yalnızca model boyutunu değil aynı zamanda en iyi ayrıştırma modelini seçtiğimiz genelleştirilmiş bir model seçimi çözümü vaat ettiğine inanıyoruz.

Çalışmada önce önerilen modeli açıklıyoruz ve çıkarım algoritmalarını türetiyoruz. Daha sonra, ilişkisel verileri atama modelinde Poisson sayıları olarak temsil ettiğimiz deneysel kurumu anlatıyoruz. Son olarak, model çıktısının yorumlanabilirliği ve model seçimi performansı açısından algoritmamızı, sentetik ve gerçek dünya veri kümeleri üzerinde yapılan deneylerle değerlendiriyoruz.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$\mathbf{1}\{.\}$	Indicator function
$\mathcal{B}(.)$	Beta distribution
$\mathcal{BE}(.)$	Bernoulli distribution
B	Block matrix where each element B_{kl} represents interaction probability from block k to block l
C_i	Connectivity pattern of node i
$c_{i\tau}$	Indicator for token τ to select source i
$\mathcal{D}(.)$	Dirichlet distribution
$\mathcal{D}(.\|\|.)$	Divergence metric
$d_{j\tau}$	Indicator for token τ to select destination j
$\mathcal{D}_{KL}(.\|\|.)$	Kullback-Leibler divergence
E	Edge set
e_B	Estimation error of block matrix B in Frobenius form
e_{kl}	The number of edges between block k and l
e_π	Estimation error of block memberships π in Frobenius form
$\mathbb{E}_{p(.)}[.]$	Expectation with respect to the probability density function $p(.)$
G	Graph
$\mathcal{G}$	A directed graphical model $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ represents random variables and $\mathcal{E}$ represents the conditional independence relationships
$\mathcal{GA}(.)$	Gamma distribution
H	Excitation matrix in matrix factorization model
i_n	The index of n^{th} random variable in the allocation model
I_n	The number of states that random variable indexed by i_n has
$i_{pa(n)}$	The index of the parents of n^{th} random variable in the allocation model
$J(.)$	Objective function to optimize
K_{opt}	Optimal number of blocks

$L(.)$	Auxialiary function representing ELBO
$l(.)$	Marginal log likelihood
$l_c(.)$	Complete log likelihood
$\mathcal{M}(.)$	Multinomial distribution
$\mathcal{N}(.)$	Gaussian distribution
N_+^-	Total negative tokens in weighted pseudocount heuristics
N_+^+	Total positive tokens in weighted pseudocount heuristics
$p_\theta(.)$	Probability density function parametrized by θ
$\mathcal{PO}(.)$	Poisson distribution
$\mathcal{PP}(.)$	Poisson process
$q_\phi(.)$	Variational density function parametrized by ϕ
S	Allocation tensor
s^τ	An increment in allocation tensor S at time τ
V	Vertex set
W	Template matrix in matrix factorization
X	Observation matrix/tensor
X^o	Observed indices of the observation matrix/tensor
X^m	Missing indices of the observation matrix/tensor
Y	Adjacency matrix
Z	Latent variable in topic models representing topic or block memberships
$z_{k\tau}^{\rightarrow}$	Indicator for token τ to select source block k
$z_{l\tau}^{\leftarrow}$	Indicator for token τ to select destination block l
$\alpha(i_{1:N})$	Dirichlet measures for Dirichlet variable indexed by $i_{1:N}$
β_k	Word/Dictionary distribution for topic k in topic models
λ	The rate of Poisson variable
μ	Mean
σ	Variance
π	Mixing proportions
τ	Time index

$\theta(i_{1:N})$	Each element of the probability tensor representing assignment probabilities for each index $i_{1:N}$

LIST OF ACRONYMS/ABBREVIATIONS

ARI	Adjusted Rand Index
BAM	Bayesian Allocation Model
BAM-MMSB	Proposed model where MMSB is replicated in BAM
BNMF	Bayesian Non-negative Matrix Factorization
ELBO	Variational Lower Bound
EM	Expectation Maximization
GaP	Gamma-Poisson Factorization Model
LDA	Latent Dirichlet Allocation
LSI	Latent Semantic Indexing
MAP	Maximum A-Posteriori Estimation
MCMC	Markov Chain Monte Carlo
MF	Matrix Factorization
MLE	Maximum Likelihood Estimation
MM	Mixture Models
MMSB	Mixed Membership Stochastic Blockmodel
mPCA	Multinomial Principal Component Analysis
NMF	Non-negative Matrix Factorization
PCA	Principal Component Analysis
PLSI	Probabilistic Latent Semantic Indexing
PMF	Probabilistic Matrix Factorization
RI	Rand Index
SBM	Stochastic Blockmodel
VI	Variational Inference
wALS	Weighted Alternating Least Squares

1. INTRODUCTION

Complex systems display a collective behavior which arises from the combinations of interactions among individual components [3]. Commonly, they appear in diverse fields of scientific research such as social, biological, and data sciences. For example, human interactions in sociometry, protein-protein interactions in biology, and computer interactions in information technology show complex interaction structures. Figure 1.1 illustrates a physical complex system which consists of interacting computer devices on the internet.

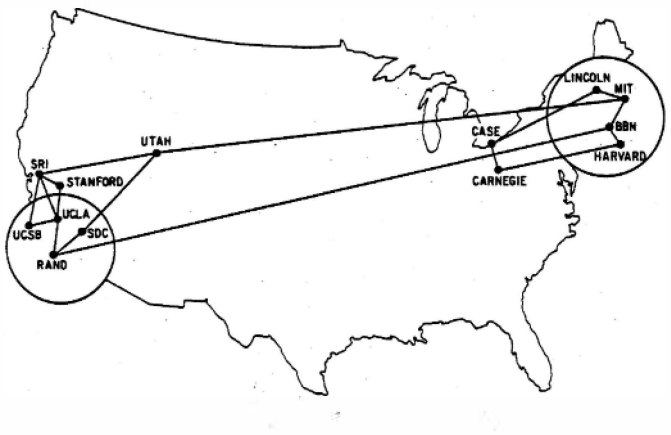

Figure 1.1. Interacting computer devices of Arpanet, the ancestor of the internet, in December 1970 (image taken from [1]).

Large relational data sets have emerged in the past decades as it gets easier and cheaper to measure the interactions on a complex system. These data sets can be represented conveniently as networks or graphs. Graphs are mathematical objects encoding interacting objects as nodes connected by edges. More formally, a graph $G = (V, E)$ consists of a pair of sets where V is the set of vertices and $E \subset V \times V$ is the set of edges that connects the pairs of vertices, $E \subseteq \{\{u, v\} : u, v \in V\}$.

Now, consider the example in Figure 1.1. We observe the connections between computer devices such as UCS-SRI, SRI-UTA, and so on. This relational data set can be abstracted as an undirected graph $G = (V, E)$ as shown in Figure 1.2 (Left). Here,

the node set V corresponds to the devices, and the edge set E corresponds to the set of communication links. Moreover, graphs are commonly represented as adjacency matrices, as shown in Figure 1.2 (Right). The elements Y_{ij} of the binary adjacency matrix $Y \in \{0,1\}^{|V|\times|V|}$ is equal to 1 if an edge exists between nodes i and j, and 0 otherwise.

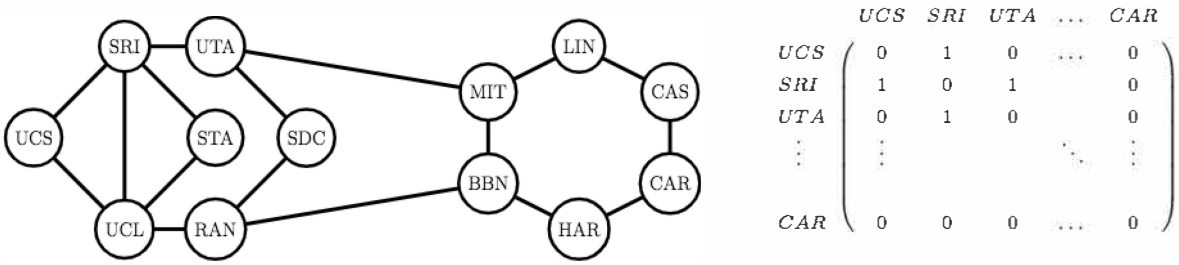

Figure 1.2. The complex system in Figure 1.1 represented as an undirected graph (Left). Adjacency matrix representation for the same graph (Right).

The network in Figure 1.2 displays a heterogeneous connectivity structure where some nodes show similar structural behavior in terms of their connectivity and some not. With this in mind, we can follow a simple approach and color the nodes that seem to have more edges among themselves. This way, we divide the network into two subgroups, as in Figure 1.3. Notice that this simple approach results in two communities, such that each one contains geographically closer nodes (Figure 1.1).

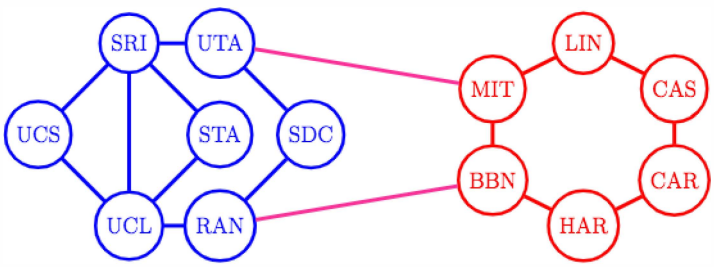

Figure 1.3. Possible Arpanet communities.

As the relational data sets have grown enormously, one challenge is to understand and interpret the characteristics of large, complex networks similar to the toy example in Figure 1.3. To this end, network analysis aims to discover latent structures in large

relational data sets so that it is possible to reason about the underlying systems [4]. In this regard, a fundamental tool for discovering them is to decompose a complex network into its building blocks called *communities* [5].

In contrast to the massive growth in data set sizes, real-world networks are mostly sparse, i.e., their adjacency matrices contain many zeros called negative samples. Although it is common to accept the missing data as negative samples, the reason behind these samples is ambiguous. It can result from (i) lack of interaction or (ii) lack of information such as limited opportunities or measurements. One way to deal with the ambiguity is to apply heuristics such as weighted alternating least squares or down-sampling negative examples as proposed by Pan *et al.* [6]. However, a statistically rigorous way would be to integrate the observation process into the generative model if the missing data are not missing at random [7–10].

Despite the sparsity, most real-world networks are well-connected thanks to the heterogeneity in their connectivity patterns. More specifically, many large networks seem to display *community structure* [11, 12] where some nodes are more densely connected compared to others. These structures are assumed to provide insight into the topology and evolution of the networks. As a result, they have been widely used for various applications including link prediction [13], functional classification [14], epidemic spreading [15], information diffusion [16] and topic modeling [17].

A considerable amount of methods has been proposed to discover community structures. Most methods optimize the cost function of a given metric such as modularity [18]. However, these suffer from being only heuristically motivated [17]. On the other hand, a statistically principled approach is to formulate probabilistic generative models that are responsible for the network evolution. Additionally, probabilistic generative models are known to provide rigorous methods for the model selection problem based on statistical evidence [19].

Compared to the amount of work on detecting communities, there is little work on the model selection problem. Formally, the model selection problem corresponds to

selecting the optimal number of communities for the detection algorithm for a given model. For this problem, the generative models provide principled likelihood-based approaches exploiting Bayesian model selection procedures. Recent work on novel approaches depends on exact or approximate computations of the marginal likelihood by using variational approximations [20], BIC-based approximations [21], and non-parametric methods [19].

For relational data, one of the most popular generative models is the stochastic blockmodel (SBM) [22]. It is a random graph model that defines a mixture of Bernoullis over relational data. Its generative process assigns each node i to a block z_i and accordingly, the edges are drawn independently conditioned on their block memberships: for each node pair $\{i, j\}$, the probability of an edge $\{i, j\}$ is equal to the element $_{z_i, z_j}$ where denotes the $K \times K$ block matrix containing connection probabilities. Here, K denotes the number of blocks.

The generative process of SBM produces non-overlapping communities with homogeneous Poisson degree distributions within the blocks. However, neither assumptions hold for real-world networks. For this reason, some extensions such as overlapping [23], mixed membership (MMSB) [24] and degree corrected SBMs [25] are proposed to address both issues. Among these, MMSB is a mixed-membership model similar to *latent Dirichlet allocation* (LDA) [26] but defined for relational data. The generative process of MMSB associates each node with multiple blocks through a membership vector $\theta_i \in R^K$, which allows for non-overlapping communities.

Many distinct generative models are proposed for relational data in different contexts [26–30] even though they share much in common. Afterward, the authors of [30–33] describe their relevance to each other and their correspondences with matrix factorization models. In this regard, *Bayesian allocation model* (BAM) [33] proposes a dynamical model that is able to replicate other discrete generative processes within a generic allocation framework. Particularly, BAM allocates the observations to latent variables which respect a given factorization implied by a domain-specific directed graphical model $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

In this thesis, we propose to model mixed-membership stochastic blockmodels of relational data as an instance of Bayesian allocation model. This choice is motivated by the fact that BAM provides a generic allocation framework for discrete observations. We consider that the generic modeling framework promises a generalized model selection solution where we not only select the model order but also can choose the most appropriate model for a given empirical network. Furthermore, BAM allows for a principled Bayesian model selection procedure.

1.1. Scope of the Thesis

The scope of our work can be summarized as follows:

(i) BAM-MMSB: We define BAM-MMSB which replicates the generative process of the MMSB within the generic allocation framework of BAM. First, the proposed model is described in detail. Next, the inference algorithms are derived and implemented.

(ii) Handling missing data: The variational inference algorithm is extended to handle the case where missing data are missing at random. Next, block recovery performance is evaluated under the effect of missing data.

(iii) Model selection performance: We use variational approximations of the marginal likelihood for the model order selection task similar to the work of Latouche *et al.* [20]. Then, we show the model selection performance of the proposed algorithm both under synthetic and benchmark networks.

This thesis is organized as follows: First, Chapter 2 summarizes the modeling elements including matrix factorization models and probabilistic generative models for text and graphs. Next, Chapter 3 describes the proposed model in detail. After that, Chapter 4 represents the inference algorithms. Finally, Chapter 5 displays the experiments and results.

2. MODELING ELEMENTS FOR RELATIONAL DATA

In this chapter, we walk through the modeling elements needed to understand the proposed model. First, matrix factorization and topic models are described while their correspondences which motivate Bayesian allocation model are highlighted. Next, Bayesian allocation model is presented. Lastly, the generative models of graphs are detailed.

2.1. Matrix Factorization Models

Matrix factorization models aims to decompose the observed data into its factors. They have gained popularity in diverse fields of machine learning such as topic modeling [29], community detection [34], recommender systems [35], computer vision [30], audio applications [36], etc, where the observations are available in matrix form. For example, the observations are represented as a data matrix $X \in \mathcal{N}^{I \times J}$ in topic modeling. Here, the row i corresponds to the i^{th} word in the dictionary and the column j represents the j^{th} document in the corpus. The element X_{ij} stands for the number of occurrences of word i in document j.

As the definition suggests, the column vectors x_j are the data samples. Furthermore, these samples can be modeled as a factorization into two factor matrices $W \in R^{I \times K}$ and $H \in R^{K \times J}$ such that X is approximately equal to their product.

$$X \approx WH.$$

In the literature, W and H are commonly called *template* and *excitation* matrices and the model is illustrated in Figure 2.1.

Matrix factorization models for topic modeling such as *latent semantic indexing (LSI)* [2] and *non-negative matrix factorization (NMF)* [27] are initially based on

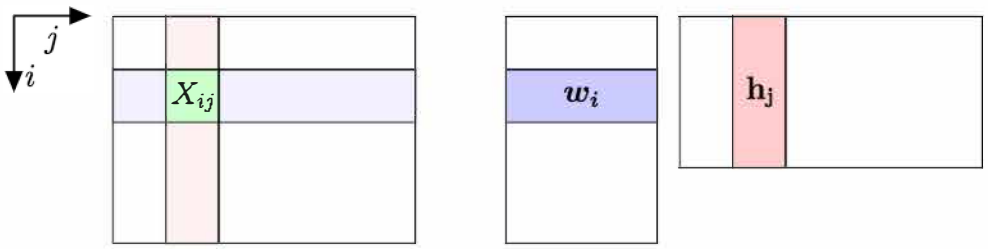

Figure 2.1. Matrix factorization model.

heuristic approaches, ie, they optimize an arbitrarily chosen cost function. Formally, they are defined as a minimization problem:

$$W, H = \arg\min_{W^*, H^*} D(X||WH),$$

where a divergence D serves as the chosen cost function. For example, LSI decomposes word-document count matrix by singular value decomposition that minimizes the Euclidean distance between X and WH. As an alternative to LSI, popular NMF versions optimize the cost functions of Euclidean distance or Kullback-Liebler divergence between X and WH while constraining W and H to be nonnegative. Note that this non-negativity constraint leads to additive and interpretable representations.

Despite their simplicity, heuristic-based approaches are limited in terms of their extendability to other problem formulations such as model selection or active learning. In this respect, a statistically more principled approach is to formulate probabilistic generative models. This way, the problem can be extended within the theoretical framework of Bayesian statistics [21].

2.2. Probabilistic Generative Models

A probabilistic generative model assumes that an underlying random process is responsible for data generation. Further, it makes assumptions on the form of the probability density function defined over the process by defining a function $p_\theta(X)$ such that $X \sim p_\theta(X)$ for a given input X. Here, the inference problem corresponds to the estimation of the parameters θ such that X is most likely to be observed under $p_\theta(X)$.

In addition, generative models make conditional dependence assumptions on the variables encoded by *directed graphical models*. Graphical models are probability networks $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where each node $v \in \mathcal{V}$ represents a random variable, and the edge set $\mathcal{E}$ is used to encode the conditional dependence relationships. For example, consider a simple regression model with input matrix X and output matrix Y. Generally, we assume that knowing X changes our belief on Y. This relationship is simply encoded as: $x \rightarrow y$. Intuitively, the direction of the arrow implies the flow of data generation: $X \sim p_{\theta_x}(X), Y \sim p_{\theta_y}(Y|X)$.

A major tool in probabilistic modeling is introducing *latent variables* Z to the model which are not observed. This allows for richer density models that are multimodal. As an example, consider the following graphical model: $z \rightarrow x$. When $p_{\theta_z}(Z)$ and $p_{\theta_x}(X|Z)$ are chosen in the forms of categorical and Gaussian distributions, respectively, we obtain a mixture of Gaussians model. One interpretation of this mixture is that each data point comes from a different Gaussian component/cluster leading to a multimodal density function.

As can be seen from the Gaussian mixture example, Gaussian distributions are popular likelihood choices for continuous observations. However, the observations are discrete in the case of relational data. For the latter case, one important class of generative models are topic models [37] where Bernoulli, Multinomial or Poisson likelihoods are natural alternatives.

2.3. Topic Models

Topic models are hierarchical generative models aiming to produce low-dimensional representations of document collections called *topics*.

2.3.1. Mixture Models

Mixture models assume that the observed data is a combination of different clusters representing each subpopulation. The ratio of data points from these clusters is specified by the *mixing proportions* $\pi \in \mathcal{R}^K$. First, the generative process assigns each data sample to a cluster $\{Z_j = k\}$ as shown in Figure 2.2. Next, each data sample X_n is sampled from the cluster likelihood $p_{\beta_k}(X_j|Z_j = k)$.

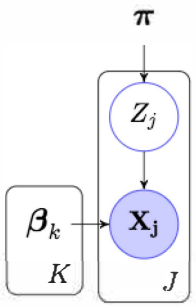

Figure 2.2. GM for Multinomial mix.

