MODELING BOUNDED DATA WITH SUM CONDITIONED

POISSON FACTORIZATION

by

Taha Yusuf Ceritli

B.S., Electrical and Electronics Engineering, Boğaziçi University, 2015

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 Computational Science and Engineering

Boğaziçi University

2017

MODELING BOUNDED DATA WITH SUM CONDITIONED

POISSON FACTORIZATION

APPROVED BY:

Assoc. Prof. Ali Taylan Cemgil   . . . .

(Thesis Supervisor)

Assoc. Prof. Albert Ali Salah   . . . .

Assist. Prof. Sinan Yıldırım   . . . .

DATE OF APPROVAL: 03.08.2017

ACKNOWLEDGEMENTS

ABSTRACT

MODELING BOUNDED DATA WITH SUM
CONDITIONED
POISSON FACTORIZATION

Non-negative matrices appears in many domains from item recommendation, audio signal processing to computer vision in which data instances have a bounded non-negative range. For various tasks in these areas, probabilistic approaches have been widely applied where matrix factorizations are among the state-of-the-art methods. A particular one is a latent variable model called Poisson Factorization which models bounded data with Poisson distribution assigning them unbounded ranges. In this work, we extend Poisson Factorization to model bounded data with bounded distributions such as Bernoulli, Binomial, Categorical and Multinomial. The resulting model is named as Sum Conditioned Poisson Factorization as the model is constructed by conditioning multiple Poisson Factorizations on their sum.

We present two algorithms for inference in Sum Conditioned Poisson Factorization: Gibbs sampler and Expectation-Maximization. The algorithms and the model are tested with simulated and real data sets. First, we compare the algorithms with data generated from the model synthetically. Then, we demonstrate the interpretability of the model on a binary valued data set named Swimmer. In order to measure the performance of the model on ordinal ratings data, we use MovieLens 500-K. The results indicate that the proposed model outperforms Poisson Factorization and other models in terms of predictive performance for test ratings and top-K recommendation. Finally, we conduct experiments on piano roll data extracted from Bach Chorales for investigating the use of the model in time series. The experiments reveal that the model provides parameters that can be used for prior distribution in time series analysis.

ÖZET

SINIRLI VERİYİ TOPLAMA KOŞULLU POİSSON AYRIŞTIRMA İLE MODELLEME

Negatif olmayan matrisler, veri örneklerinin negatif olmayan sınırlı değerler aldığı kalem önerme ve ses sinyali işleme gibi birçok alanda karşımıza çıkmaktadır. Olasılıksal yaklaşımlar bu alanlardaki birçok görev için kullanılmakta ve matris ayrıştırma modelleri bu alandaki en ileri metodlar arasında yer almaktadır. Bunlardan biri Poisson Ayrıştırma adında gözlemleri Poisson dağılımıyla modelleyen bir saklı değişken modelidir. Ancak bu şekilde, değer aralığı sınırlı olan gözlemlere, Poisson dağılımıyla sınırlı olmayan değer aralığı verilmektedir. Bu çalışmada, Poisson Ayrıştırma genişletilmiş ve gözlemler Bernoulli, İkiterimli, Kategorik ve Çok terimli gibi sınırlı değer aralığına sahip dağılımlarla modellenmiştir. Ortaya çıkan model, birçok Poisson Ayrıştırmanın, kendilerinin toplamlarına koşullandırılmasıyla oluşturulduğundan Toplama Koşullu Poisson Ayrıştırması olarak adlandırılmıştır.