In document collections, data samples are the documents while clusters are the topics. Accordingly, each document X_n is assigned to a topic by the generative process. Here, each topic β_k is actually a probability density function over J words, ie, a word distribution. When the document likelihood is chosen as a multinomial distribution, we obtain a mixture of multinomials. The joint distribution is as follows:

$$p_{\beta,\pi}(X, Z) = \prod_j p_\pi(Z_n)\, p_\beta(X_n|Z_n) \tag{2.1}$$

$$= \prod_j \prod_k \pi_k \cdot \mathcal{M}(X_n|\beta_k) \tag{2.2}$$

Mixture models are the simplest forms of topic models in terms of latent variable dimensionality since $|Z_n| = 1$. As a result, they have limited expressive power compared to high-dimensional versions such as mixed-membership models where $|Z_n| > 1$.

2.3.2. Mixed-Membership Models

2.3.2.1. Latent Dirichlet Allocation. As Section 2.3.1 describes, mixture models provide a natural methodology for clustering, but they produce inflexible density models compared to the mixed-membership models. Similar to the last section, mixed-membership models define each topic β_k as a word distribution. On the other hand, they represent each data sample as a mixture of the topics. So, each document j now becomes a probability density function over K topics, ie, a topic distribution θ_j. Moreover, this leads to more realistic text models where each document can consist of words from different topics.

Latent Dirichlet allocation (LDA) [26] is the most well-known example of mixed-membership models defined for topic modeling. It builds upon the earlier work of *probabilistic latent semantic indexing (PLSI)* [38] by treating each parameter set as random variables in a full Bayesian setting. In contrast to the matrix factorization models, LDA focuses on word identities at each word position in a document. In this context, the observations are represented as a word positions matrix $W_{J\times N}$. Here, the row j corresponds to the j^{th} document and the column n represents the n^{th} word position in document j. Therefore, the element W_{jn} stands for the word identity in the document j at the position n.

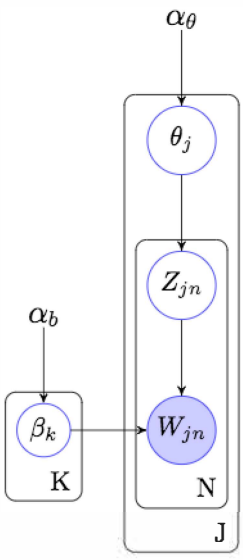

Figure 2.3. GM for LDA.

The generative process is as follows:

(i) For each topic k, draw a word distribution $\beta_k \sim \mathcal{D}(\alpha_\lfloor)$, where $\beta_k, \alpha_b \in \mathcal{R}^J$.

(ii) For each document j, draw a topic distribution $\theta_j \sim \mathcal{D}(\alpha)$, where $\theta_j, \alpha \in \mathcal{R}^K$.

(iii) For each word position $n \in [N]$ in document j:

  (i) Draw a topic $Z_{jn} \sim \mathcal{M}(Z_{jn}; 1, \theta_j)$.

  (ii) From word distribution $\beta_{z_{jn}}$, draw a word identity $W_{jn} \sim \mathcal{M}(W_{jn}; 1, \beta_{z_{jn}})$.

Notice that priors for β and θ are also provided in the generative process for the full Bayesian treatment. Since LDA is defined over the word identities instead of the word counts, each document is now in the form of a series of categorical distributions rather than a multinomial distribution. This results in the following joint distribution:

$$p_\alpha(W, Z, \theta, \beta) = \Big(\prod_k p_{\alpha_b}(\beta_k)\Big)\Big(\prod_j p_{\alpha_a}(\theta_j)\Big)\Big(\prod_j \prod_n p(Z_{jn}|\theta_j)p(W_{jn}|\beta_{Z_{jn}})\Big)$$

2.3.2.2. Toy Example. The expressive power of mixed-membership models is best illustrated through an example. To achieve this, we create a sample data set by revising the example corpus in *latent semantic indexing* paper by Deerwester [2]. The corpus and its corresponding word-document matrix are shown in Table 2.1 and Table 2.2 respectively.

The original corpus consists of 9 documents: (i) 5 documents from topic c and (ii) 4 documents from topic m. Furthermore, an artificial document denoted by c/m is added into the corpus. Note that the words of document c/m are selected such that they come from both topics c and m. This way, we can compare how the models behave when the corpus consists of documents associated with (i) only one topic and (ii) more than one topic.

Encoded	Titles
c1	*Human* machine *interface* for Lab ABC *computer* applications
c2	A *survey* of *user* opinion of *computer system response time*
c3	The *EPS user interface* management *system*
c4	*System* and *human system* engineering testing of *EPS*
c5	Relation of *user* perceived *response time* to error measurement
c/m	*Graph trees system user*
m1	The generation of random, binary, unordered *trees*
m2	The intersection *graph* of paths in *trees*
m3	*Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4	*Graph minors*: A *survey*

Table 2.1. Document corpus from [2] with additional artificial document c/m.

The inferred topic distributions of LDA and mixture of multinomials are compared in Figure 2.4. When all documents are associated with only one topic, the topic distributions are very similar. However, when there exists a document associated with more than a topic, we see that the standard mixture model is not flexible enough for correct representation. The reason is that the standard mixture model constraints each document to have an isolated topic assignment. As a side effect, the resulting topics are constrained to contain words from other themes as well. Consequently, the model produces coupled topic representations which are hard to interpret.

On the contrary, LDA is able to represent c/m as a mixture of both topics correctly. When a document is represented as a mixture, it can consist of several words from its topic mixture. As a result, the model can produce well-separated topic representations. Even though this example is on topic modeling, the same analogy holds for network structures.

Terms	Documents									
	c1	c2	c3	c4	c5	c/m	m1	m2	m3	m4
human	1	0	0	0	0	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0	0
user	0	1	1	0	1	1	0	0	0	0
system	0	1	1	2	0	1	0	0	0	0
response	0	1	0	0	1	0	0	0	0	0
time	0	1	0	0	1	0	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	1	0
graph	0	0	0	0	0	1	0	1	1	1
minors	0	0	0	0	0	0	0	0	1	1

Table 2.2. The corresponding word-document matrix of the corpus in Table 2.1.

2.3.2.3. Other Mixed-Membership Models. While LDA focuses on the word positions matrix W as input, there are other mixed-membership models working on the word-document matrix $X \in \mathcal{N}^{I \times J}$. Buntine [28] suggests using a multinomial distribution as the document likelihood in *multinomial PCA (mPCA)* model. Similar to LDA, each document is represented as a topic distribution $\theta_j \in \mathcal{R}^K$ where K corresponds to the number of topics. First, each topic's total word count c_j is drawn from a multinomial distribution. Next, these total counts are allocated to word counts according to each topic's word distribution $\beta_k \in \mathcal{R}^I$. The generative process is as follows:

$$\theta_j \sim \mathcal{D}(\alpha), \text{ for } j = 1, ..., J$$

$$c_j \sim \mathcal{M}(L_j, \theta_j), \text{ for } j = 1, ..., J \text{ where } c_j \in \mathcal{R}^K, \text{ and } L_j = \sum_i X_{ij},$$

$$X_{:j} \sim \prod_k \mathcal{M}(c_{jk}, \beta_k), \text{ for } j = 1, ..., J.$$

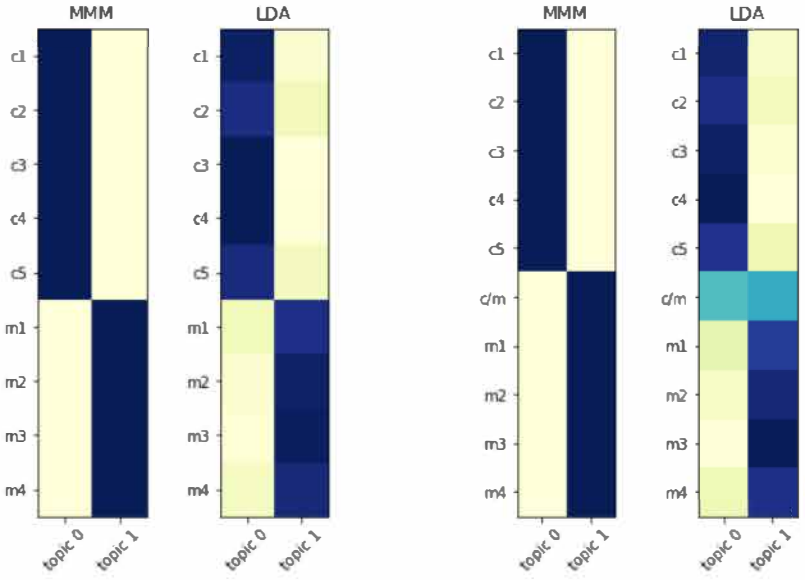

Figure 2.4. Comparison of topic distributions of LDA and mixture of multinomials for (Left) the original corpus [2], and (Right) the revised corpus.

Or an equivalent process (up to a constant) can be written more compactly as in [28]:

$$\theta_j \sim \mathcal{D}(\alpha), \text{ for } j = 1, ..., J \tag{2.3}$$

$$X_{:j} \sim \mathcal{M}(L_j, \beta^T \theta_j), \text{ for } j = 1, ..., J. \tag{2.4}$$

In addition to the multinomial likelihoods, each document can be represented as a series of Poisson random variables representing each word count. As an early example, Canny [29] proposes a Gamma-Poisson generative model, GaP, which approximates the word-document matrix X by the product of two factor matrices θ and β. The variable interpretations are the same as definitions of LDA and mPCA:

$$\theta_j \sim \mathcal{GA}(a_j, b_j), \text{ for } j = 1, ..., J, \tag{2.5}$$

$$X_{:j} \sim \mathcal{PO}(\beta^T \theta_j), \text{ for } j = 1, ..., J. \tag{2.6}$$

In this formulation, GaP also has a factorization interpretation showing that probabilistic topic models are highly related to non-negative matrix factorization.

2.3.2.4. Correspondences between Various Topic Models and Matrix Factorization. Paisley *et al.* [32] indicates that we can derive the matrix factorization formulation by integrating out the latent variables in a topic model. This relationship is also clear from the equations (2.4) and (2.6) since integrating out the latent variables results in a similar formulation in the case of discrete latent variables: $\lambda_{ij} \propto \sum_k \beta_{ik}^T \theta_{kj}$. The product $\beta^T \theta_j$ is equal to the weighted sum of the columns of β^T which correspond to the word distributions. This creates a weighted average of word distributions where weights reflect each topic k's contribution to document d. In this regard, word distributions β and topic distributions θ display a similar functionality to the template and excitations matrices respectively.

A more detailed analysis of this connection has been carried off by Buntine and Jakulin in [31]. In their work, they have stated a set of correspondences among different topic models and non-negative matrix factorization:

- LDA is the full Bayesian version of PLSI.
- The equivalence of LDA and mPCA is rather obvious since both models utilize Dirichlet-Multinomial conjugacy. While LDA chooses to draw topics/words from a series of categorical distributions, mPCA uses multinomial distributions making use of the bag-of-words assumption.
- LDA is closely related to GaP. This relationship depends on the interplays between (1) Dirichlet and Gamma, and (2) Poisson and Multinomial distributions:
  (i) Dirichlet prior can be recovered from a Gamma distribution by choosing constant scale parameters.
  (ii) Poisson and Multinomial distributions are linked through the following identity: when the sum of n Poisson variables is known, the set of Poisson variables has a multinomial distribution conditioned on their sum [39]:

$$\mathbf{1}\{y_+ = \sum_n y_n\} \cdot \prod_n \mathcal{PO}(y_n; \lambda_n) = PO(y_+; \lambda_+) \cdot M(y; y_+, \lambda/\lambda_+)\}$$

- NMF is closely related to GaP since maximum likelihood inference on a Poisson likelihood model such as $X \sim \mathcal{PO}(WH)$ results in the update equations of NMF with Kullback-Leibler divergence (KL-NMF). Therefore, KL-NMF is equivalent to GaP without Gamma priors on β and θ, ie, its maximum likelihood version.

2.4. Probabilistic Matrix Factorization

As the hierarchical generative models have been widely successful in many applications, they are adopted to matrix factorization models as well. *Probabilistic matrix factorization (PMF)* by Salakhutdinov and Mnih [40] is an early example that has a full Gaussian structure. It consists of two Gaussian factor matrices coupled with a Gaussian observation noise:

$$
\begin{aligned}
U &\sim \mathcal{N}(0, \sigma_u \mathbf{I}), \\
V &\sim \mathcal{N}(0, \sigma_v \mathbf{I}), \\
R &\sim \mathcal{N}(U^T V, \Sigma).
\end{aligned}
$$

A more convenient model for discrete count data is *Bayesian non-negative matrix factorization (BNMF)* by Cemgil [30]. BNMF extends GaP by providing full Bayesian inference with Gamma priors on both template and excitation matrices:

$$
\begin{aligned}
T &\sim \mathcal{GA}(A_t, B_t), \\
V &\sim \mathcal{GA}(A_v, B_v), \\
S_{ikj} &\sim \mathcal{PO}(T_{ik} V_{kj}), \\
X_{ij} &= \sum_k S_{ikj}.
\end{aligned}
$$

The paper further shows that multiplicative update equations for NMF is equivalent to the maximization step of EM algorithm when the priors are ignored.

All these correspondences underline the fact that BNMF generalizes many other discrete models such as NMF, LDA, mPCA, and GaP. Eventually, this suggests a research question whether a generic framework for discrete latent variables exists. *Bayesian Allocation Model (BAM)* [33] is a step towards this direction and analyzed in the next section.

2.5. Bayesian Allocation Model

BAM builds up a generic generative model framework for discrete count data. It is basically composed of two processes: *generation* and *allocation*:

(i) *Generation*: Initially, it defines a base Poisson process which is expected to generate T number of tokens equal to total observations at timestamps $0 < t_1, t_2, ..., t_T < 1$.

(ii) *Allocation*: At each timestamp, each token is marked as a member of a specific Poisson process indexed by $i_{1:N}$ where each index i_n represents a discrete random variable with I_n many states. Then, the index collection $i_{1:N}$ represents the set of all possible indices for $\left(\prod_n I_n\right)$ possible values of state combinations.

Allocation process produces $\left(\prod_n I_n\right)$ different Poisson processes which can be viewed as the indices of an allocation tensor, S. Hence, it is insightful to think of each process $S(i_{1:N})$ as a box, each generated token at timestamp τ as balls and allocation tensor S as the collection of boxes filled with balls. The allocation process during the lifespan of S can be summarized as follows:

(i) S is empty at $t = 0$.

(ii) Base process generates T balls with the timestamps $0 < t_1, t_2, ..., t_T < 1$.

(iii) Each balls is marked/colored to an index of the allocation tensor $S(i_{1:N})$ with probability $\theta(i_{1:N})$ independently.

(iv) Here, each joint probability $\theta(i_{1:N})$ can be factorized into *conditional probability tables (CPT)* implied by the given Bayesian network $\mathcal{G}$ of the domain-specific model.

(v) At $t = T$, the total of T balls are marked and allocated to the allocation tensor S.

Joint distribution of the assignments become a high-dimensional array for discrete models where each index $i_{1:N}$ corresponds to the likeliness of a specific configuration. As stated above, the probability tensor $\theta \in \mathcal{R}^N$ obeys a given factorization implied by a Bayesian network $\mathcal{G}$, representing conditional dependence assumptions of the domain-specific model. In box analogy, each entry $\theta_{i_{1:N}}$ tells us how likely it is for a ball to be marked with color $i_{1:N}$ and placed into the box $i_{1:N}$. From the factorization implied by $\mathcal{G}$, the probability that a ball is marked with color $i_{1:N}$ is:

$$\theta(i_{1:N}) = \prod_n \theta_{n|pa(n)}(i_n, i_{pa(n)})$$

This relationship is best illustrated with an example. For instance, some models in Section 2.3.2.4 such as LDA or NMF imply a Bayesian network $\mathcal{G}$ with conditional assumptions of the form: $i \to k \to j$. If we consider that the probability tensor θ obeys the factorization implied by $\mathcal{G}$, the probability that a ball is marked with color $i_{1:N}$ becomes:

$$\theta_{ijk}(i, j, k) = \theta_i(i)\theta_{k|i}(k, i)\theta_{j|k}(j, k).$$

Specific contractions of the probability tensor, θ correspond to marginals of different variables. Therefore, they can be used to obtain conditional distributions implied by the model. We can either construct the probability tensor from these views or construct views from the probability tensor itself. However, usually, we are not given *conditional probability tables* as input. As a result, one inference problem is the posterior probabilities of the possible probability tensor configurations given the observed data.

The hyperparameter for the probability tensor θ is a parameter tensor α that contains Dirichlet measures with entries $\alpha(i_{1:N})$. Furthermore, Cemgil *et al.* [33] suggests that it is important to keep the measures of each Dirichlet random variable consistent. To impose structural constraints consistently for implied factorizations, the following contractions are needed:

$$\alpha_{n|pa(n)}(i_n, i_{pa(n)}) = \sum_{i_{\neg fa(n)}} \alpha(i_{1:N})$$

where each $\alpha_{n|pa(n)}(i_n, i_{pa(n)})$ represent Dirichlet measures for the corresponding Dirichlet random variable $\theta_{n|pa(n)}(i_n, i_{pa(n)})$:

$$\theta_{n|pa(n)}(:, i_{pa(n)}) \sim \mathcal{D}(\alpha_{n|pa(n)}(:, i_{pa(n)}))$$

Then, we can summarize the generative process of BAM as follows;

$$\lambda \sim \mathcal{GA}(a, b) \tag{2.7}$$

$$\theta_{n|pa(n)}(:, i_{pa(n)}) \sim \mathcal{D}(\ \alpha_{n|pa(n)}(:, i_{pa(n)})\ ), \forall n \in [N], \forall i_{pa(n)} \tag{2.8}$$

$$S(i_{1:N}) \sim \mathcal{PO}(\lambda \prod_{n=1}^{N} \theta_{n|pa(n)}(:, i_{pa(n)})) \tag{2.9}$$

$$X(i_V) = \sum_{i_{\bar{V}}} S(i_{1:N}) \tag{2.10}$$

2.6. Generative Models of Graphs

2.6.1. Stochastic Blockmodel

Stochastic blockmodels (SBM) are mixture models defined for relational data. They are based on finding building blocks of a network which consists of *similar* nodes

in terms of their connectivity patterns. The building blocks are also referred as blocks or communities. For a graph $G = (V, E)$ with N nodes, K blocks and the adjacency matrix $Y \in \{0,1\}^{N \times N}$, the connectivity pattern C_i of node i can be formalized as follows [4]:

$$C_i \equiv \{Y(i, j \in k) : \forall k \in [K]\},$$

where $j \in k$ iterates over each node in block k. Here, the connectivity pattern C_i represents how node i connects to the nodes belonging to each $k \in [K]$ given the nodes and their corresponding blocks. Furthermore, the similarity measure is stated formally as *stochastic equivalence* by Holland *et al.* in [22]. Holland *et al.* states that two nodes i and r are stochastically equivalent if exchanging them does not change the probability of interactions concerning them, ie, they connect to the same set of nodes with similar probabilistic measures:

$$C_i \approx C_r.$$

As an example, consider a social network of N students at a university. The network consists of students in different years such as freshmen, sophomores, juniors and seniors. Assume that the students from each year are highly connected with the students from the same year, but they only know a few others from different years.

Let f and g denote classification and permutation functions respectively for each student such that $f : [N] \to \{1, 2, 3, 4\}$ and $g : [N] \to [N]$ where $[N] \equiv 1, ..., N$. The stochastic equivalence indicates the probability events associated with the adjacency matrix are invariant to exchanging the nodes as long as $f(g(i)) = f(i), \forall i \in V$.