Toplama Koşullu Poisson Ayrıştırması modelinde çıkarım için iki algoritma sunuyoruz: Gibbs örnekleyicisi ve Beklenti-Enbüyütme. Algoritmalar ve model, benzeştirilmiş ve gerçek veri kümeleriyle test edilmiştir. İlk olarak, üretici modellen elde edilen sentetik veriyle, iki algoritmayı kıyaslıyoruz. Daha sonra, Swimmer adında bir veri kümesinde modelin yorumlanabilirliğini gösteriyoruz. Modelin sıralama ölçekli puanlama verisindeki performansı ölçmek içinse MovieLens 500-K adında kullanıcı film puan veri setini kullanıyoruz. Sonuçlar önerilen modelin, mevcut diğer modellerden test puanlarını tahmin etmede ve üst-K önermede daha üstün olduğunu gösteriyor. Son olarak, modelin zaman serisindeki kullanımını araştırmak için Bach Korallerinden çıkarılan piyano rulo verisiyle deney yapıyoruz. Burada, modelin zaman serisi analizinde önsel dağılımlar için kullanılabilecek parametreler sağladığını göstermekteyiz.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$A * B$	Inner product of two matrices A and B
$(A)^T$	Transpose operation applied to matrix A
$\mathbb{E}\big[\cdot\big]_q$	Expectation w.r.t. q function.
$h_{k,r,j}$	Latent excitation variable
$\mathcal{M}(\cdot;\cdot,\cdot)$	Multinomial distribution
$m_{k,i,j}$	Observed mask variable
$n_{i,j}$	Observed cardinality variable
$\tilde{n}_{i,j}$	Observed residual variable
$\mathcal{PO}(\cdot;\cdot)$	Poisson distribution
$s_{k,i,j,r}$	Latent source variable
$w_{k,i,r}$	Latent template variable
$x_{k,i,j}$	Observation variable
$\mathcal{G}(\cdot;\cdot,\cdot)$	Gamma distribution
$\Gamma(\cdot)$	Gamma function
$\delta(\cdot)$	Dirac delta function
$\mathcal{BE}(\cdot;\cdot)$	Bernoulli distribution
Θ	Parameter set
$\sigma(\cdot)$	Sigmoid function

LIST OF ACRONYMS/ABBREVIATIONS

2D	Two Dimensional
EM	Expectation-Maximization
GMF	Gaussian Matrix Factorization
KL	Kullback-Leibler
MAE	Mean Absolute Error
MAP	Mean Absolute Precision
MCMC	Markov chain Monte Carlo
ML	Maximum-Likelihood
NMF	Non-negative Matrix Factorization
OMF	Ordinal Matrix Factorization
PCA	Principal Component Analysis
PF	Poisson Factorization
RMSE	Root Mean Square Error
SCPF	Sum Conditioned Poisson Factorization
SGD	Stochastic Gradient Descent
VB	Variational Bayes
VQ	Vector Quantization

1. INTRODUCTION

During the last decades, tremendous amounts of data have became available. One interesting and popular type of data is dyadic data which consists of measurements on pairs where the observed measurement $x_{i,j}$ is assumed to carry information about the interaction of i and j. For instance, $x_{i,j}$ can be the rating of movie j given by user i.

Dyadic data can be modeled with latent variable models where latent or hidden variables are used to represent the underlying structures in data. These models assume that data instances are realizations of observed variables which are conditioned on hidden variables. Hence, even though latent variables are hidden and can not be observed directly, they can be inferred given observed variables.

Matrix factorization models can be seen as latent variable models which decomposes a given dyadic data into its factors. As data is often available in matrix form, these models can be used in diversed array of applications. In image processing, a matrix might consist of coefficients where x-axis and y-axis represent samples and features, respectively. Movie ratings can also be used with matrix factorization models where x-axis and y-axis denote users and movies, respectively. Text processing is another research area where matrices may be constructed by word counts in which x-axis and y-axis correspond to text documents and words, respectively. For audio processing applications, entries of matrices might be designed to include Fourier coefficients where x-axis and y-axis represent time and frequencies, respectively.

For modeling non-negative dyadic data, one popular matrix factorization model is Nonnegative Matrix Factorization (NMF). NMF models have been widely used in various domains including text mining [1, 2], computer vision [3, 4], document clustering [5, 6], audio signal processing [7–13], video processing [14–17], bioinformatics [18–21], community discovery [22, 23] and item recommendation [24, 25]. Generally, two problems are of interest in NMF: (i) exploration and (ii) prediction. In the first one, the goal is to understand the underlying structure of data thanks to the inter-

pretability of the model. For the latter, the task is to predict unobserved or unknown entries of data.

NMF is often able to solve the aforementioned problems by providing interpretable factors and accurate estimates of data. For modeling bounded data such as binary, categorical or ordinal, NMF with Kullback-Leibler divergence, also known as Poisson Factorization (PF), is one of the state-of-the-art methods. However, it might not be the best choice since NMF models observations with Poisson distribution assuming data instances are unbounded. On the other hand, to model bounded data alternative approaches to KL-NMF (PF) can be found in the literature. [26–29] But, they suffer from the interpretability issue since inferred components are difficult to analyze.

1.1. Scope of the Thesis

In this thesis, we propose a simple extension to KL-NMF (PF) to model bounded data such as Bernoulli, Binomial, Categorical and Multinomial. The resulting model still provides interpretable factors for understanding a given data set and performs a higher accuracy as shown in the experiments.

The main contribution of this thesis can be summarized as follows:

- Sum Conditioned Poisson Factorizatoin (SCPF): We first describe our model from a statistical perspective. We further derive algorithms for inference of latent variables in the model and present efficient and generic implementation of the algorithms.
- Comparison of Gibbs sampler and EM algorithm: We compare two methods on a synthetic data set generated from the model.
- Interpretability: We conduct binary matrix factorization experiments on a binary data set to show interpretability of factors in the model.
- Predictive performance: We demonstrate the predictive performance of our model on matrix-completion problem.

1.2. Organization of the Thesis

The rest of the thesis is organized as follows: Chapter 2 provides theoretical background needed to understand the model and the inference methods. Chapter 3 describes Sum Conditioned Poisson Factorization (SCPF) and discusses its properties. A detailed analysis of the inference for the model is given in Chapter 4. The experiments are presented in Chapter 5. Finally, the thesis is concluded and further research directions are explained.

2. THEORETICAL BACKGROUND

This chapter contains theoretical background to understand the work in the thesis. First, we describe Poisson Factorization (PF) which the proposed model is built upon. Then, we present introductory information on the algorithms used for inference in the proposed model.

2.1. Poisson Factorization

Non-negative Matrix Factorization (NMF) was introduced as positive matrix factorization by Paatero and Tapper [30] and popularized by Lee and Seung as an alternative method to Principal Component Analysis (PCA) and Vector Quantization (VQ) for facial image analysis and semantic analysis of documents [3]. In NMF, given non-negative data matrix X, the goal is to find two non-negative matrices W and H such that multiplication of matrices approximates X:

$$X \approx WH$$

where W and H are referred as Template and Excitation matrices, respectively.

General usages of NMF includes feature learning, topic discovery, clustering, temporal segmentation, filtering and source separation.
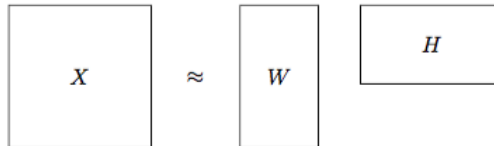
Figure 2.1. NMF as a matrix decomposition model. X, W and H are non-negative data, Template and Excitation matrices respectively.

NMF can also be cast to a optimization problem:

$$(W^*, H^*) = \operatorname*{argmin}_{W,H} D(X||WH)$$

where the error function D is a divergence.

Cemgil described NMF from a statistical perspective with Kullback-Leibler (KL) divergence as an error measure, allowing full Bayesian treatment of the model [4]. KL-NMF is also be known as Poisson Factorization (PF) in the literature. Gopalan *et. al.* proposes a hierarchical PF with a variational inference algorithm that scales up to massive data sets [31]. In [32], it is extended to a Bayesian nonparametric model which outperforms its parametric counterpart. These models generally assume that the latent factors are static. Charlin addresses this issue and presents Dynamic Poisson Factorization which models time evolving latent factors [33].

Each basis vector in Template matrix, each column of W, represents a fundamental characteristics in data. For example, consider the data in Figure 2.2, packet type histograms collected from a SIP network where rows and columns denote features and time, respectively. In this application, inferred basis vectors in W represent certain packet types histograms generated when a certain action in network is taken. Here, the basis vector with column id 2 stands for actions in which a user initiates a call to another by sending INVITE packets to a server. The entries of the Excitation matrix denotes the contribution of the corresponding behaviors to approximation.