A set of stochastically equivalent nodes form a cluster or block. A block corresponds to a specific social group in social networks, a specific scientific discipline in citation networks and a specific functionality in protein-protein interaction networks [12]. For this reason, the blocks are considered as useful statistical tools providing insight

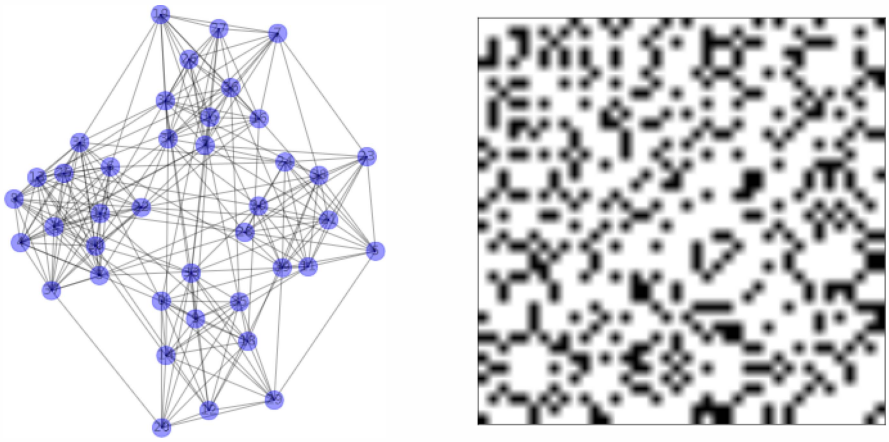

Figure 2.5. Generated data for the school network.

about network topology and evolution. From a generative perspective, they let us divide a large and complex network into manageable pieces such that a generative model can be defined with only a small set of parameters.

A probabilistic generative model can bu built to find out these building blocks. As Peixoto explains in [5], a possible statistics for this purpose is the set of edges e_{kl} between blocks k and l where $k, l \in [K]$.

$$e_{kl} = \sum_{ij} Y_{ij} \cdot \mathbb{I}\{z_i = k\}\mathbb{I}\{z_j = l\}, \text{ for } k, l = 1, ..., K,$$

$$\langle e_{kl} \rangle = \sum_{y} p(y) \cdot e_{kl},$$

where z_i is the latent variable corresponding to the block membership of node i. Now, suppose that $\langle e_{kl} \rangle$ are equal to known constants c_{kl} for $k, l = 1, ..., K$. According to the maximum entropy principle, the generative distribution should be the one which maximizes entropy subject to the constraints of information on the empirical moments c_{kl} for $k, l = 1, ..., K$ [41].

A common way to maximize an objective subject to the constraints is to use Lagrange multipliers:

$$\begin{aligned} J(p(y),\lambda) &= -\sum_y p(y)\log p(y) - \sum_{kl}\lambda_{kl}(c_{kl} - \langle e_{kl}\rangle) - \lambda_y\big(\sum_y p(y) - 1\big), \\ &= -\sum_y p(y)\log p(y) - \sum_{kl}\lambda_{kl}(c_{kl} - \sum_y p(y)\cdot e_{kl}) - \lambda_y\big(\sum_y p(y) - 1\big), \end{aligned}$$

where the first term is the entropy $H(y) = -\sum_y p(y)\log p(y)$ and we add the last constraint $\lambda_y\big(\sum_y p(y) - 1\big)$ so that $p(y)$ is a proper probability distribution. The function $J(p(y),\lambda)$ is called the Lagrange function.

Next, the Lagrange function $J(p(y),\lambda)$ is differentiated with respect to the probability distribution $p(y)$. Notice that calculus of variations is used for this operation since it involves the differentiation of a function with respect to another function.

$$\begin{aligned} \frac{dJ(p(y),\lambda)}{dp(y)} &= -1 - \log p(y) - \sum_{kl}\lambda_{kl}e_{kl} - \lambda_y \\ &= 0 \end{aligned}$$

This equality leads to the following distribution:

$$p(y) = \exp(-1 - \lambda_y - \sum_{kl}\lambda_{kl}e_{kl}) \tag{2.11}$$

$$= \frac{1}{Z}\exp(-\sum_{kl}\lambda_{kl}\sum_{ij}Y_{ij}\cdot\mathbb{I}\{z_i = k\}\mathbb{I}\{z_j = l\}) \tag{2.12}$$

$$= \frac{1}{Z}\exp(-\sum_{ij}\sum_{kl}\mathbb{I}\{z_i = k\}\mathbb{I}\{z_j = l\}Y_{ij}\lambda_{kl}) \tag{2.13}$$

We can complete the equation (2.13) to a Bernoulli distribution by choosing the lagrangian λ_{kl} as follows:

$$\begin{aligned} -\lambda_{kl} &= \log\big(\frac{B_{kl}}{1 - B_{kl}}\big) \\ B_{kl} &= \frac{\exp(-\lambda_{kl})}{1 + \exp(-\lambda_{kl})}. \end{aligned}$$

The resulting maximum entropy distribution is shown below:

$$p(y) = \frac{1}{Z}\exp(-\sum_{ij}\sum_{kl}\mathbb{I}\{z_i = k\}\mathbb{I}\{z_j = l\}Y_{ij}\log\left(\frac{B_{kl}}{1-B_{kl}}\right) \tag{2.14}$$

$$= \prod_{ij}\prod_{kl}\left(B_{kl}^{Y_{ij}}(1-B_{kl})^{1-Y_{ij}}\right)^{\mathbb{I}\{z_i=k\}\mathbb{I}\{z_j=l\}} \tag{2.15}$$

$$= \prod_{ij} B_{z_i z_j}^{Y_{ij}}(1-B_{z_i z_j})^{1-Y_{ij}} \tag{2.16}$$

Notice that this distribution is a mixture of Bernoullis where the number of mixtures is equal to $K \times K$. Another way to visualize these mixtures is to think of them on a $K \times K$ grid. The element with index pair (k, l) corresponds to the mixture responsible for the directed interactions between the blocks $k \to l$.

The maximum entropy derivation provides a justifying motivation for SBMs. Also, Bayesian perspective treats each parameter as a random variable. As a result, we add prior distributions for mixing proportions π and mixture component parameters B. Accordingly, the updated generative process is described as follows:

1. For each block pair $(k, l) \in K \times K$:
    i. Draw interaction probability, $B_{kl} \sim \mathcal{B}(a_{kl}, b_{kl})$.
2. Draw block proportions, $\pi \sim \mathcal{D}(\alpha)$, where $\pi, \alpha \in \mathcal{R}^K$
3. For each node $i \in V$:
    i. Draw block membership, $z_i \sim \mathcal{M}(\pi)$.
4. For each node pair $(i, j) \in N \times N$:
    i. Draw interaction, $Y(i, j) \sim \mathcal{BE}(B_{z_i z_j})$.

This process leads to the following joint probability distribution:

$$p(Y, B, Z, \pi) = p(\pi|\alpha)\prod_{kl} p(B_{kl}|a_{kl}, b_{kl})\prod_{i} p(z_i|\pi)\prod_{ij} p(Y_{ij}|B_{z_i z_j}).$$

Similar to topic models, SBMs consist of discrete latent variables. This similarity hints a close relation with the matrix factorization models. Equivalently, this model has one template matrix in the form of interaction probabilities B. However, there need to be two excitation matrices for z_i and z_j since an interaction has two units: source and destination. Such a factorization form is illustrated in Figure 2.6.

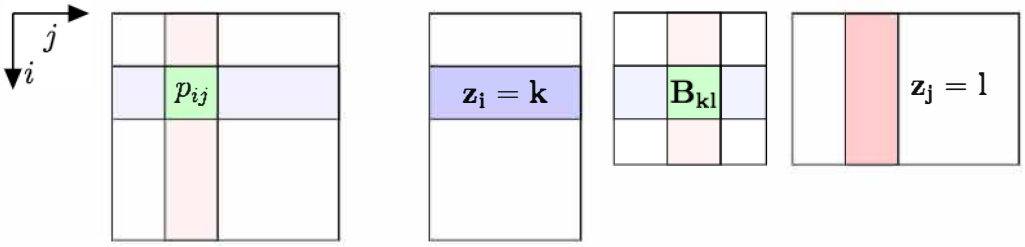

Figure 2.6. SBM in matrix factorization form

In Figure 2.6, two excitation matrices Z_{source} and $Z_{destination}$ are actually selection matrices since each node belongs to one cluster. Accordingly, the row vector z_i and the column vector z_j are one-hot encoded. Therefore, they are unable to generate overlapping communities and hence, they have limited expressive power compared to the mixed-membership versions.

2.6.2. Mixed-Membership Stochastic Blockmodel

In real-world networks, blocks or communities are not mutually exclusive, i.e., their intersections may not always be empty sets. Community borders are rarely thin lines. Instead, they are in the form of overlapping regions. Consider a social network where people are linked via their connections from school, work, neighborhood, etc. There exist some people who are members of both a workplace and a neighborhood community. Despite their popularity, SBMs are unable to model this phenomenon.

The reason behind this drawback is that SBMs follows a hard clustering methodology. They consider each node as a member of one block strictly. Alternatively, from a generative perspective, they commit only one latent variable for the block membership representation. A more flexible representation can be achieved by a mixed-membership version as described in Section 2.3.2.

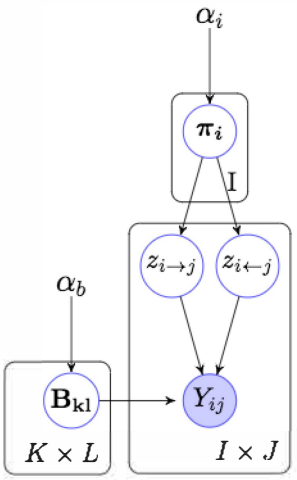

Figure 2.7. Graphical model for MMSB.

Two possible extensions are proposed for overlapping communities: mixed membership stochastic blockmodel (MMSB) [42] and overlapping stochastic blockmodel (OSBM) [23]. OSBM models each membership vector θ_i of node i as a sequence of Bernoulli random variables where $\theta_i \in \{0,1\}^K$. On the other hand, MMSB considers each membership vector θ_i of node i as a Dirichlet distribution where $\theta_i \in \mathcal{R}^{\mathcal{K}}$, ie, a point on $K-1$ simplex. Each point on $K-1$ simplex represents K non-negative weights whose sum is equal to 1. By assigning varying weights on each membership relation, the MMSB offers a more realistic type of soft clustering. And, the generative process is as follows;

1. For each block pair $(k,l) \in K \times K$:
   i. Draw an interaction probability, $B_{kl} \sim \mathcal{B}(a_{kl}, b_{kl})$.
2. For each node $i \in V$:
   i. Draw a mixed membership vector, $\pi_i \sim \mathcal{D}(\alpha_K)$.
3. For each node pair $(i,j) \in N \times N$:
   i. Draw a membership indicator for source, $z_{i\to j} \sim \mathcal{M}(\pi_i)$.
   ii. Draw a membership indicator for destination, $z_{i\leftarrow j} \sim \mathcal{M}(\vec{\pi}_j)$.
   iii. Draw an interaction, $Y(i,j) \sim \mathcal{BE}(z_{i\to j}^T B\ z_{i\leftarrow j})$.

Note that its matrix factorization form is equivalent to the model shown in Figure 2.6. The only difference is that z_i and z_j vectors are no longer one-of-N vectors. Instead,

they are proper probability distributions whose elements' sum is equal to one. Then, the joint probability becomes:

$$p(Y, Z, B, \pi) = \prod_{kl} p(B_{kl}|a_{kl}, b_{kl}) \prod_{i} p(\pi_i|\alpha) \prod_{ij} p(\vec{z}_{i\to j}|\pi_i) p(\vec{z}_{i\leftarrow j}|\pi_j) p(Y_{ij}|\vec{z}_{i\to j}^{T} B \vec{z}_{i\leftarrow j}).$$

Another feature of MMSB is that it is built as a mixed-membership model which have a close relation with the generalized allocation scheme of BAM. From the generalization perspective, although it's possible to infer the latent variables directly from the generative process above, we choose to model MMSB as an instance of BAM. This way, we aim to exploit the flexible framework of BAM.

3. PROPOSED MODEL

3.1. Mixed-Membership Block Model as an instance of Bayesian Allocation Model

MMSB described in Section 2.6.2 is a hierarchical latent model defined on discrete network data that can be realized through BAM detailed in Section 2.5. This choice is motivated by the fact that BAM provides a generic allocation framework for discrete observations. Particularly, BAM allows for allocating discrete observations to latent classes with respect to any given factorization implied by a directed graphical model $\mathcal{G}$. Thanks to its inherent flexibility, we consider that this perspective promises a generalized model selection solution where we not only select the model order but also choose the most appropriate model for a given empirical network.

To be able to see the relation, let us define the following indicators $c_{i\tau}$, $d_{j\tau}$, $z^{\rightarrow}_{k\tau}$, $z^{\leftarrow}_{l\tau}$ and $t_{s\tau}$ to encode events for token $\tau \in [S_+]$:

(i) $c_{i\tau}$: token τ selects source i.
(ii) $d_{j\tau}$: token τ selects destination j.
(iii) $z^{\rightarrow}_{k\tau}$: token τ selects source block k.
(iv) $z^{\leftarrow}_{l\tau}$: token τ selects destination block l.
(v) $t_{s\tau}$: token τ selects interaction s;

Similar to the generative process of MMSB described in Section 2.6.2, we can define a hierarchical Dirichlet-Multinomial model over the indicators. The generative process for the indicators is as follows:

$$\gamma_{:} \sim \mathcal{D}(\eta_\gamma) \qquad \phi_{:} \sim \mathcal{D}(\eta_\phi)$$

$$c_{:\tau} \sim \mathcal{M}(\gamma_{:}, 1) \qquad d_{:\tau} \sim \mathcal{M}(\phi_{:}, 1)$$

$$\pi_{:i} \sim \mathcal{D}(\eta_{\pi_i}) \qquad \pi_{:j} \sim \mathcal{D}(\eta_{\pi_j})$$

$$z^{\rightarrow}_{:\tau} | c_{:\tau} \sim \prod_i \mathcal{M}(\pi_{:i}, 1)^{c_{i\tau}} \qquad z^{\leftarrow}_{:\tau} | d_{:\tau} \sim \prod_j \mathcal{M}(\pi_{:j}, 1)^{d_{j\tau}}$$

$$\beta_{:kl} \sim \mathcal{D}(\eta_\beta)$$

$$t_{:\tau} | z^{\rightarrow}_{:\tau}, z^{\leftarrow}_{:\tau} \sim \prod_k \prod_l \mathcal{M}(\beta_{:kl}, 1)^{z^{\rightarrow}_{k\tau} z^{\leftarrow}_{l\tau}}$$

BAM visualizes this sequential index selection through its graphical model notation. Each generated token selects an index set of the form $\{i, k, s, l, j\}$ while the observed index set is $V = \{i, s, j\}$ and latent index set is $\bar{V} = \{k, l\}$. This notation is arguably lighter than the plate notation which is used traditionally to show graphical models of indexed data. The difference is illustrated in Figure 3.1.

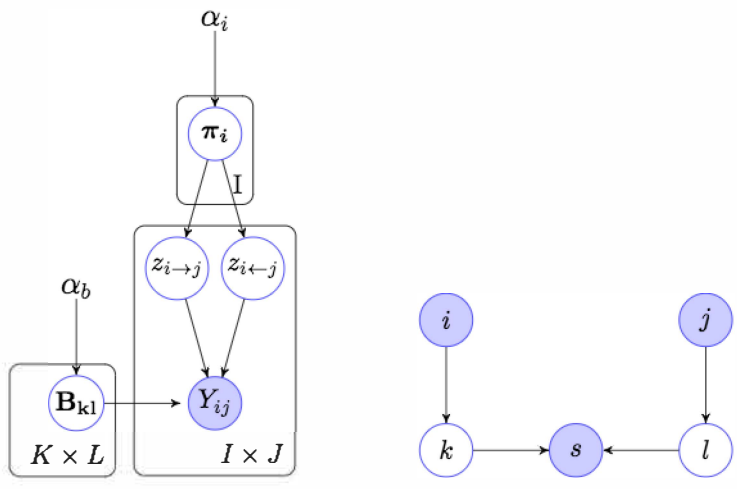

Figure 3.1. Comparison of MMSB graphical models in two different notations: (Left) Traditional graphical model, and (Right) Graphical model in the notation of BAM.

Then, each index of the joint indicator becomes;

$$s^{\tau}_{iksljl} = c_{i\tau} \wedge z^{\rightarrow}_{k\tau} \wedge t_{s\tau} \wedge z^{\leftarrow}_{l\tau} \wedge d_{j\tau} \tag{3.1}$$

This implies that the joint indicator is categorically distributed with $s^{\tau} \sim M(\theta, 1)$ with each cell having the assignment probability;

$$\begin{aligned} \theta_{iksljl} &= \gamma_i \cdot \pi_{ki} \cdot \beta_{skl} \cdot \pi_{lj} \cdot \phi_j && (3.2)\\ &= \theta_i(i) \cdot \theta_{i|k}(k,i) \cdot \theta_{s|k,l}(s,k,l) \cdot \theta_{l|j}(l,j) \cdot \theta_j(j) && (3.3)\\ &= \theta_i \cdot \theta_{k|i} \cdot \theta_{s|k,l} \cdot \theta_{l|j} \cdot \theta_j && (3.4) \end{aligned}$$

From this point, we will abuse the notation and write shortly $\theta_{s|k,l}$ in place of $\theta_{s|k,l}(s,k,l)$ and other corresponding θ variables. The index $s|k,l$ implies the indices ordered as s,k,l.

Notice that the random variable index s is added to the random variable indices k,l,i,j of MMSB because BAM is defined on Poisson counts in contrast to Bernoulli random variables representing relational data in the generative model of MMSB. The added index s allows for an equivalent representation in the form of count data when the sum of each $S_{ik:lj}$ fiber is constrained to 1. This setup is described in detail in Chapter 5.

Continuing with the generative process of BAM, each index of the allocation tensor S is defined as the collection of all tokens occuring at times τ:

$$S_{iksjl} = \sum_{\tau} s^{\tau}_{iksjl}. \tag{3.5}$$

Accordingly, the sum $S_{+} = \sum_{iksljl} S_{iksjl} = \sum_{iksljl} \sum_{\tau} s^{\tau}_{iksjl}$ is the sum of categorical random variables s^{τ}_{iksjl}. Conditioned on the sum, the allocation tensor S is multinomially distributed:

$$S \sim \mathcal{M}(\theta, S_+). \tag{3.6}$$

The relation of Equation (3.6) to the generative process of BAM can be seen through the interplay between a multinomial distribution and N independent Poisson random variables. The joint distribution of N independent Poisson random variables whose sum equals to S_+ can be factorized into the product of (i) a Poisson random distribution over the total sum S_+ and (ii) a Multinomial distribution over N random variables condition on the total sum S_+:

$$1\{S_+ = \sum_{iksl j} S_{iksjl}\} \cdot \prod_{iksjl} \mathcal{PO}(S_{iksjl}; \lambda\theta_{iksjl}) = \mathcal{PO}(S_+; \lambda) \cdot \mathcal{M}(S; S_+, \theta)\} \tag{3.7}$$

The identity in Equation (3.7) allows us to transform Dirichlet-Multinomial model over the selection indicators to the generative process of BAM as follows:

(i) Draw tokens from a base Poisson Process $\mathcal{PP}(\lambda)$ where $\lambda \sim \mathcal{GA}(a, b)$

(ii) Mark each token according to the graphical model $\mathcal{G}$, implied by MMSB as in Figure 3.1:

$$\theta_i \sim \mathcal{D}(\alpha_i), \qquad \theta_j \sim \mathcal{D}(\alpha_j)$$
$$\theta_{k|i} \sim \mathcal{D}(\alpha_{k|i}), \qquad \theta_{l|j} \sim \mathcal{D}(\alpha_{l|j})$$
$$\theta_{s|k,l} \sim \mathcal{D}(\alpha_{s|k,l})$$

(iii) Allocate the marked tokens to the allocation tensor, S:

$$S_{iksjl} \sim \mathcal{PO}(\lambda\theta_i\theta_{i|k}\theta_{s|k,l}\theta_{l|j}\theta_j)$$

(iv) The observations X_{ijs} are equal to specific contractions of the allocation tensor S where we integrate out the latent variables k, l:

$$X_{ijs} = \sum_{k,l} S_{iksjl}$$

We refer to this generative model as BAM-MMSB.