PF can be interpreted in various ways one of which is as a feature extraction method. Inferred Template matrix contains local features extracted from a data set. Hence, corresponding vectors in Excitation matrix can be seen as feature vectors which are actually linear combinations of local features in Excitation matrix. PF can be also interpreted as a low-rank approximation method as dimension of the Template matrix is typically chosen as lower than dimension of the original data.
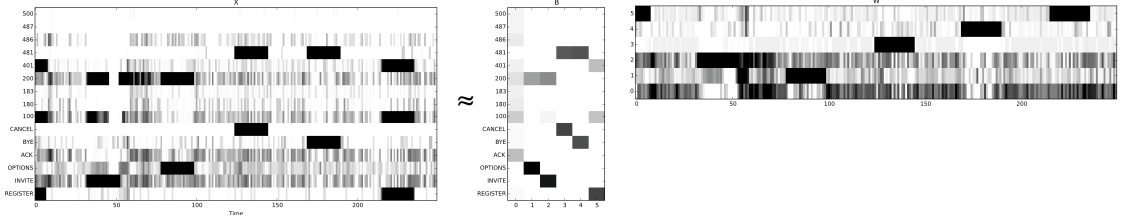
Figure 2.2. The first plot visualizes a network data, packet type histograms collected from a SIP network. The rest corresponds to template and excitation matrices inferred by PF, respectively.

Formal description of the model can be given as follows: Let X be a $I \times J$ sized data matrix whose each entry $x_{i,j}$ corresponds to the data at i^{th} row and j^{th} column where $i = 1, 2, \ldots, I$ and $j = 1, 2, \ldots, J$. The goal is to approximate X with the multiplication of two non-negative matrices W and H whose sizes are $I \times R$ and $R \times J$, respectively, where R denotes the rank of decomposition. Each entry $x_{i,j}$ is modeled as a sum of Poisson random variables which are denoted by $s_{i,j,1:R}$ where R denotes the number of hidden Poisson random variables. This allows $x_{i,j}$ to be a Poisson random variable as well, whose intensity parameter $\lambda_{i,j}$ is the sum of intensity parameters of other Poisson random variables $s_{i,j,1:R}$. The intensity parameter of each Poisson random variable $s_{i,j,r}$ is given as the product of $w_{i,r}$ and $h_{r,j}$. The non-negative entries in W and H are modeled with Gamma random variables since the Gamma distribution is the conjugate prior of Poisson distribution and its support is the set of non-negative real numbers.

Hence, the generative model of PF can be given as follows:

$$w_{i,r} \sim \mathcal{G}(w_{i,r}; a^w, b^w/a^w) \qquad\qquad h_{r,j} \sim \mathcal{G}(h_{r,j}; a^h, b^h/a^h)$$

$$s_{i,j,r} \sim \mathcal{PO}(s_{i,j,r}; w_{i,r} \times h_{r,j}) \qquad\qquad x_{i,j} = \sum_{r=1}^{R} s_{i,j,r}$$

where a^w and b^w/a^w are the shape and scale parameters of Gamma distributions used for the variables in W while a^h and b^h/a^h are the shape and scale parameters of Gamma distributions used for the variables in H. Note that this parameter choice makes b^w and b^h mean parameters of Gamma distributions. Here, $\mathcal{G}(\cdot)$ and $\mathcal{PO}(\cdot)$ denote Gamma and Poisson distributions which can be given as:

$$\mathcal{G}(w; a, b/a) = \frac{w^{a-1}\exp(-w\frac{a}{b})}{\Gamma(a)(b/a)^a}$$
$$\mathcal{PO}(s; \lambda) = \frac{\lambda^s \exp(-\lambda)}{s!}$$

where a, b and λ are the shape, scale and intensity parameters, respectively.

From a Bayesian perspective, the goal becomes calculating the joint posterior distribution of W and H given X. The multiplicative update rules in the original NMF paper [3] appear as Maximum-Likelihood (ML) estimates of latent variables with Expectation-Maximization (EM) algorithm for KL-NMF where priors on W and H are omitted [4].

Various hierarchical PF models can be found in the literature. For instance, Cemgil uses Variational Bayes and Gibbs sampler for inference in a hierarchical PF model [4]. In [31], Gopalan *et. al.* developed a model named hierarchical Poisson matrix factorization for recommendation in which variational inference is used for approximate posterior inference that scales up to massive data sets. Gopalan further proposed a Bayesian nonparametric Poisson factorization model for model selection where the latent components and the latent dimensionality are found simultaneously with an efficient algorithm based on variational inference [32].

2.2. The Expectation-Maximization (EM) Algorithm

Suppose we are given a model with data X, latent variables Z and an unknown parameter Θ. In order to estimate the unknown parameters, Maximum-Likelihood

(ML) which is defined as follows:

$$\hat{\Theta}_{ML} := \underset{\Theta}{\mathrm{argmax}}\, p(X|\Theta)$$

Whilst the former corresponds to finding parameter value that maximizes the marginal likelihood.

Consider the ML estimation procedure in which the objective function can be written as the marginalization of a joint distribution over the latent variables:

$$\hat{\Theta}_{ML} = \underset{\Theta}{\mathrm{argmax}}\, p(X|\Theta))$$
$$= \underset{\Theta}{\mathrm{argmax}}\, p(\sum_{Z} p(X, Z|\Theta))) \tag{2.1}$$

where sum is used for the marginalization as we assume Z is discrete without loss of generality.

The sum given in the Equation 2.1 is often intractable; however it is still possible to find $\hat{\Theta}_{ML}$ using Expectation-Maximization (EM) procedure.

EM is an iterative algorithm to find the Maximum-Likelihood (ML) or Maximum-A-Posteriori estimate of an unknown parameter in latent variable models [34]. At each iteration, new estimates for the unknown parameter are calculated using observations and current estimates of unknown parameter. This is carried out via finding parameter value that maximizes the expectation of the likelihood function $p(X, Z|\Theta)$ with respect to the posterior distribution of latent variables given current estimates. In order to show this, we now proceed with EM derivations.

Note that as $\log(x)$ is a strictly increasing function, maximizing $p(x|\Theta)$ is equivalent to maximizing $\log p(x|\Theta)$. This in turn lets one use the log-likelihood function which not only simplifies analysis but also provides a numerically more stable solution.

Let us first write the log-likelihood:

$$\log p(X|\Theta) = \log \left(\sum_Z p(X, Z|\Theta) \right)$$
$$= \log \left(\sum_Z p(X, Z|\Theta) \frac{q(Z)}{q(Z)} \right)$$
$$= \log \left(\mathbb{E}\left[\frac{p(X, Z|\Theta)}{q(Z)} \right]_{q(Z)} \right)$$

where we multiply the equation by $\frac{q(Z)}{q(Z)}$ at the third line. Here, $q(Z)$ is an arbitrary function of random variables Z.

Recall that the Jensen's inequality states that $f(\mathbb{E}[x]) \geq \mathbb{E}[f(x)]$ for a concave function $f(x)$. As $\log(x)$ is a concave function, we can derive the following thanks to the Jensen's inequality:

$$\log p(X|\Theta) = \log \left(\mathbb{E}\left[\frac{p(X, Z|\Theta)}{q(Z)} \right]_{q(Z)} \right)$$
$$\geq \mathbb{E}\left[\log \left(\frac{p(X, Z|\Theta)}{q(Z)} \right) \right]_{q(Z)}$$
$$= \mathbb{E}\left[\log \left(p(X, Z|\Theta) \right) \right]_{q(Z)} - \mathbb{E}\left[\log \left(q(Z) \right) \right]_{q(Z)}$$

where the last term is the lower bound that EM maximizes at each iteration. The first expectation is called energy, which is the expected complete data log-likelihood. The second term is referred as entropy which is independent of Θ. Hence, maximizing the lower bound is equivalent to maximizing expectation of complete data log-likelihood w.r.t. $q(Z)$.