4. INFERENCE

In latent variable models, the main inference problem is to compute the posterior of latent variables given the observed ones. Intuitively, this operation can be viewed as reversing the generative process of the proposed model in order to find out *the most likely configuration of both the hyperparameters and the latent variables that could produce the observed variables* [37]. In this section, we will explore two closely related inference methods: expectation-maximization (EM) and variational inference (VI).

4.1. Expectation-Maximization

Expectation-maximization (EM) is a fundamental inference method for latent variable models proposed by Dempster *et al.* [43]. Let us denote the observed variables by X, the latent variable set by Z and the parameter set by Φ. To start with, assume we would like to perform maximum likelihood estimation on the joint log likelihood of X and Z. If all X and Z would be observed, the inference problem would take the complete log likelihood l_c into account:

$$l_c(\Phi; X, Z) = \log p(X, Z|\Phi) \tag{4.1}$$

However, Z is not observed. Integrating out Z results in the marginal log likelihood $l(\Phi; X)$ (assuming discrete Z):

$$l(\Phi; X) = \log p(X|\Phi) = \log \sum_z p(X, Z|\Phi) \tag{4.2}$$

Normally, maximum likelihood (MLE) or maximum-a-posteriori (MAP) estimations are performed on the marginal likelihood $l(\Phi; X)$. When $p(X|\Phi)$ is factorized properly, differentiating the marginal likelihood $l(\Phi; X)$ with respect to each parameter $\phi \in \Phi$ is an easy process. On the contrary, the logarithm of the sum $\sum_z p(X, Z|\Phi)$

can not be factorized. Besides, the sum/integral $\sum_z p(X, Z|\Phi)$ is mostly intractable.

This motivates us to work on a possible lower bound on $l(\Phi; X)$. When this lower bound is tight enough, the inference process approximates the parameter set Φ within a small error margin. Similar to the derivation trick applied in importance sampling, we add a proposal to the sum $\sum_z p(X, Z|\Phi)$:

$$l(\Phi; X) = \log p(X|\Phi) = \log \sum_z p(X, Z|\Phi) \tag{4.3}$$

$$= \log \sum_z q(Z) \frac{p(X, Z|\Phi)}{q(Z)} \tag{4.4}$$

Since logarithm is a concave function, we can obtain the following lower bound by using Jensen's inequality:

$$l(\Phi; X) \geq \sum_z q(Z) \log \left(\frac{p(X, Z|\Phi)}{q(Z)} \right) \tag{4.5}$$

$$= L(q; \theta, \lambda), \tag{4.6}$$

where $L(q; \Phi)$ is called the auxialiary function.

4.1.1. Expectation-Maximization for BAM-MMSB

For the BAM, the latent variable set Z is equal to $Z = \{S\}$ and the parameter set Φ is equal to $\Phi = \{\theta, \lambda\}$. Inserting these into Equation (4.5) yields the following inequality:

$$l(\theta, \lambda; X) \geq \sum_s q(S) \log \left(\frac{p(X, S|\theta, \lambda)}{q(S)} \right) \tag{4.7}$$

$$= L(q; \theta, \lambda), \tag{4.8}$$

Let us look at the difference to understand this lower bound:

$$l(\Phi; X) - L(q; \Phi) = \log p(X|\theta, \lambda) - \sum_s q(S) \log \left(\frac{p(X, S|\theta, \lambda)}{q(S)}\right) \tag{4.9}$$

$$= \sum_s q(S) \log p(X|\theta, \lambda) - \sum_s q(S) \log \left(\frac{p(X, S|\theta, \lambda)}{q(S)}\right) \tag{4.10}$$

$$= \sum_s q(S) \left[\log p(X|\theta, \lambda) - \log \left(\frac{p(X, S|\theta, \lambda)}{q(S)}\right)\right] \tag{4.11}$$

$$= \sum_s q(S) \log \left(\frac{p(X|\theta, \lambda)\, q(S)}{p(X, S|\theta, \lambda)}\right) \tag{4.12}$$

$$= \sum_s q(S) \log \left(\frac{q(S)}{p(S|X, \theta, \lambda)}\right) \tag{4.13}$$

$$= D_{KL}(q(S) \,||\, p(S|X, \theta, \lambda)) \tag{4.14}$$

where D_{KL} is the Kullback-Leibler divergence. KL divergence is non-negative and equal to zero only when two probability distributions are the same: $q(S) = p(S|X, \theta, \lambda)$.

The equality in Equation (4.14) shows that when $q(S)$ is selected as the posterior distribution $p(S|X)$, the lower bound is tight. Therefore, we can optimize the log likelihood $l(\Phi; X)$ by optimizing the lower bound $L(q; \Phi)$ iteratively: at each iteration (i) first, maximize parameters θ and λ for a fixed $q(S)$ and (ii) next, compute $q(S)$ for fixed θ and λ:

$$\text{E-STEP: } q(S)^{(n)} = p(S|X, \theta^{(n-1)}, \lambda^{(n-1)}) \tag{4.15}$$

$$\text{M-STEP: } \theta^{(n)}, \lambda^{(n)} = \text{argmax}_{\theta,\lambda}\, L(q; \theta, \lambda) \tag{4.16}$$

4.1.1.1. E-Step. For E-Step, we need to find the posterior of S given the observations X and the parameters θ and λ:

$$p(S|X, \theta, \lambda) = \frac{p(X, S|\theta, \lambda)}{p(X|\theta, \lambda)} \tag{4.17}$$

The joint distribution $p(X, S|\theta, \lambda)$ is as follows:

$$p(X, S|\theta, \lambda) = \exp\Big(-\lambda \sum_{iksl j} \theta_i \theta_{k|i} \theta_{s|k,l} \theta_{l|j} \theta_j + \sum_{iksl j} S_{iksl j} \log(\lambda \theta_i \theta_{k|i} \theta_{s|k,l} \theta_{l|j} \theta_j) - \sum_{iksl j} \log \Gamma(S_{iksl j} + 1) + \sum_{ijs} \log \mathbf{1}[X_{ijs} = \sum_{kl} S_{iksl j}] \Big)$$

By integrating out S from the joint distribution $p(X, S|\theta, \lambda)$, we can find the marginal distribution $p(X|\theta, \lambda)$. Here, the key observation is that the sum of N Poisson variables is another Poisson distribution.

$$\begin{aligned} p(X|\theta, \lambda) &= \prod_{ijs} PO\big(X_{ijs}; \lambda \sum_{kl} \theta_i \theta_{k|i} \theta_{s|k,l} \theta_{l|j} \theta_j\big) \\ &= \exp\Big(-\lambda \sum_{iksl j} \theta_i \theta_{k|i} \theta_{s|k,l} \theta_{l|j} \theta_j + \sum_{iksl j} S_{iksl j} \log(\lambda \sum_{kl} \theta_i \theta_{k|i} \theta_{s|k,l} \theta_{l|j} \theta_j) \\ &\quad - \sum_{ijs} \log \Gamma(X_{ijs} + 1) \Big) \end{aligned}$$

Then, the posterior $p(S|X, \theta, \lambda)$ becomes:

$$p(S|X, \theta, \lambda) = \exp\Big(\sum_{iksl j} S_{iksl j} \log\big(\frac{\lambda \theta_i \theta_{k|i} \theta_{s|k,l} \theta_{l|j} \theta_j}{\lambda \sum_{kl} \theta_i \theta_{k|i} \theta_{s|k,l} \theta_{l|j} \theta_j}\big) - \sum_{iksl j} \log \Gamma(S_{iksl j} + 1) + \sum_{ijs} \log \Gamma(X_{ijs} + 1) + \sum_{ijs} \log \mathbf{1}[X_{ijs} = \sum_{kl} S_{iksl j}] \Big),$$

which is exactly in the form of a multinomial distribution [30].

$$p(S_{i:s:j}|X, \theta, \lambda) \sim M(X_{ijs}, p_{i:s:j}),$$
$$\text{where } p_{iksl j} = \frac{\theta_{s|k,l} \theta_{k|i} \theta_{l|j} \theta_j \theta_i}{\sum_{k,l} \theta_{s|k,l} \theta_{k|i} \theta_{l|j} \theta_j \theta_i}$$

The sufficient statistics of the multinomial distribution $p(S|X, \theta, \lambda)$ is:

$$E_q[S_{iksl j}] = X_{ijs} \cdot p_{iksl j}. \tag{4.18}$$

4.1.1.2. M-Step. Since $q(S)$ is fixed while iteratively maximizing θ and λ, optimizing $L(q;\theta,\lambda)$ with respect to θ and λ is equivalent to optimizing $E_q[\log p(X,S|\theta,\lambda)]$ with respect to θ and λ.

$$\mathbb{E}_{q(S)}[\log p(X,S|\theta,\lambda)] = \mathbb{E}_{q(S)}\Bigg[-\lambda \sum_{iksij} \theta_i \theta_{k|i} \theta_{s|k,l} \theta_{l|j} \theta_j + \sum_{iksij} S_{iksij} \log(\lambda \theta_i \theta_{k|i} \theta_{s|k,l} \theta_{l|j} \theta_j) - \sum_{iksij} \log \Gamma(S_{iksij}+1) + \sum_{ijs} \log \mathbb{1}[X_{ijs} = \sum_{k} S_{iksij}] \Bigg]$$

When we separate the terms with λ and θ:

$$\mathbb{E}_{q(S)}[\log p(X,S|\theta,\lambda)] \approx -\lambda + \sum_{ikj} \mathbb{E}_{q(S)}[S_{ikj}] \cdot \log(\lambda \theta_i \theta_{k|i} \theta_{s|k,l} \theta_{l|j} \theta_j)$$

Then, the following update equations are obtained with the help of Lagrange multipliers:

$$\lambda = \mathbb{E}_{q(S)}[S_+], \qquad \theta_i = \frac{\mathbb{E}_{q(S)}[S_{i++++}]}{\mathbb{E}_{q(S)}[S_+]}, \qquad \theta_j = \frac{\mathbb{E}_{q(S)}[S_{++++j}]}{\mathbb{E}_{q(S)}[S_+]}$$

$$\theta_{k|i} = \frac{\mathbb{E}_{q(S)}[S_{ik+++}]}{\mathbb{E}_{q(S)}[S_{i++++}]}, \qquad \theta_{l|j} = \frac{\mathbb{E}_{q(S)}[S_{+++lj}]}{\mathbb{E}_{q(S)}[S_{++++j}]}$$

$$\theta_{s|k,l} = \frac{\mathbb{E}_{q(S)}[S_{+ksl+}]}{\mathbb{E}_{q(S)}[S_{+k+l+}]}$$

4.2. Variational Inference

Variational inference (VI) is a method where the intractable posterior distribution $p(Z|X)$ is approximated by a fully factorized variational distribution $q(Z)$. In contrast to EM, variational inference is applicable in the full Bayesian setting where each parameter is considered as a random variable. In this case, the set of latent variables Z becomes: $Z = \{S, \theta, \lambda\}$. Using the same proposal trick as in Section 4.1 (and importance sampling), we can write the following equality:

$$\log p(X|\Phi) = L(q) + D_{KL}(q(Z) \,||\, p(Z|X,\Phi)), \text{ where,} \tag{4.19}$$

$$L(q) = \int_Z dZ\; q(Z) \log \left(\frac{p(X,Z)}{q(Z)}\right) \tag{4.20}$$

$$D_{KL}(q(Z) \,||\, p(Z|X)) = -\int_Z dZ\; q(Z) \log \left(\frac{p(Z|X)}{q(Z)}\right) \tag{4.21}$$

4.2.1. Variational Inference for BAM-MMSB

Similar to EM case, the variational lower bound (ELBO) $L(q)$ provides a lower bound for the log likelihood since $D_{KL}(q(Z) \,||\, p(Z|X))$ is non-negative. However, the posterior distribution $p(Z|X,\Phi) = p(S,\theta,\lambda|X,\Phi)$ does not have a known form as in EM case. As a result, it is not possible to find out a tight lower bound and our aim is to find a convenient proposal for $q(Z)$. The mean-field approach proposes a variational distribution $q(Z)$ that can be fully decomposed into its factors:

$$q(S,\theta,\lambda) = q(S) \cdot q(\theta) \cdot q(\lambda)$$

Equation (4.19) implies that maximizing the lower bound $L(q)$ with respect to $q(S)$, $q(\theta)$ and $q(\lambda)$ is equivalent to minimizing the KL divergence between fully factorized $q(Z)$ and posterior distribution, $p(Z|Y)$. The idea is to find a local maxima of the lower bound $L(q)$ with respect to each variational factor $q(S)$, $q(\theta)$ and $q(\lambda)$. Since both ELBO and variational factors are functions, typically calculus of variations is used to find the best distributions $\hat{q}(S)$, $\hat{q}(\theta)$ and $\hat{q}(\lambda)$ which constrain the ELBO to be as tight as possible.

Instead, we will follow a KL divergence based derivation similar to the one proposed by Bishop [44]. Let us consider optimizing $q(S)$ first. Then, we will generalize the expressions by symmetry. We can start by separating the terms including $q(S)$ from the ELBO expression in Equation (4.20):

$$
\begin{aligned}
L(q) &= \sum_S \int_{\theta,\lambda} q(S,\theta,\lambda) \log \big(\frac{p(X,S,\theta,\lambda)}{q(S,\theta,\lambda)}\big) \\
&= \sum_S \int_{\theta,\lambda} d\theta\, d\lambda\, q(S,\theta,\lambda) \log \big(\frac{p(X,S,\theta,\lambda)}{q(S,\theta,\lambda)}\big) \\
&= \sum_S q(S) \int_{\theta,\lambda} q(\theta)q(\lambda) \log p(X,S,\theta,\lambda) - \sum_S q(S) \log q(S) + \text{const} \\
&= \sum_S q(S) \log \hat{p}(X,S) - \sum_S q(S) \log q(S) + \text{const},
\end{aligned}
$$

where $\log \hat{p}(X,S) = \int_{\theta,\lambda} q(\theta)q(\lambda) \log p(X,S,\theta,\lambda) = \mathbb{E}_{q(\theta),q(\lambda)}[\log p(X,S,\theta,\lambda)]$. The last line is equal to the KL divergence between $\log \hat{p}(X,S)$ and $q(S)$:

$$
L(q) \approx -KL(q(S) \mid\mid \hat{p}(X,S))
$$

Hence, when $q(\theta)$ and $q(\lambda)$ is fixed, maximizing the lower bound $L(q)$ is equivalent to minimizing the KL divergence between $q(S)$ and $\hat{p}(X,S)$. The minimum is obtained when $q(S) = \hat{p}(X,S)$. Then, the general expression becomes;

$$
\begin{aligned}
\log q(S) &= \mathbb{E}_{q(\theta),q(\lambda)}[\log p(X,S,\theta,\lambda)] + const \\
q(S) &\propto \exp\big(\mathbb{E}_{q(\theta),q(\lambda)}[\log p(X,S,\theta,\lambda)]\big)
\end{aligned}
$$

By symmetry, we can write the expressions for $q(\theta)$ and $q(\lambda)$ and the variational distributions are summarized below:

$$
\begin{aligned}
q(S) &\propto \exp\big(\mathbb{E}_{q(\theta),q(\lambda)}[\log p(X,S,\theta,\lambda)]\big), \\
q(\theta) &\propto \exp\big(\mathbb{E}_{q(S),q(\lambda)}[\log p(X,S,\theta,\lambda)]\big), \\
q(\lambda) &\propto \exp\big(\mathbb{E}_{q(S),q(\theta)}[\log p(X,S,\theta,\lambda)]\big).
\end{aligned}
$$

Following the optimization steps, we obtain the update equations for $q(S)$, $q(\theta)$ and $q(\lambda)$.

$$q(S) \propto \prod_{i,j,s} \mathcal{M}(S_{k,l|i,s,j}; X_{ijs}, p_{k,l|i,\ s}),_j \tag{4.22}$$

$$q(\theta) \propto \mathcal{D}(\mathbb{E}_{q(S)}[S_{n|pa(n)}] + \alpha_{n|pa(n)}), \tag{4.23}$$

$$q(\lambda) \propto \mathcal{GA}(\mathbb{E}_{q(S)}[S_+] + a, b + 1), \tag{4.24}$$

where p and $\mathbb{E}_{q(S)}[S_{n|pa(n)}]$ is defined as follows:

$$p_{k,l|i,s,j} \propto \mathbb{E}_{q(\theta,\lambda)}[\log(\lambda\theta_{s|k,l}\theta_{k|i}\theta_{l|j}\theta_i\theta_j)], \tag{4.25}$$

$$\mathbb{E}_{q(S)}[S_{n|pa(n)}] = \sum_{i'_{\neg fa(n)}} \mathbb{E}_{q(S)}[S(i'_{1:N})]. \tag{4.26}$$

Here, $\neg fa(n)$ denotes the indices excluding index n and its parents with respect to the graphical model in Figure 3.1 (Right). For example, index s has parents k and l which makes $i_{\neg fa(s)} = \{i, j\}$. Then, $\mathbb{E}_{q(S)}[S_{s|pa(s)}]$ becomes:

$$\mathbb{E}_{q(S)}[S_{s|k,l}](s\,|k,l) = \sum_{ij} \mathbb{E}_{q(S)}[S_{iksij}].$$

The detailed derivations can be found in the Appendix.

4.2.1.1. Computing ELBO. Following factorization of the form: $p(X, S, \theta, \lambda) = p(X|S)p(\theta|S)p(\lambda|S)p(S)$, the evidence lower bound $L(q)$ can be written as follows:

$$\begin{aligned} L(q) &= \sum_S \int_{\theta,\lambda} q(S, \theta, \lambda) \log\left(\frac{p(X, S, \theta, \lambda)}{q(S, \theta, \lambda)}\right) \\ &= E_q[p(X|S)] + E_q[p(\theta|S)] + E_q[p(\lambda\,|S)] + E_q[p(S)] \\ &- E_q[q(S)] - E_q[q(\theta)] - E_q[q(\lambda)] \end{aligned}$$

The detailed derivation can be found in Appendix A.

4.2.2. Handling Missing Data

The update equations of variational inference can be adapted to missing data. Similar to the fully observed case, the latent variable set $Z = \{S, \theta, \lambda\}$ is defined such that it contains both missing and observed indices of the data tensor $X \in \mathcal{N}^{I \times J \times S}$. Let us partition the data matrix X into two sets: $X = \{X^o, X^m\}$ where X^o and X^m represents observed and missing indices respectively. Then, the same operation can also be performed on the allocation tensor S: $S = \{S^o, S^m\}$ such that the contractions of S^o and S^m are equal to X^o and X^m respectively. This partition leads to the following variational distribution:

$$q(S, \theta, \lambda) = q(S^o) \cdot q(S^m) \cdot q(\theta) \cdot q(\lambda)$$

In this setup, the update equations for the observed part of the allocation tensor S^o, the probability tensor θ and the rate parameter λ remain unchanged. The key observation for the missing part of the allocation tensor S^m is that when the conditioning variables X_{ijs} are missing, the variational factor $q(S^m)$ is no longer multinomially distributed. For the missing indices $(ijs) \in X^m$, $q(S^m)$ is a Poisson distribution. Following the same steps as the observed version, we obtain the following update equations:

$$q(S^o) \propto \prod_{i,j,s:(ijs)\in X^o} \mathcal{M}(S_{k,l|i,s,j}; X^o_{ijs}, p_{k,l|i,s,j}), \tag{4.27}$$

$$q(S^m) \propto \prod_{iksl j:(ijs)\in X^m} \mathcal{PO}(S_{iksl j}; \tau_{iksl j}), \tag{4.28}$$

$$q(\theta) \propto \mathcal{D}(\mathbb{E}_{q(S)}[S_{n|pa(n)}] + \alpha_{n|pa(n)}), \tag{4.29}$$

$$q(\lambda) \propto \mathcal{GA}(\mathbb{E}_{q(S)}[S_+] + a, b + 1), \tag{4.30}$$

where p and $\mathbb{E}_{q(S)}[S_{n|pa(n)}]$ are already derived in Equations (4.25) and (4.26) respectively, and $\tau_{iksl j}$ is defined as follows:

$$\tau_{iksij} = \mathbb{E}_{q(\theta),q(\lambda)}[\log(\lambda\theta_{s|k,i}\theta_{k|i}\theta_{l|j}\theta_i\theta_j)] \tag{4.31}$$

Notice that the expectations of the allocation tensor S need to be updated for Poisson indices:

$$\mathbb{E}_q[S_{iksij}] = \begin{cases} X_{ijs} \cdot p_{iksij}, & \text{for } (ijs) \in X^{\bullet} \\ \tau_{iksij}, & \text{for } (ijs) \in X^{m} \end{cases}$$

4.2.2.1. Computing ELBO. Following factorization of the form: $p(X, S, \theta, \lambda) = p(X|S)p(\theta|S)p(\lambda|S)p(S)$, the evidence lower bound $L(q)$ can be written as follows:

$$\begin{aligned} L(q) &= \sum_S \int_{\theta,\lambda} q(S,\theta,\lambda) \log\left(\frac{p(X,S,\theta,\lambda)}{q(S,\theta,\lambda)}\right) \\ &= \mathbb{E}_q[p(X|S)] + \mathbb{E}_q[p(\theta|S)] + \mathbb{E}_q[p(\lambda|S)] + \mathbb{E}_q[p(S)] \\ &\quad - \mathbb{E}_q[q(S^{\bullet})] - \mathbb{E}_q[q(S^{m})] - \mathbb{E}_q[q(\theta)] - \mathbb{E}_q[q(\lambda)] \end{aligned}$$

The detailed derivation can be found in Appendix B.

4.3. Model Selection

For a given latent variable model, the model selection problem corresponds to selecting the dimensionality of the latent space. In the case of blockmodels, the dimensionality of the latent space is equal to the number of communities. Moreover, it is a more challenging task than inferring the block structure given the correct number of communities K. According to Murphy [41], the model selection problem can be solved by:

(i) Comparing log-likelihoods of different models on a test set via cross validation.

(ii) Computing Bayes factors of models $m \in \mathcal{M}$ while approximating the marginal likelihood of each model $\log p(D|m)$ by its variational approximation [20].

(iii) Applying annealed importance sampling (AIS) [45] for estimating the marginal likelihood.

(iv) Applying Bayesian nonparametric methods [19].

Although the gold standard is applying AIS, we compare Bayes factors of variational approximations since it is much more simple and efficient to implement, yet it still provides a principled likelihood-based approach. More formally, the goal is to compute the posterior of each model given the observed data:

$$p(m|D) \propto p(D|m)p(m).$$

When there is no prior knowledge about the models, it is convenient to choose a uniform prior for $p(m)$. Then,

$$\begin{aligned} p(m|D) &\propto p(D|m)p(m) \\ &\propto p(D|m) \\ &\geq L(q|D,m), \end{aligned}$$

where $L(q|D,m)$ is the ELBO given the observed data D and the model m corresponding to a specific number of communities K_m. This inequality shows that the evidence lower bound provides a simple, yet principled approach for the model selection problem.

5. EXPERIMENTS AND EVALUATION

In this chapter, we initially describe the experimental setup where we investigate convenient count representations for relational data, initialization strategies, and hyperparameter choices. Next, we perform experiments both on synthetic and real-world benchmark networks to assess our model in terms of (i) interpretability of the model output, (ii) block recovery performance and (iii) the model selection performance.

5.1. Experimental Setup

This section describes the experimental setup in detail where we investigate convenient count representations for relational data, initialization strategies, and hyperparameter choices.

5.1.1. Count Representations for Relational Data

BAM-MMSB is defined on Poisson counts in contrast to Bernoulli trials that are commonly used for representing a binary adjacency matrix. Therefore, we aim to come up with an equivalent count representation for the adjacency matrix Y of a given network. Consider an adjacency matrix where each element Y_{ij} is a Bernoulli trial parametrized by the parameter ϕ. Then, the probability distribution for Y is:

$$p_{\phi}(Y) = \prod_{ij} \mathcal{BE}(\phi_{ij}).$$

The binary variables can also be encoded as two independent Poisson variables whose sum equals to 1. Conditioned on their sum, two Poisson random variables are distributed as a Binomial distribution where the probability of selecting a category is proportional to the normalized Poisson rates. First, we remind the equality of Equation (3.7) in Chapter 3 between a multinomial distribution and N independent

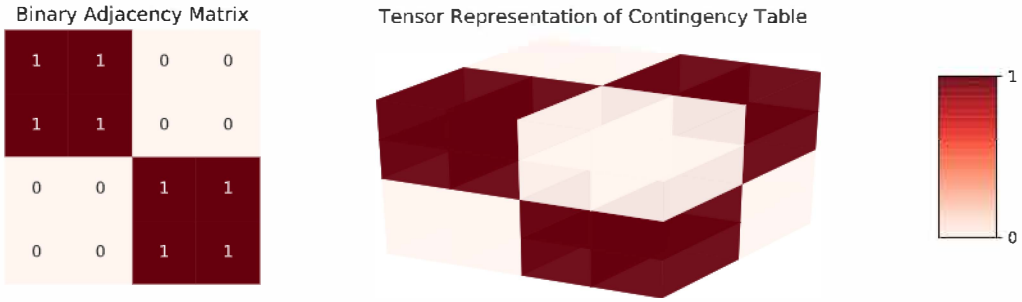

Figure 5.1. Count tensor representation of a binary adjacency matrix.

Poisson random variables:

$$\mathrm{I}\{V_+ = \sum_n V_n\} \cdot \prod_n \mathcal{PO}(V_n; \lambda_n) = \mathcal{PO}(V_+; \lambda_+) \cdot \mathcal{M}(V; V_+, \lambda_n/\lambda_+) \tag{5.1}$$

The same argument can be adapted to the adjacency matrix by considering a count tensor $X \in \mathcal{N}^{I \times J \times S}$ where we extend the adjacency matrix Y by an additional index s.

$$\prod_{ij} \left(\mathbb{I}\{Y_{ij} = \sum_s X_{ijs}\} \cdot \prod_s \mathcal{PO}(X_{ijs}; \lambda_{ijs}) \right) = \prod_{ij} \mathcal{PO}(Y_{ij}; \lambda_{ij+}) \cdot \mathcal{M}(X_{ij:}; Y_{ij}, p_{\lambda_{ij:}})\},$$

where $p_{\lambda_{ij:}} = \left(\frac{\lambda_{ij0}}{\lambda_{ij0}+\lambda_{ij1}}, \frac{\lambda_{ij1}}{\lambda_{ij0}+\lambda_{ij1}}\right)$. The index s represents the possible categories of the observed data. For example, the fibers $(s = 1)$ and $(s = 0)$ denote the positive (1s) and negative (0s) samples of the adjacency matrix respectively. This representation is illustrated in Figure 5.1 and the preprocessing steps are summarized below.

(i) A binary adjacency matrix $Y \in \{0, 1\}^{I \times J}$ is observed.

(ii) We add a dimension and create its corresponding count tensor $X \in \mathcal{N}^{I \times J \times S}$ where $|s| = 2$ for the binary case.

(iii) We place the observations into X with respect to the following rule: $X_{ijs} = \mathbb{I}\{Y_{ij} = s\}$.

5.1.2. Initialization and Hyperparameters of BAM-MMSB

For each parameter configuration in the experiments, the variational inference step is performed for several times (from 35 to 100 initializations). The one which provides maximum ELBO is chosen. However, empirically, we see that the algorithm needs a large number of runs to converge to a local maximum if started with random initializations. For this reason, we use clustering algorithms of sklearn such as kmeans or spectral clustering for initialization purposes.

The hyperparameters of BAM are initialized according to Cemgil *et al.* [33]. Let us denote the total number of tokens by $S_+ = \sum_{i_{1:N}} S(i_{1:N})$. Note that our expectation is that the allocation tensor S is allocated sparsely. Since the hyperparameter $\alpha(i_{1:N})$ represents Dirichlet measure of the index $(i_{1:N})$, it is chosen between $\alpha(i_{1:N}) \in \{0.05, 0.25\}$ to induce sparsity to the allocation tensor S. And, if no prior information is provided, it is reasonable to choose uniform values for $\alpha(i_{1:N}) = \alpha = \frac{a}{\prod_n I_n}$. Furthermore, the parameter λ controls the prior expectation of the total number of tokens. Since Gamma expectation is $\mathbb{E}[\lambda] = \frac{a_\lambda}{b_\lambda}$, the scale hyperparameter b_λ can chosen as a_λ / S_+. Correspondingly, the shape parameter a_λ can be chosen as $a_\lambda = \left((\prod_n I_n) \cdot \alpha\right)$ so that the allocation tensor S is encouraged to be sparse through the parameter α.

5.2. Interpreting the Model Output

With BAM-MMSB, we aim to discover latent block structures in relational data sets so that we can reason about the underlying generative processes. In this regard, we infer the block matrix and the block memberships from the model and investigate how we can reason about these model parameters on synthetic networks. The block matrix B can be computed from the model as follows:

$$\hat{B} = \theta_{s|k,l}(s = 1, :, :),$$

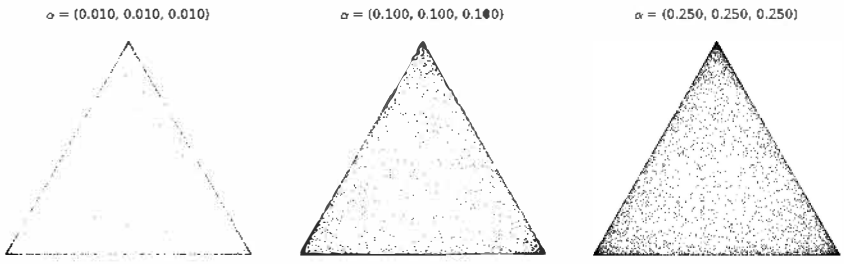

Figure 5.2. Dirichlet samples for $\alpha_\pi = \{0.01, 0.1, 0.25\}$.

while the block membership vectors for source and destination nodes are given as:

$$\hat{Z}_{source} = \theta_{k|i}(:,:)$$
$$\hat{Z}_{destination} = \theta_{l|j}(:,:)$$

Synthetic networks in the experiments are sampled by the generative process of the MMSB. For this process, the block membership vector π_i for each node i is distributed according to the parameter α_π: $\pi_i \sim \mathcal{D}(\alpha_\pi \cdot \mathbf{1_K})$. Therefore, the parameter vector $\alpha_\pi \cdot \mathbf{1_K}$ corresponds to the uniform Dirichlet parameters. For small α_π where $\alpha_\pi < 1$, Dirichlet samples are sparse and as α_π gets closer to 1, the samples get close to uniform. This result is shown in Figure 5.2 where 5000 points are sampled from each parameter configuration.

Figure 5.2 shows that the block membership vector gets more sparse when $\alpha_\pi = 0.01$ and, hence, the overlapping regions between different blocks are smaller. Conversely, when $\alpha_\pi = 0.25$, the generative process produces more uniform membership vectors where the overlapping regions between different blocks get larger. As a result, we expect to see apparent latent block structures at the parameter value $\alpha_\pi = 0.01$, and indistinguishable structures as α_π goes to 0.25.

5.2.1. Synthetic Networks

In this experiment, we explore the latent structures that BAM-MMSB infers. 9 synthetic networks are sampled by the generative model of MMSB. The networks have $|V| = \{90, 120, 150\}$ nodes with $K = \{3, 4, 5\}$ blocks respectively. For each $\{|V| - K\}$ pair, we sample three networks while varying the parameter $\alpha_\pi = \{0.01, 0.1, 0.25\}$ which controls the sparsity of the block memberships.

The block matrix consists of elements equal to $\{\epsilon, 1 - \epsilon\}$ in the generative model where ϵ is set to 0.95. For each $\{|V| - K\}$ pair, n_ϵ elements are set to ϵ, and the rest are set to $1 - \epsilon$ where $n_\epsilon = floor((K \times K)/2)$. Once the parameter n_ϵ is computed, n_ϵ elements are selected from the block matrix at random. The permuted adjacency matrices are displayed for 9 cases in Figure 5.3.

Notice that the adjacency matrices in Figure 5.3 are permuted according to the inferred block memberships. For the permutation, we pretend that each node belongs to a block. Then, assign each one to the block for which the membership probability is maximized. Although the nodes do not have to belong to a unique block in the model, the permuted adjacency matrix is useful for inspection. It reveals the underlying block matrix, and hence, provides a way to understand the network topology.

Figure 5.3 displays the results on a grid where from left to right the parameter α_π is increased and from top to down the complexity of the block structure is increased by the parameters $|V|$ and K. The leftmost column ($\alpha = 0.01$) corresponds to the case where block memberships are sparse, ie, most of the nodes belong to only one block. Conversely, the rightmost column ($\alpha = 0.25$) corresponds to the case where nodes exhibit a uniform membership among different blocks. In the other direction, the top row has the simplest block structure with $K = 3$ blocks and $|V| = 90$ nodes while the bottom row has the most complex structure with $K = 5$ blocks and $|V| = 150$ nodes.

As the block memberships get more uniform (from left to right), it gets harder for the BAM-MMSB model to infer the latent structure. The reason is that the overlapping

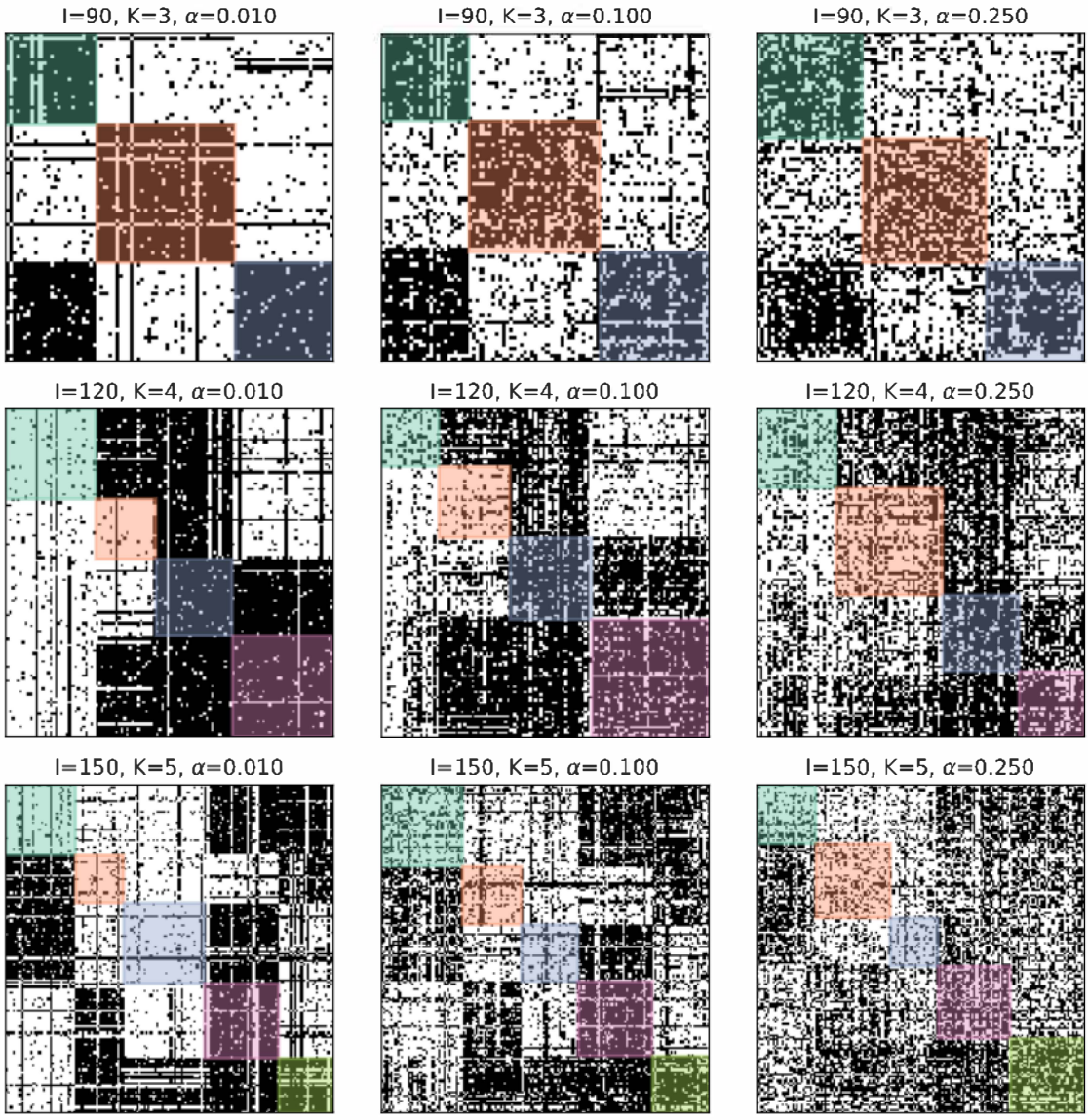

Figure 5.3. The inferred latent block structure while the parameters $\{\alpha_\pi\}$ and $\{|V|, K\}$ and are varied vertically and horizontally respectively.

regions are larger for these cases and, accordingly, the noise increases in this direction. As a result, the latent block structures in the rightmost column are not as apparent as the ones in the leftmost column.

A similar relationship exists in the downwards direction. In this direction, the number of nodes and blocks are increasing at the same rate. Still, the latent block structure is more complex for the bottom row compared to the higher rows, and the increased complexity lowers the quality of the inference. Both effects shape the model behavior of the right bottom corner with the configuration ($|V| = 150, K = 5, \alpha = 0.25$) where the block structure is nearly disappearing.

BAM-MMSB seems to recover latent block structures, and their interpretation is discussed in this section. In the next section, we discover how inferred structures can be evaluated numerically.

5.3. Measuring the Block Recovery Performance

As discussed in Section 5.2, the block recovery task contains two main components: the block matrix and the block memberships. Accordingly, we can analyze how well the blocks are recovered by defining error metrics on these two entities. For these two variables, Tabouy et al [46] measures the estimation error e_B between the true block matrix B and the estimated block matrix $\hat{\theta}_{s|k,l}$ in the Frobenius norm while measuring the adjusted Rand index (ARI) [47] between the true block memberships π and the estimated block memberships $\hat{\theta}_{k|i}$ and $\hat{\theta}_{l|j}$. Even though ARI is a metric defined for hard clustering, it provides a good measure even in the soft clustering case when the block memberships are sparse. Still, we add the estimation error e_π between the true and estimated block memberships as another metric.