In classical EM derivations, the arbitrary function $q(Z)$ is chosen as $p(Z|X, \Theta)$ since it makes the bound tight:

$$
\begin{aligned}
\mathbb{E}\left[\log\left(\frac{p(X, Z|\Theta)}{q(Z)}\right)\right]_{q(Z)} &= \mathbb{E}\left[\log\left(\frac{p(X, Z|\Theta)}{p(Z|X, \Theta)}\right)\right]_{p(Z|X, \Theta)} \\
&= \mathbb{E}\left[\log\left(\frac{p(X, Z|\Theta)}{\frac{p(Z, X|\Theta)}{p(X|\Theta)}}\right)\right]_{p(Z|X, \Theta)} \\
&= \mathbb{E}\left[\log\left(p(X|\Theta)\right)\right]_{p(Z|X, \Theta)} \\
&= \sum_Z \log\left(p(X|\Theta)\right)p(Z|X, \Theta) \\
&= \log\left(p(X|\Theta)\right)
\end{aligned}
$$

The algorithm can be divided into 2 steps: Expectation (E) and Maximization (M). In E-Step, one needs to calculate expectation of complete data log-likelihood w.r.t. $p(Z|X, \Theta^{(t)})$ using previous estimates of unknown parameters. Then, unknown parameters are updated such that parameter values maximizing the expectation becomes new estimations. Hence, the iterative procedure can be written as follows:

- Expectation Step (E-step)

$$
Q(\Theta|\Theta^{(t)}) = \mathbb{E}\left[\log p(X, Z|\Theta)\right]_{p(Z|X, \Theta^{(t)})}
$$

- Maximization Step (M-step)

$$
\Theta^{(t+1)} = \underset{\Theta}{\operatorname{argmax}}\, Q(\Theta|\Theta^{(t)})
$$

where $\Theta^{(t)}$ denotes estimates for the unknown parameter at iteration t.

In any step of EM algorithm, the log-likelihood never decreases. This can be showed by using the KL divergence, which is a distance measure between two proba-

bility distributions $p(x)$ and $q(x)$. We define the KL divergence as follows:

$$KL(p||q) = -\int p(x) \log \frac{q(x)}{p(x)} \, dx$$

where $KL(p||q)$ is always non-negative and equal to zero if and only if $p(x) = q(x)$.

The energy term of the lower bound can be rewritten as:

$$
\begin{aligned}
\mathbb{E}\left[\log \left(\frac{p(X, Z|\Theta)}{q(Z)} \right) \right]_{q(Z)} &= \sum_Z q(Z) \log \left(\frac{p(X, Z|\Theta)}{q(Z)} \right) \\
&= \sum_Z q(Z) \log \left(\frac{p(Z|X, \Theta) p(X|\Theta)}{q(Z)} \right) \\
&= \sum_Z q(Z) \log p(X|\Theta) + \sum_Z q(Z) \log \left(\frac{p(Z|X, \Theta)}{q(Z)} \right) \\
&= \log p(X|\Theta) - KL\big(q(Z)||p(Z|X, \Theta)\big)
\end{aligned}
$$

where the first term is the log-likelihood and the second term is the KL divergence. This means that, for fixed Θ, the lower bound is bounded above by the log-likelihood, and achieves that bound when KL divergence is equal to 0. Hence, after an E-step, the energy equals the log-likelihood. As M-step maximizes the energy w.r.t. Θ, EM never decreases the log-likelihood.

2.3. The Gibbs Sampler Algorithm

Markov chain Monte Carlo (MCMC) methods are powerful statistical techniques used to approximate intractable densities by a finite set of samples. Suppose we are given a target distribution $\pi(x)$. Given samples generated from the target distribution, we can represent the analytically inexpressible distribution as follows:

$$\tilde{\pi}(x) = \frac{1}{M} \sum_{i=1}^{M} \delta(x - x^i) \tag{2.2}$$

where x^i is sample at i^{th} iteration and $\delta(\cdot)$ is dirac delta function.