Rand index (RI) measures how two different partitions of the same data fit. Consider two partitions $P_1 = \{P_{11}, P_{12}, ...\}$ and $P_2 = \{P_{21}, P_{22}, ...\}$. RI counts the pairs that belong to the same cluster both in P_1 and P_2, eg, the pairs that belong to P_{11} and P_{23} at the same time; and adds the pairs that belong to distinct clusters in P_1 and P_2,

eg, the pairs $i-j$ where if $i \in P_{1q}, j \in P_{1w} : q \neq w$, then $i \in P_{2r}, j \in P_{2t} : r \neq t$. This pair count is divided by the all possible pairs to get a measure for clustering. When two partitions fit completely, RI is 1. However, when the input is two random partitions, RI does not produce a constant value. ARI is the adjusted version of RI such that the ARI value of two random partitions is equal to 0.

The other two metrics are error measures in the Frobenius form:

$$e_B = \frac{||\hat{\theta}_{s|k,l} - B_{true}||_F}{||B_{true}||_F},$$
$$e_\pi = \frac{||\hat{\theta}_{k|i} - \pi_{true}||_F}{||\pi_{true}||_F}.$$

For computing these, all possible block permutations are considered and the one which gives the lowest error is used.

5.3.1. Synthetic Networks

Synthetic networks are sampled by the generative model of MMSB similar to Section 5.3. However, the network topology is fixed as assortative instead of the random generation of the block matrix. This choice supports a stable comparison as the parameter space is explored.

Assortative networks have simple connectivity patterns where nodes from same blocks connect with a probability ϵ which is larger than the probability of interaction between nodes from different blocks. In the experiment, we choose this value to be $\epsilon/10$. The block matrix structure is as follows:

$$B = \begin{pmatrix} \epsilon & \epsilon/10 & \dots & \epsilon/10 \\ \epsilon/10 & \epsilon & & \epsilon/10 \\ \vdots & & \ddots & \vdots \\ \epsilon/10 & \epsilon/10 & \dots & \epsilon \end{pmatrix}$$

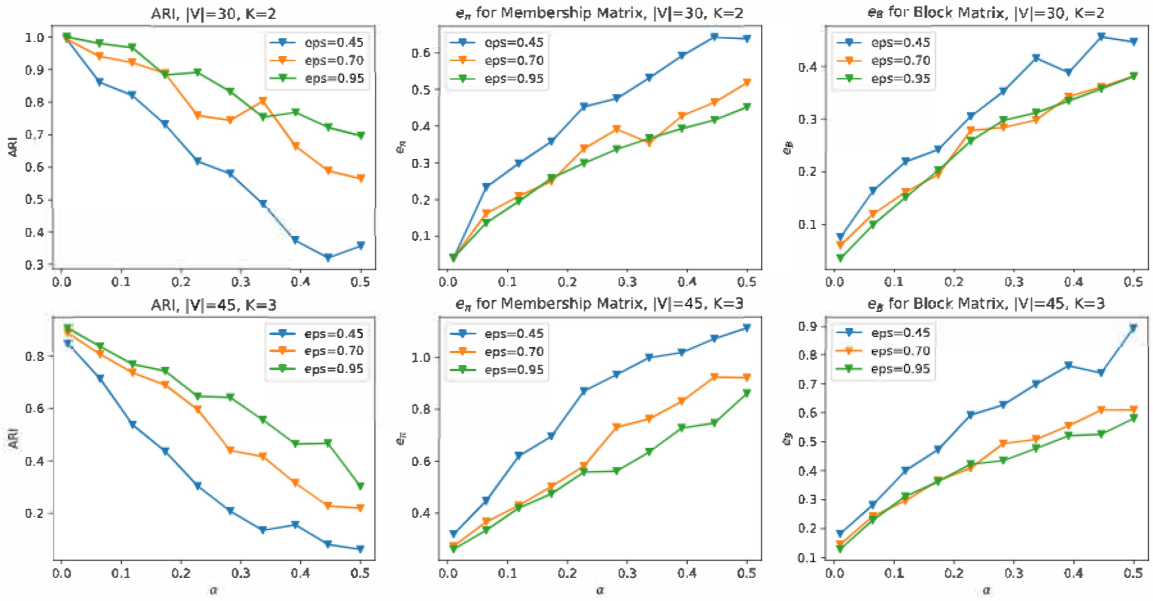

Figure 5.4. Evaluation metrics ARI, e_B and e_π as α is varied for two networks.

For affiliation topology, two different networks with $|V| = \{30, 45\}$ nodes are considered. The number of blocks is assumed to be known: $\{K = L = 2, K = L = 3\}$ for two distinct sizes respectively. This way, we aim to see how the size and the structure of the networks affect the simulation metrics. Furthermore, the parameter ϵ controls the connectivity of the graph. We perform the experiment for three different connectivity levels: $\epsilon = \{0.45, 0.7, 0.95\}$. For each parameter configuration, we sample 20 different assortative networks while changing the parameter $\alpha_\pi \in \{0.01, ..., 0.5\}$. The evaluation metrics ARI, e_B and e_π are averaged over 20 network samples. The results are displayed in Figure 5.5.

The results support the visual interpretations of Section 5.2. BAM-MMSB model recovers the latent block structure completely for the parameter configuration $|V| = 30, K = 2$ even in the graphs with small connectivity, ie, with parameter ϵ set to 0.45. As expected, ARI drops while e_B and e_π grow as the hyperparameter α_π is increased at all connectivity levels. However, as the hyperparameter α_π is rising, the evaluation metrics have a worse decay for the sparsest case where $\epsilon = 0.45$. This suggests the block recovery performance does not degrade gracefully on sparse graphs as the overlapping

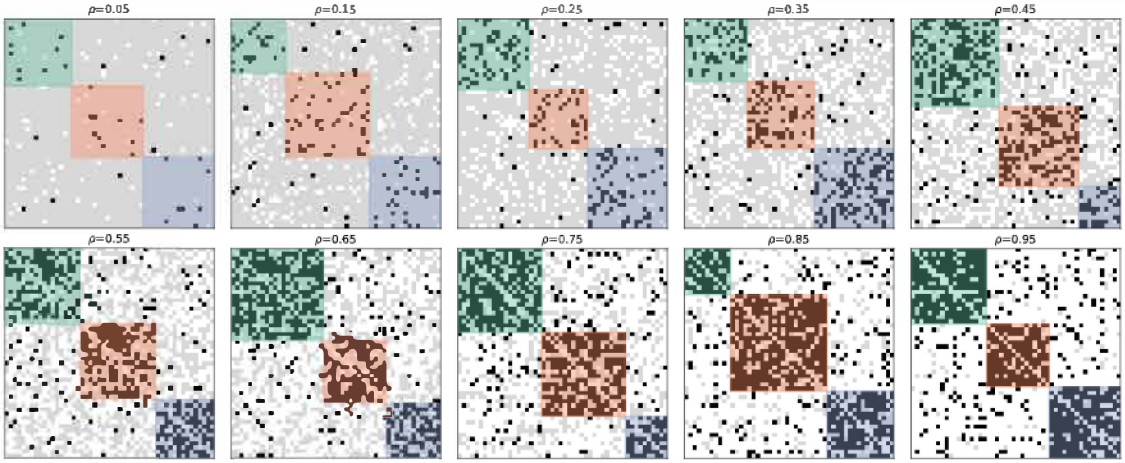

Figure 5.5. Illustration of the missing data ratios in the adjacency matrices.

regions between blocks increase.

Another point is that the block recovery performance is consistently lower for the parameter configuration $\{|V| = 45, K = 3\}$ compared to the the parameter configuration $\{|V| = 30, K = 2\}$. This numerical result also supports the visual interpretation shown in Figure 5.3 where it gets harder for the model to discover the latent structure in case of more complex block structures.

5.3.2. Effect of Missing Data on Block Recovery

In this section, we analyze the effect of missing data on block recovery. Again, the network topology is chosen as assortative. Our intuition is that the blocks would become indistinguishable as the rate of missing data increases. Consider a setting where we randomly sample observations from an adjacency matrix. Let us denote the probability of observing an edge by the parameter ρ.

A network with $|V| = 30$ nodes is considered. The number of blocks is assumed to be known: $\{K = L = 2\}$. The parameter ρ takes values of $\rho \in \{0.05, 0.15, ..., 0.95\}$. For each value of ρ in the missing data process, we sample 10 MMSB networks. The evaluation metrics ARI and e_B are averaged over 10 data samples. The results are displayed in Figure 5.6.

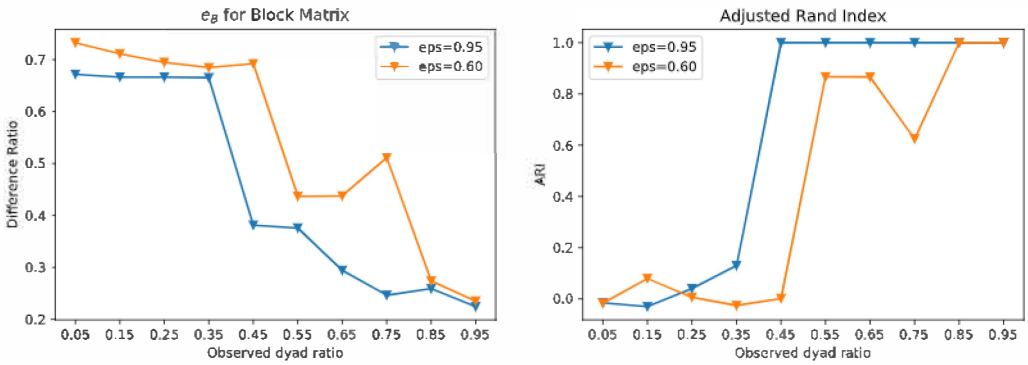

Figure 5.6. Evaluation metrics ARI and e_π as ρ is varied for an assortative network.

As expected, the evaluation metrics ARI and e_B produce very low values when the parameter ρ is low, and the blocks are indistinguishable. Interestingly, the model recover blocks with a low error until a threshold is passed and fails very rapidly afterwards.

5.4. Measuring the Model Selection Performance

For a given latent variable model, the model selection problem corresponds to choosing the optimal number of blocks K_{opt} that explains the latent structure in the observed data best. Bayesian statistics provide principled likelihood-based approaches for this task. The aim is to choose the model which produces the largest marginal likelihood of the observed data X. However, the marginal of X is often intractable. Therefore, we choose to approximate the marginal by its mean-field variational approximation similar to the work of Latouche et al. [20].

First, we perform experiments on synthetic networks. Next, the model selection performance is evaluated for real-world benchmark networks.

5.4.1. Synthetic Networks

To assess model selection performance, we use the assortative networks as in Section 5.3. Similar to Section 5.3, the probability of an edge inside a block is varied to have three different values: $\epsilon = \{0.9, 0.7, 0.5\}$. The parameter ϵ controls the connec-

tivity of the graph. Conversely, the probability of an edge between two distinct blocks p_{out} is set to 0.01.

The synthetic networks have two different node partition schemes: (i) balanced and (ii) unbalanced blocks as in the work of Latouche et al. [48]. Let us denote the set of blocks as $\{k_1, ..., k_K\}$. Balanced blocks have equal number of nodes with $|k_t| \approx |V|/K, \forall t \in [K]$. This effect is achieved in the MMSB generative model by drawing the membership vectors $\pi_i \in \mathcal{R}^K$ for each node i from uniform and sparse Dirichlet distributions as $\pi_i \sim \mathcal{D}(0.01 \cdot \mathbf{1_K})$. For unbalanced blocks, we use geometric sizes where each block's size is proportional to $|k_t| \propto 0.7^t, \forall t \in [K]$ similar to the setup in [48]. As an example, for $K = 5$, the corresponding α_π values are: $\alpha_\pi = \{0.36, 0.25, 0.17, 0.12, 0.08\}$.

In the experiment, each sampled network has $|V| = \{40\}$ nodes. The number of blocks is varied as $K_{true} \in \{2, 3, 4\}$, but in the inference process K is assumed to be unknown. For each $\{K_{true}$, connectivity ϵ, block sizing method$\}$ configuration, we sample 50 different assortative networks and estimate the optimal number of clusters K_{est}. The results are displayed in Table 5.1 and Table 5.2.

True K	Estimated K				
	1	2	3	4	5
2	0	50	0	0	0
3	0	2	43	5	0
4	0	7	33	10	0
2	0	48	2	0	0
3	0	14	36	0	0
4	0	26	21	3	0
2	0	50	0	0	0
3	0	41	9	0	0
4	10	37	3	0	0

Table 5.1. As the connectivity parameter $\epsilon = \{0.9, 0.7, 0.5\}$ is varied vertically, K_{est} estimations in balanced block sizing.

True K	Estimated K				
	1	2	3	4	5
2	0	49	1	0	0
3	0	8	38	4	0
4	0	11	24	13	2
2	0	49	1	0	0
3	0	20	27	3	0
4	0	23	24	2	1
2	0	46	4	0	0
3	0	33	17	0	0
4	0	40	8	2	0

Table 5.2. As the connectivity parameter $\epsilon = \{0.9, 0.7, 0.5\}$ is varied vertically, K_{est} estimations in unbalanced block sizing.

As expected, networks with unbalanced block sizes provide worse block recovery performance for $\epsilon = \{0.9, 0.7\}$ since the block structure is slightly more complex. Interestingly, networks with unbalanced block sizes provide equal or better performance for the cases where K is large. This may result from the fact that the model tends to find one block first and continues to divide the rest of the network if there are enough data samples. This behavior may favor unbalanced network sizes for the harder inference problems.

In this basic setting, our model estimates block memberships and true K when there are enough positive samples with large enough blocks. For $|V| = 40$ nodes that are highly connected ($\alpha = 0.9$), we can recover latent structure of networks for $K_{true} = \{2, 3\}$. Notice that as K increases, the inference problem gets harder since networks get sparser and blocks get smaller. In such cases, our model can not infer the latent structure from one Bernoulli experiment per interaction index.

One observation for large values of K is that the model tends to find one exact block and combines the rest into a big cluster. This pattern suggests that if we continued to observe Bernoulli trials for each index or if we had more observed data, we may

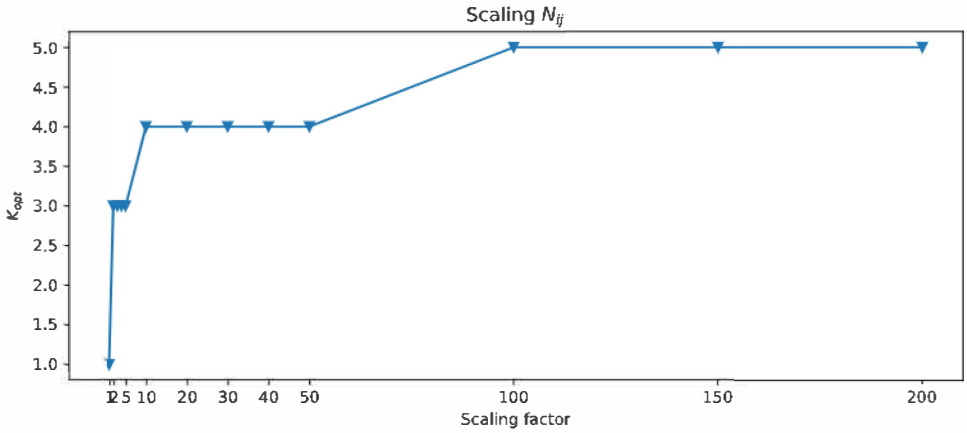

Figure 5.7. Estimated number of blocks K_{opt} as the scaling factor N_{ij} for each index is increased.

capture all of the true clusters. If that is not the case, then, we may apply heuristics such as scaling Bernoulli trials for each index. BAM-MMSB generative model allows us to choose the number of Bernoulli trials N_{ij} per each index of the adjacency matrix.

Notice that $N_{ij} = 1$ corresponds to the count tensor shown in Figure 5.1 as if there has been only one coin toss to represent an interaction. When $N_{ij} = n$ such that $n > 1$, each observation model for a pair (i, j) becomes a binomial experiment with n trials. This procedure brings up the effect of added precision to the node classification. Therefore, we perform an experiment where the contingency tensor is scaled up with increasing $N_{ij} = n$ where $K_{true} = 4$.

As $N_{ij} = n$ increases, the model's confidence in the observations increases, and hence, the model continues to divide existing blocks and create new ones. One interesting point is how the model order estimation K_{opt} behaves with increased pseudocounts $N_{ij} = n$. This is illustrated in Figure 5.7. K_{opt} rises quickly until it is level with the true number of blocks $K_{true} = 4$. Then, it stays at the value of $K_{est} = 4$ for some values of the scaling factor before continuing to increase gradually. For large n values, the model seems to overfit and select overly complex models due to the scaled noise factors. Therefore, it seems reasonable to employ this heuristic approach for a certain data regime.

Although it is not a statistically principled method, we see that scaling pseudo-counts heuristics work well in practice for synthetic networks. However, the noise ratio is relatively large for the real-world networks compared to the synthetic ones since the real-world networks are inherently sparse. For this reason, it is not directly obvious how to leverage this scaling idea. As a result, we borrow a scaling heuristics idea from collaborative filtering in the next section.

5.4.2. Real-World Networks

In this section, simulations are performed on three real-world benchmark networks to assess the model comparison performance. These networks are chosen as (i) Zachary's karate club network [49], (ii) Lusseau et al.'s dolphin social network [50] and (iii) adjacency network of adjectives and nouns in the book David Copperfield by Charles Dickens [51].

Most real-world networks exhibit sparsity. Our benchmark networks are also sparse having 34, 62, and 112 nodes with only 156, 318, and 850 edges, respectively. As a result, our algorithm tends to select model orders with insufficient complexity. Under the circumstances, scaling Poisson counts is a heuristics option. However, Figure 5.7 shows that scaling the contingency tensor directly may have a negative effect on the model order selection when there is inherent noise in the observations. Scaling the noise drives the model to select overly complex representations which are highly sensitive to small fluctuations. We suppose that this is due to the inherent missing data in networks such that a negative sample $Y_{ij} = 0$ can result from the lack of interaction or lack of information.

For the missing data problem in collaborative filtering, Pan et al. [6] proposed weighted alternating least squares (wALS) for sparse binary data sets which contain ambiguity in the interpretation of the negative samples. The idea is that each positive sample has a constant confidence level, which is higher than ambiguous negative samples. This relationship is expressed mathematically by weighting the cost of each index according to its confidence level.

5.4.2.1. Weighted Pseudocounts Heuristics. We transform wALS scheme to count representations as follows. Let us denote the total negative tokens by N_+^- and the total positive tokens by N_+^+. Here, any value combinations for N_+^- and N_+^+ can be chosen, but Pan et al. [6] suggests to choose an equal total weight for positive and negative samples. Following [6], we choose to use the same amount of tokens for both positive and negative samples with $N_+^- = N_+^+ = \sum_{ij}(1 - Y_{ij})$ since the number of negative samples is generally larger than the number of positive samples because of network sparsity.

Once the total number of tokens is decided, they need to be redistributed according to their confidence levels. Since we have a constant confidence level for positive indices, N_+^+ positive tokens are distributed uniformly. Then, N_+^- negative tokens are distributed according to three schemes:

(i) *Uniform*: Each negative sample is represented by a single token, $(N_{ij}^- = 1)$,
(ii) *Source-only*: Each negative sample is represented by a number of tokens proportional to the source degree, $N_{ij}^- \sim \sum_j Y_{ij}$
(iii) *Source-dest*: Each negative sample is represented by a number of tokens proportional to the product of source and destination degrees, $N_{ij}^- \sim (\sum_j Y_{ij})(\sum_i Y_{ij})$.

Notice that tokens in N_+^+ are distributed evenly and stay constant in all cases. It is the negative tokens that are distributed according to distribution schemes. The negative token distributions according to cases (i), (ii) and (iii), and their difference from the count representation provided in Figure 5.1 are illustrated in Figure 5.8.

We performed experiments on three networks with respect to two weighting schemes: (i) uniform and (iii) proportional to source and destination popularity. The results are displayed in Figure 5.9.

For the (i) *uniform* case of Karate Club and Word Adjacency networks, BAM-MMSB estimates blocks that have a leader-follower topology instead of an assortative topology. This is a well-known characteristic of the blocks inferred by standard SBMs.

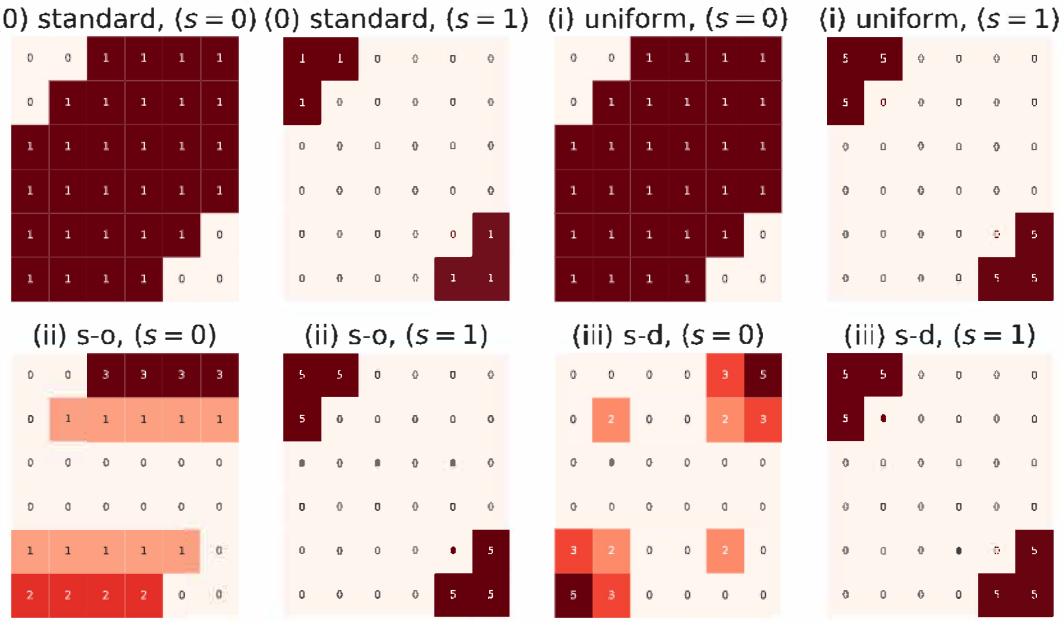

Figure 5.8. Weighted pseudocounts of the contingency tensor for each weighting scheme.

Specifically, the generative model tends to cluster nodes with similar degrees into the same block. As shown in the top row of Figure 5.9, this behavior results in two blocks where the green ones consisting of low-degree nodes (followers) seem to be following the red ones consisting of high-degree nodes (leaders).

Interestingly, the model behaves similar to the degree-corrected extension of stochastic blockmodels for the (iii) *source-dest* case. In this case, we obtain blocks with heterogeneous degree distributions in contrast to standard SBMs. This effect shifts the estimated topologies from leader-follower to assortative in Karate Club and Word Adjacency networks respectively. Scaling the negative pseudocounts with respect to the source and destination degrees brings up the same effect, even though we do not re-weight positive samples. We opt to keep the confidence level constant for each positive observation. The estimated model orders $K_{est} = 2$ are the same with (i) uniform case and commonly proposed model orders for these networks in the literature. However, variational approximations for the marginal likelihood are slightly larger for all networks in (iii) source-dest, which also suggests that degree-corrected extensions are favored over regular SBMs for these networks.

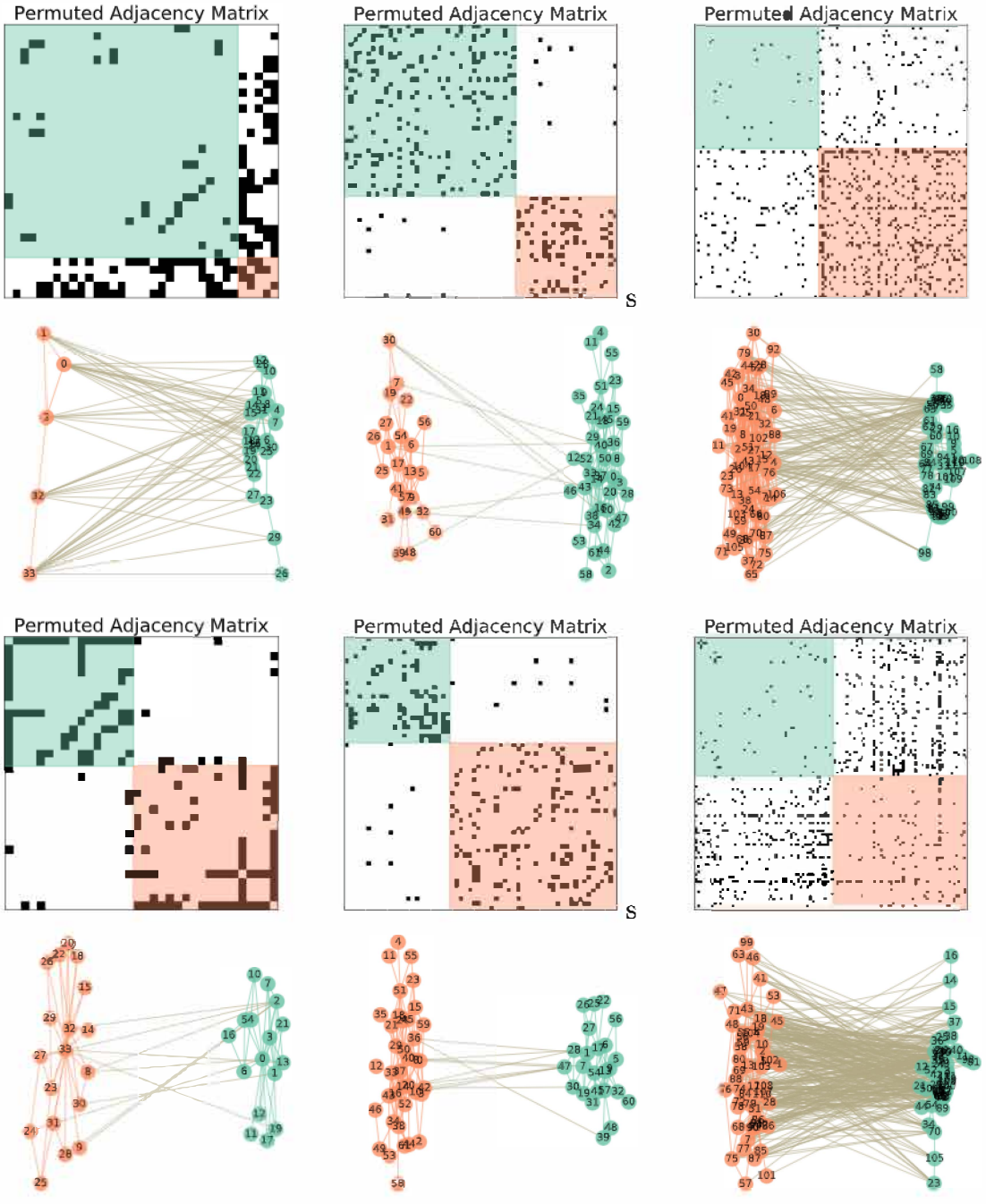

Figure 5.9. Model selection for Karate, Dolphins, Word-Adj. networks from left to right. Top and bottom rows correspond to uniform and source-dest respectively.

6. CONCLUSION

In this thesis, we propose BAM-MMSB which replicates the generative process of the MMSB within the generic allocation framework of BAM. Our model considers the observations as Poisson tokens generated by a Poisson process and marked according to the generative process of MMSB. From a modeling perspective, two Poisson random variables can represent each Bernoulli element Y_{ij} of the input matrix by adding a new index s for each (i, j) pair. This representation is equivalent to a Bernoulli trial when the sum is constrained to 1. Besides, this feature provides a natural extension possibility to weighted graphs or hypergraphs for future work.

A variational Bayes algorithm is derived to solve the inference problem. The first experiment illustrates the interpretation of the model output through synthetic network examples. Next, block recovery performance is analyzed numerically in the second experiment. As expected, BAM-MMSB displays a similar behavior with the original MMSB in the first two experiments. Furthermore, it is worth noting that uniform membership vectors and increased complexity in block structure reduce the block recovery performance.

Our model selection algorithm approximates the marginal likelihood by variational evidence lower bound to select the optimal number of blocks K_{opt}. The experiment results on real-world benchmark networks are similar to the results in the literature. However, the weighted count heuristics proposed by Pan *et al.* in [6] provide limited extendability of the task in hand since they are only heuristically motivated. A more principled approach is to integrate these heuristics into the model as random variables and infer their characteristics from the observed data [7].

Additionally, BAM offers a generic allocation framework which allows for rapid prototyping of distinct generative models of discrete observations. Therefore, another natural future direction is to perform model selection not only for the model order but also across different generative models such as NMF and tensor factorization models

proposed by Schein *et al.* [52, 53]. In this respect, another task for future work is to compute the exact marginal likelihood via annealed importance sampling instead of approximating it.

REFERENCES

1. David, E. and K. Jon, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*, Cambridge University Press, New York, NY, USA, 2010.

2. Deerwester, S., S. T. Dumais, G. W. Furnas, T. K. Landauer and R. Harshman, "Indexing by Latent Semantic Analysis", *Journal of the American Society for Information Science*, Vol. 41, No. 6, pp. 391–407, 1990, `http://superbook.bellcore.com/~remde/lsi/LSI.papers.html`.

3. Newman, M. E., "Complex systems: A survey", *arXiv preprint arXiv:1112.1440*, 2011.

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

5. Peixoto, T. P., "Bayesian stochastic blockmodeling", *arXiv preprint arXiv:1705.10225*, 2017.

6. Pan, R., Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz and Q. Yang, "One-class collaborative filtering", *2008 Eighth IEEE International Conference on Data Mining*, pp. 502–511, IEEE, 2008.

7. Rubin, D. B., "Inference and missing data", *Biometrika*, Vol. 63, No. 3, pp. 581–592, 1976.

8. Little, R. J. and D. B. Rubin, *Statistical analysis with missing data*, Vol. 793, Wiley, 2019.

9. Marlin, B., R. S. Zemel, S. Roweis and M. Slaney, "Collaborative filtering and the missing at random assumption", *arXiv preprint arXiv:1206.5267*, 2012.

10. Hernández-Lobato, J. M., N. Houlsby and Z. Ghahramani, "Probabilistic matrix factorization with non-random missing data", *International Conference on Machine Learning*, pp. 1512–1520, 2014.

11. Newman, M. E. and M. Girvan, "Finding and evaluating community structure in networks", *Physical review E*, Vol. 69, No. 2, p. 026113, 2004.

12. Fortunato, S. and D. Hric, "Community detection in networks: A user guide", *physrep*, Vol. 659, pp. 1–44, Nov. 2016.

13. Tan, F., Y. Xia and B. Zhu, "Link prediction in complex networks: a mutual information perspective", *PloS one*, Vol. 9, No. 9, p. e107056, 2014.

14. 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", *Proceedings of the international biometrics society annual meeting*, Vol. 15, 2006.

15. Salathé, M. and J. H. Jones, "Dynamics and control of diseases in networks with community structure", *PLoS computational biology*, Vol. 6, No. 4, p. e1000736, 2010.

16. Chen, W., Y. Wang and S. Yang, "Efficient influence maximization in social networks", *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 199–208, ACM, 2009.

17. Gerlach, M., T. P. Peixoto and E. G. Altmann, "A network approach to topic models", *Science advances*, Vol. 4, No. 7, p. eaaq1360, 2018.

18. Newman, M. E., "Modularity and community structure in networks", *Proceedings of the national academy of sciences*, Vol. 103, No. 23, pp. 8577–8582, 2006.

19. Riolo, M. A., G. T. Cantwell, G. Reinert and M. E. Newman, "Efficient method for

estimating the number of communities in a network", *Physical review e*, Vol. 96, No. 3, p. 032310, 2017.

20. Latouche, P., E. Birmele and C. Ambroise, "Variational Bayesian inference and complexity control for stochastic block models", *Statistical Modelling*, Vol. 12, No. 1, pp. 93–115, 2012.

21. Peixoto, T. P., "Model selection and hypothesis testing for large-scale network models with overlapping groups", *Physical Review X*, Vol. 5, No. 1, p. 011033, 2015.

22. Holland, P. W., K. B. Laskey and S. Leinhardt, "Stochastic blockmodels: First steps", *Social networks*, Vol. 5, No. 2, pp. 109–137, 1983.

23. Latouche, P., E. Birmelé, C. Ambroise *et al.*, "Overlapping stochastic block models with application to the french political blogosphere", *The Annals of Applied Statistics*, Vol. 5, No. 1, pp. 309–336, 2011.

24. Airoldi, E. M., D. M. Blei, S. E. Fienberg and E. P. Xing, "Mixed membership stochastic blockmodels", *Journal of machine learning research*, Vol. 9, No. Sep, pp. 1981–2014, 2008.

25. Karrer, B. and M. E. Newman, "Stochastic blockmodels and community structure in networks", *Physical review E*, Vol. 83, No. 1, p. 016107, 2011.

26. Blei, D. M., A. Y. Ng and M. I. Jordan, "Latent Dirichlet Allocation", *Journal of Machine Learning Research*, Vol. 3, pp. 993–1022, 2003, `http://www.jmlr.org/papers/v3/blei03a.html`.

27. Lee, D. D. and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization", *Nature*, Vol. 401, pp. 788–791, 1999.

28. Buntine, W., "Variational Extensions to EM and Multinomial PCA", T. Elomaa,

H. Mannila and H. Toivonen (Editors), *Machine Learning: ECML 2002*, pp. 23–34, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

29. Canny, J. F., "GaP: a factor model for discrete data", *SIGIR 2004: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, July 25-29, 2004*, pp. 122–129, 2004, https://doi.org/10.1145/1008992.1009016.

30. Cemgil, A. T., "Bayesian Inference for Nonnegative Matrix Factorisation Models", *Comp. Int. and Neurosc.*, Vol. 2009, pp. 785152:1–785152:17, 2009, https://doi.org/10.1155/2009/785152.

31. Buntine, W. and A. Jakulin, "Discrete Component Analysis", C. Saunders, M. Grobelnik, S. Gunn and J. Shawe-Taylor (Editors), *Subspace, Latent Structure and Feature Selection*, pp. 1–33, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

32. Paisley, J. W., D. M. Blei and M. I. Jordan, "Bayesian Nonnegative Matrix Factorization with Stochastic Variational Inference", E. M. Airoldi, D. M. Blei, E. A. Erosheva and S. E. Fienberg (Editors), *Handbook of Mixed Membership Models and Their Applications*, pp. 205–224, Chapman and Hall/CRC, 2014, http://www.crcnetbase.com/doi/abs/10.1201/b17520-15.

33. Taylan Cemgil, A., M. Burak Kurutmaz, S. Yildirim, M. Barsbey and U. Simsekli, "Bayesian Allocation Model: Inference by Sequential Monte Carlo for Nonnegative Tensor Factorizations and Topic Models using Polya Urns", *arXiv e-prints*, p. arXiv:1903.04478, Mar 2019.

34. Psorakis, I., S. Roberts, M. Ebden and B. Sheldon, "Overlapping community detection using bayesian non-negative matrix factorization", *Physical Review E*, Vol. 83, No. 6, p. 066114, 2011.

35. Koren, Y., R. Bell and C. Volinsky, "Matrix factorization techniques for recommender systems", *Computer*, , No. 8, pp. 30–37, 2009.

36. Févotte, C., E. Vincent and A. Ozerov, "Single-channel audio source separation with NMF: divergences, constraints and algorithms", *Audio Source Separation*, pp. 1–24, Springer, 2018.

37. Blei, D. M., "Build, compute, critique, repeat: Data analysis with latent variable models", *Annual Review of Statistics and Its Application*, Vol. 1, pp. 203–232, 2014.

38. Hofmann, T., "Probabilistic latent semantic analysis", *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pp. 289–296, Morgan Kaufmann Publishers Inc., 1999.

39. Ross, S. M., *Introduction to Probability Models*, Academic Press, 1997.

40. Mnih, A. and R. R. Salakhutdinov, "Probabilistic matrix factorization", *Advances in neural information processing systems*, pp. 1257–1264, 2008.

41. Murphy, K. P., *Machine learning: a probabilistic perspective*, MIT press, 2012.

42. Airoldi, E. M., D. M. Blei, S. E. Fienberg and E. P. Xing, "Mixed Membership Stochastic Blockmodels", *Journal of Machine Learning Research*, Vol. 9, pp. 1981–2014, 2008, `http://doi.acm.org/10.1145/1390681.1442798`.

43. Dempster, A. P., N. M. Laird and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society: Series B (Methodological)*, Vol. 39, No. 1, pp. 1–22, 1977.

44. Bishop, C. M., *Pattern recognition and machine learning*, Springer, 2006.

45. Neal, R. M., "Annealed importance sampling", *Statistics and computing*, Vol. 11, No. 2, pp. 125–139, 2001.

46. Tabouy, T., P. Barbillon and J. Chiquet, "Variational Inference for Stochastic Block Models from Sampled Data", *Journal of the American Statistical Association*, , No.

just-accepted, pp. 1–20, 2019.

47. Rand, W. M., "Objective criteria for the evaluation of clustering methods", *Journal of the American Statistical association*, Vol. 66, No. 336, pp. 846–850, 1971.

48. Latouche, P., E. Birmelé, C. Ambroise *et al.*, "Model selection in overlapping stochastic block models", *Electronic journal of statistics*, Vol. 8, No. 1, pp. 762–794, 2014.

49. Zachary, W. W., "An information flow model for conflict and fission in small groups", *Journal of anthropological research*, Vol. 33, No. 4, pp. 452–473, 1977.

50. Lusseau, D., K. Schneider, O. J. Boisseau, P. Haase, E. Slooten and S. M. Dawson, "The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations", *Behavioral Ecology and Sociobiology*, Vol. 54, No. 4, pp. 396–405, 2003.

51. Newman, M. E., "Finding community structure in networks using the eigenvectors of matrices", *Physical review E*, Vol. 74, No. 3, p. 036104, 2006.

52. Schein, A., J. W. Paisley, D. M. Blei and H. M. Wallach, "Bayesian Poisson Tensor Factorization for Inferring Multilateral Relations from Sparse Dyadic Event Counts", *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pp. 1045–1054, 2015, `https://doi.org/10.1145/2783258.2783414`.

53. Schein, A., M. Zhou, D. M. Blei and H. M. Wallach, "Bayesian Poisson Tucker Decomposition for Learning the Structure of International Relations", *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pp. 2810–2819, 2016, `http://jmlr.org/proceedings/papers/v48/schein16.html`.

APPENDIX A: DERIVATIONS OF BAM-MMSB INFERENCE

Variational inference is a method where we try to approximate intractable posterior distribution $p(Z|X)$ with a fully factorized distribution $q(Z)$. In any latent variable model, we can write the following equalities for p and q:

$$\log p(X) = L(q) + KL(q \,||\, p),$$

where

$$L(q) = \int_Z dZ\; q(Z) \log \left(\frac{p(X,Z)}{q(Z)}\right)$$

$$KL(q \,||\, p) = -\int_Z dZ\; q(Z) \log \left(\frac{p(Z|X)}{q(Z)}\right)$$

Since $KL(q \,||\, p)$ is non-negative, $L(q)$ provides a lower bound for the log likelihood. Furthermore, latent variable set is $Z = \{S, \theta, \lambda\}$ and X corresponds to the contingency matrix in *BAM*. So, we can choose the following factorized joint distribution for $q(Z)$:

$$q(S, \theta, \lambda) = q(S) \cdot q(\theta) \cdot q(\lambda)$$

Then, we can work on the lower bound $L(q)$ to find suitable distributions for $q(S)$, $q(\theta)$ and $q(\lambda)$.

A.1. Update Equation for $q(S)$

We can work on the definition of ELBO so that we derive a closed form solution for the variational distribution $q(S)$.

$$
\begin{aligned}
L(q) &= \sum_S \int_{\theta,\lambda} q(S,\theta,\lambda) \log \left(\frac{p(X,S,\theta,\lambda)}{q(S,\theta,\lambda)}\right) \\
&= \sum_S \int_{\theta,\lambda} d\theta\, d\lambda\, q(S,\theta,\lambda) \log \left(\frac{p(X,S,\theta,\lambda)}{q(S,\theta,\lambda)}\right) \\
&= \sum_S q(S) \int_{\theta,\lambda} q(\theta)q(\lambda) \log p(X,S,\theta,\lambda) - \sum_S q(S) \log q(S) + const \\
&= \sum_S q(S) \log \hat{p}(X,S) - \sum_S q(S) \log q(S) + const, \\
&\text{where } \log \hat{p}(X,S) = \int_{\theta,\lambda} q(\theta)q(\lambda) \log p(X,S,\theta,\lambda) = E_{q(\theta),q(\lambda)}[\log p(X,S,\theta,\lambda)] \\
&= -KL(q(S) \,||\, \hat{p}(X,S))
\end{aligned}
$$

Hence, when $q(\neg S)$ is fixed, maximizing the lower bound $L(q)$ is equivalent to minimizing KL divergence between $q(S)$ and $\hat{p}(X,S)$. The minimum is obtained when $q(S) = \hat{p}(X,S)$. Then, the general expression becomes;

$$
\begin{aligned}
\log q(S) &= E_{q(\theta),q(\lambda)}[\log p(X,S,\theta,\lambda)] + const \\
q(S) &\propto \exp\left(E_{q(\theta),q(\lambda)}[\log p(X,S,\theta,\lambda)]\right) \\
&= \frac{\exp\left(E_{q(\theta),q(\lambda)}[\log p(X,S,\theta,\lambda)]\right)}{\sum_S \exp\left(E_{q(\theta),q(\lambda)}[\log p(X,S,\theta,\lambda)]\right)}
\end{aligned}
$$

Following factorization of the form;

$$
p(X,S,\theta,\lambda) = p(S|X,\theta,\lambda)p(X,\theta,\lambda) = p(S|X)p(X,\theta,\lambda),
$$

$q(S)$ is proportional to;

$$
\begin{aligned}
q(S) &\propto \exp\left(E_{q(\theta),q(\lambda)}[\log p(S|X,\theta,\lambda)]\right) \\
&\propto \exp\Bigg(\sum_{ikj} S_{ikj} \cdot E_q\Big[log(\frac{\lambda\theta_{5|2,4}(s,k,l)\theta_{2|1}(k,i)\theta_{4|3}(l,j)\theta_1(i)\theta_3(j)}{\lambda\sum_{k,l}\theta_{5|2,4}(s,k,l)\theta_{2|1}(i,k)\theta_{4|3}(l,j)\theta_1(i)\theta_3(j)})\Big] \\
&- \sum_{kl} \log\Gamma(S_{ikslj}+1) + \sum_{ijs}\log\Gamma(X_{ijs}+1) + \sum_{ijs}\log \mathbf{1}[X_{ijs}=\sum_{kl} S_{ikslj}]\Bigg) \\
&\propto \prod_{i,j} M(S_{2,4|1,3,5}(i,:,s,:,j); X_{ijs}, p_{2,4|1,3,5}(i,:,s,:,j)),
\end{aligned}
$$

where

$$
\begin{aligned}
p(i,:,s,:,j)) &\propto E_{q(\theta),q(\lambda)}\left[log(\lambda\theta_{5|2,4}(s,:,:)\theta_{2|1}(:,i)\theta_{4|3}(:,j)\theta_1(i)\theta_3(j))\right] \\
&= \frac{E_q[log\lambda] + E_q[\theta_{5|2,4}(s,:,:) + \theta_{4|3}(:,j) + \theta_{2|1}(:,i) + \theta_3(j) + \theta_1(i)]}{\sum_{kl} E_q[log\lambda] + E_q[\theta_{5|2,4}(s,:,:) + \theta_{4|3}(:,j) + \theta_{2|1}(:,i) + \theta_3(j) + \theta_1(i)]}
\end{aligned}
$$

A.2. Update Equation for $q(\theta)$

Following the factorization of the form; $p(X, S, \theta, \lambda) = p(X|S)p(\theta|S)p(\lambda|S)p(S)$, $q(\theta)$ is proportional to:

$$
\begin{aligned}
q(\theta) &\propto \exp\left(E_{q(S),q(\lambda)}[\log p(S, X, \theta, \lambda)]\right) \\
&\propto \exp\left(E_{q(S),q(\lambda)}[\log p(\theta|S)]\right) \\
&\propto \exp\Big(\sum_{ki}(E_{q(S)}[S_{+ksl+}] + \alpha_{5|2,4} - 1)\log\theta_{5|2,4} \\
&+ \sum_{ki}(E_{q(S)}[S_{ik+++}] + \alpha_{2|1} - 1)\log\theta_{2|1} + \sum_{lj}(E_{q(S)}[S_{+++lj}] + \alpha_{4|3} - 1)\log\theta_{4|3} \\
&+ \sum_{i}(E_{q(S)}[S_{i+++}] + \alpha_1 - 1)\log\theta_1\Big) + \sum_{j}(E_{q(S)}[S_{++++j}] + \alpha_3 - 1)\log\theta_3\Big) \\
&\propto \exp\Big(\sum_{pa(n)}\sum_{n}(E_{q(S)}[S_{n|pa(n)}] + \alpha_{n|pa(n)} - 1)\log\theta_{n|pa(n)}\Big) \\
&\propto D(E_{q(S)}[S_{n|pa(n)}] + \alpha_{n|pa(n)})
\end{aligned}
$$

A.3. Update Equation for $q(\lambda)$

Following factorization of the form; $p(X, S, \theta, \lambda) = p(X|S)p(\theta|S)p(\lambda|S)p(S)$, $q(\lambda)$ is proportional to;

$$
\begin{aligned}
q(\lambda) &\propto \exp\left(E_{q(S),q(\theta)}[\log p(S, X, \theta, \lambda)]\right) \\
&\propto \exp\left(E_{q(S),q(\theta)}[\log p(\lambda|S)]\right) \\
&\propto \exp\left(E_{q(S),q(\theta)}[-\lambda(b+1) + (a + S_+ - 1)\log\lambda]\right) \\
&\propto Gamma(E_{q(S)}[S_+] + a, b + 1)
\end{aligned}
$$

A.4. Computing ELBO

Following factorization of the form; $p(X, S, \theta, \lambda) = p(X|S)p(\theta|S)p(\lambda|S)p(S)$, the evidence lower bound $L(q)$ is proportional to;

$$\begin{aligned} L(q) &= \sum_{S} \int_{\theta,\lambda} q(S, \theta, \lambda) \log \left(\frac{p(X, S, \theta, \lambda)}{q(S, \theta, \lambda)}\right) \\ &= \mathbb{E}_q[p(X|S)] + \mathbb{E}_q[p(\theta|S)] + \mathbb{E}_q[p(\lambda|S)] + \mathbb{E}_q[p(S)] \\ &- \mathbb{E}_q[q(S)] - \mathbb{E}_q[q(\theta)] - \mathbb{E}_q[q(\lambda)] \\ &= F[Q] - H[Q] \end{aligned}$$

where $F[Q]$ is:

$$
\begin{aligned}
F[Q] \equiv\ & \mathbb{E}_q[\log p(X|S)] + \mathbb{E}_q[\log p(\theta_i|S)] + \mathbb{E}_q[\log p(\theta_j|S)] + \mathbb{E}_q[\log p(\theta_{k|i}|S)] \\
& + \mathbb{E}_q[\log p(\theta_{l|j}|S)] + \mathbb{E}_q[\log p(\theta_{s|k,l}|S)] \\
& = -\mathbb{E}_q[\log B(\alpha_i + S_{i++++})] + \sum_i (\alpha_i + \mathbb{E}_q[S_{i++++}] - 1)\,\mathbb{E}_q[\log \theta_i] \\
& - \mathbb{E}_q[\log B(\alpha_j + S_{++++j})] + \sum_j (\alpha_j + \mathbb{E}_q[S_{++++j}] - 1)\,\mathbb{E}_q[\log \theta_j] \\
& - \sum_i \mathbb{E}_q[\log B(\alpha_{ik+++} + S_{ik+++})] + \sum_{ik} (\alpha_{ik+++} + \mathbb{E}_q[S_{ik+++}] - 1)\,\mathbb{E}_q[\log \theta_{k|i}] \\
& - \sum_j \mathbb{E}_q[\log B(\alpha_{+++lj} + S_{+++lj})] + \sum_{lj} (\alpha_{+++lj} + \mathbb{E}_q[S_{+++lj}] - 1)\,\mathbb{E}_q[\log \theta_{l|j}] \\
& - \sum_{kl} \mathbb{E}_q[\log B(\alpha_{+ksl+} + S_{+ksl+})] + \sum_{ksl} (\alpha_{+ksl+} + \mathbb{E}_q[S_{+ksl+}] - 1)\,\mathbb{E}_q[\log \theta_{s|k,l}] \\
& + a \log b - (a + \mathbb{E}_q[S_+]) \log(b+1) + \mathbb{E}_q[\log \Gamma(a + S_+)] - \log \Gamma(a) \\
& - \sum_{ikslj} \mathbb{E}_q[\log \Gamma(S_{ikslj} + 1)] \\
& + \mathbb{E}_q[\log B(\alpha_i + S_{i++++})] + \mathbb{E}_q[\log B(\alpha_j + S_{++++j})] \\
& + \sum_i \mathbb{E}_q[\log B(\alpha_{ik+++} + S_{ik+++})] \\
& + \sum_j \mathbb{E}_q[\log B(\alpha_{+++lj} + S_{+++lj})] + \sum_{kl} \mathbb{E}_q[\log B(\alpha_{+ksl+} + S_{+ksl+})] \\
& - \log B(\alpha_j) - \log B(\alpha_i) - \log B(\alpha_{ik}) - \log B(\alpha_{lj}) - \log B(\alpha_{ksl}) \\
& + (a + \mathbb{E}_q[S_+]) \log(b+1) - \mathbb{E}_q[\log \Gamma(a + S_+)] + (a + \mathbb{E}_q[S_+] - 1)\,\mathbb{E}_q[\log \lambda] \\
& - (b+1)\,\mathbb{E}_q[\lambda]
\end{aligned}
$$

and $H[Q]$ is:

$$
\begin{aligned}
H[Q] &\equiv \mathbb{E}_q[q(S)] + \mathbb{E}_q[q(\theta)] + \mathbb{E}_q[q(\lambda)] \\
&= \sum_{iksljq} \mathbb{E}_q[S_{iksljq}] \log p_{kl|isjq} + \sum_{isjq} \log \Gamma(X_{isjq} + 1) - \sum_{iksljq} \mathbb{E}_q[\log \Gamma(S_{iksljq} + 1)] \\
&- \log B(\alpha_i + \mathbb{E}_q[S_{i++++}]) + \sum_i (\alpha_i + \mathbb{E}_q[S_{i++++}] - 1)\, \mathbb{E}_q[\log \theta_i] \\
&- \log B(\alpha_j + \mathbb{E}_q[S_{++++j}]) + \sum_j (\alpha_j + \mathbb{E}_q[S_{++++j}] - 1)\, \mathbb{E}_q[\log \theta_j] \\
&- \sum_i \log B(\alpha_{ik+++} + \mathbb{E}_q[S_{ik+++}]) + \sum_{ik} (\alpha_{ik+++} + \mathbb{E}_q[S_{ik+++}] - 1)\, \mathbb{E}_q[\log \theta_{k|i}] \\
&- \sum_j \log B(\alpha_{+++lj} + \mathbb{E}_q[S_{+++lj}]) + \sum_{lj} (\alpha_{+++lj} + \mathbb{E}_q[S_{+++lj}] - 1)\, \mathbb{E}_q[\log \theta_{l|j}] \\
&- \sum_{kl} \log B(\alpha_{+ksl+} + \mathbb{E}_q[S_{+ksl+}]) + \sum_{ksl} (\alpha_{+ksl+} + \mathbb{E}_q[S_{+ksl+}] - 1)\, \mathbb{E}_q[\log \theta_{s|k,l}] \\
&+ (a + \mathbb{E}_q[S_+]) \log(b+1) - \log \Gamma(a + \mathbb{E}_q[S_+]) + (a + \mathbb{E}_q[S_+] - 1)\, \mathbb{E}_q[\log \lambda] \\
&- (b+1)\, \mathbb{E}_q[\lambda]
\end{aligned}
$$

then, $F[Q] - H[Q]$ becomes:

$$
\begin{aligned}
F[Q] - H[Q] \equiv & \\
&+ a \log b - (a + \mathbb{E}_q[S_+]) \log(b+1) + \log \Gamma(a + \mathbb{E}_q[S_+]) - \log \Gamma(a) \\
&+ \log B(\alpha_i + \mathbb{E}_q[S_{i++++}]) + \log B(\alpha_j + \mathbb{E}_q[S_{++++j}]) \\
&+ \sum_i \log B(\alpha_{ik+++} + \mathbb{E}_q[S_{ik+++}]) + \sum_j \log B(\alpha_{+++lj} + \mathbb{E}_q[S_{+++lj}]) \\
&+ \sum_{kl} \log B(\alpha_{+ksl+} + \mathbb{E}_q[S_{+ksl+}]) \\
&- \log B(\alpha_j) - \log B(\alpha_i) - \log B(\alpha_{k|i}) - \log B(\alpha_{l|j}) - \log B(\alpha_{s|kl}) \\
&- \sum_{iksljq} \mathbb{E}_q[S_{iksljq}] \log p_{kl|isjq} - \sum_{isjq} \log \Gamma(X_{isjq} + 1)
\end{aligned}
$$

APPENDIX B: DERIVATIONS OF BAM-MMSB-MISSING INFERENCE

We can choose the following factorized joint distribution for $q(Z)$:

$$q(S^o, S^m, \theta, \lambda) = q(S^o) \cdot q(S^m) \cdot q(\theta) \cdot q(\lambda)$$

B.1. Update Equations for Missing at Random Case

B.1.1. Update Equations For $q(S)$: $q(S^o)$ and $q(S^m)$

For the observed indices $(ijs) \in X^o$, $q(S^o)$ is distributed multinomially. On the other hand, for the missing indices $(ijs) \in X^m$, $q(S^m)$ is a Poisson distribution.

$$q(S^o) \propto \prod_{ijs:(ijs)\in X^o} \mathcal{M}(S_{2,4|1,3,5}(i,:,s,:,j); X_{ijs}, p_{2,4|1,3,5}(i,:,s,:,j))$$

where $p_{k,l|i,s,j}(i,:,s,:,j)$ is:

$$p_{k,l|i,s,j}(i,:,s,:,j) \propto \mathbb{E}_q[\log(\lambda\theta_{s|k,l}(s,:,:)\theta_{k|i}(:,i)\theta_{l|j}(:,j)\theta_i(i)\theta_j(j))]$$

Next, we investigate $q(S^m)$:

$$q(S^m) \propto \prod_{ikslj:(ijs)\in X^m} \mathcal{PO}(S_{ikslj}; \tau_{ikslj})$$

where τ_{ikslj} is:

$$\tau_{ikslj} = \mathbb{E}_q[\log(\lambda\theta_{s|k,i}\theta_{k|i}\theta_{l|j}\theta_i\theta_j)]$$

As a result,

$$\mathbb{E}_q[S_{ikslj}] = \begin{cases} X_{ijs} \times p_{ikslj}, & \text{for } (ijs) \in X^o \\ \tau_{ikslj}, & \text{for } (ijs) \in X^m \end{cases}$$

B.1.2. Update Equation For $q(\theta)$

The update equation is same as the case where there are no missing values.

$$q(\theta) \propto \mathcal{D}(E_{q(S)}[S_{n|pa(n)}] + \alpha_{n|pa(n)})$$

B.1.3. Update Equation For $q(\lambda)$

The update equation is same as the case where there are no missing values.

$$q(\lambda) \propto \mathcal{G}(E_{q(S)}[S_+] + a, b + 1)$$

B.2. Computing ELBO

We start by writing the variational lower bound definition for the missing at random case.

$$\begin{aligned} L_{MAR}(q) &= \sum_S \int_{\theta,\lambda} q(S,\theta,\lambda) \log\left(\frac{p(X,S,\theta,\lambda)}{q(S,\theta,\lambda)}\right) \\ &= E_q[p(X|S)] + E_q[p(\theta|S)] + E_q[p(\lambda|S)] + E_q[p(S)] \\ &- E_q[q(S^o)] - E_q[q(S^m)] - E_q[q(\theta)] - E_q[q(\lambda)] \end{aligned}$$

Notice that the only difference from the fully observed ELBO is the factor of $E_q[q(S^m)]$. This brings up the following result:

$$
\begin{aligned}
F[Q] - H[Q] \equiv & \\
& + a \log b - (a + \mathbb{E}_q[S_+]) \log(b+1) + \log \Gamma(a + \mathbb{E}_q[S_+]) - \log \Gamma(a) \\
& + \log B(\alpha_i + \mathbb{E}_q[S_{i++++}]) + \log B(\alpha_j + \mathbb{E}_q[S_{++++j}]) \\
& + \sum_i \log B(\alpha_{ik+++} + \mathbb{E}_q[S_{ik+++}]) + \sum_j \log B(\alpha_{+++lj} + \mathbb{E}_q[S_{+++lj}]) \\
& + \sum_{kl} \log B(\alpha_{+ksl+} + \mathbb{E}_q[S_{+ksl+}]) \\
& - \log B(\alpha_j) - \log B(\alpha_i) - \log B(\alpha_{k|i}) - \log B(\alpha_{l|j}) - \log B(\alpha_{s|kl}) \\
& - \sum_{ikslj:(ijs)\in X^\bullet} \mathbb{E}_q[S_{ikslj}] \log p_{kl|isj} - \sum_{isj:(ijs)\in X^\bullet} \log \Gamma(X_{isj} + 1) \\
& - \sum_{ikslj:(ijs)\in X^m} \mathbb{E}_q[S_{ikslj}] \log \tau_{ikslj} + \sum_{ikslj:(ijs)\in X^m} \tau_{ikslj}
\end{aligned}
$$