Equation 2.2 allows calculating the expectation of a function $f(x)$ under target distribution $\pi(x)$, which is not possible analytically. However, provided that the number of samples M is large enough, MCMC algorithms approximate true value of the expectation thanks to the strong and weak laws of large numbers:

$$\mathbb{E}\big[f(x)\big]_{\pi(x)} = \int f(x)\pi(x)\,\mathrm{d}x \tag{2.3}$$

$$\approx \frac{1}{M}\sum_{i=1}^{M} f(x^i) \tag{2.4}$$

where x^i is sample at i^{th} iteration. This method is called Monte-Carlo integration in the literature [35].

In order to calculate the integration given in 2.4, we need to be able to sample from the target distribution, which is not possible in some cases as the target distribution $\pi(x)$ can be non-standard. Detailed analysis of MCMC methods is not in the scope of this thesis; however more information can be found in the literature [36–39].

One well-known MCMC method is Gibbs sampling proposed by Geman and Geman in [40]. After making an analogy between images and statistical systems, they introduced Gibbs sampling as an image restoration method. An explanatory work on the convergence of Gibbs sampler with introductory examples are given by Casella and Geoerge in [41].

The idea in Gibbs sampling is to design an ergodic Markov chain whose stationary distribution is the target distribution $\pi(x)$. In order to define a Markov chain, one needs to specify an initial probability distribution $\pi^{(0)}(x)$ and a transition probability $T(x';x)$ [42]. The probability distribution of the state at the $(t+1)^{th}$ iteration of the chain is given as:

$$\pi^{(t+1)}(x') = \int_x \pi^{(t)}(x)T(x';x)\,\mathrm{d}x$$

Note that the chain must be ergodic and the desired distribution must be an invariant distribution of the chain [39]. Also, the transition probabilities must have the detailed balance property which implies invariance of $\pi(x)$ under the chain [39].

Even though sampling from $\pi(x)$ is not feasible in some cases, sampling from full conditional distributions may be possible. After some iterations, burn-in period, samples generated from full conditional distributions are treated as if they are generated from $\pi(x)$. This, in turn, allows calculating approximate estimates of statistics such as expectations.

Consider sampling from the the joint distribution $p(x, y_1, y_2, ..., y_D)$, which might be difficult to perform. Fortunately, Gibbs sampler provides an alternative procedure in which one samples from full conditional distribution of each random variable iteratively. Hence, sampling from $p(x)$ can be achieved, and mean of $p(x)$ can be calculated by averaging samples of x:

$$\mathbb{E}[x] \approx \frac{1}{M} \sum_{i=1}^{M} X^i$$

where X^i is sample from full conditional distribution $p(x|y_1, y_2, ..., y_D)$ at iteration i.

In this example, a possible sampling scheme for Gibbs sampler is given in Algorithm 2.3 which presents a recipe for calculating the expectation of x.

Initialize $X^0, Y_1^0, Y_2^0, ..., Y_D^0$ randomly

for $i = 1$ to M **do**

$\quad X^i \sim p(x^i | y_1 = Y_1^{i-1}, y_2 = Y_2^{i-1}, ..., y_D = Y_D^{i-1});$

$\quad Y_1^i \sim p(y_1^i | x = X^i, y_2 = Y_2^{i-1}, ..., y_D = Y_D^{i-1});$

$\quad Y_2^i \sim p(y_2^i | x = X^i, y_1 = Y_1^i, y_3 = Y_3^{i-1}, ..., y_D = Y_D^{i-1});$

$\quad ...$

$\quad Y_D^i \sim p(y_D^i | x = X^i, y_1 = Y_1^i, ..., y_{D-1} = Y_{D-1}^i);$

end for

$\mathbb{E}[x] \approx \frac{1}{M} \sum_{i=1}^{M} X^i;$

Figure 2.3. Pseudo-code of Gibbs sampler for a toy example.

3. PROPOSED MODEL

This chapter presents the Sum Conditioned Poisson Factorization (SCPF) model which is an extension of PF described in Chapter 2.

3.1. Sum Conditioned Poisson Factorization

The model consists of K PF models and an observed cardinality matrix N which contains the sum of observation matrices of PF models. A schematic description of SCPF is given in Figure 3.1 where each cubic plot of X_k, W_k and H_k represents a PF.
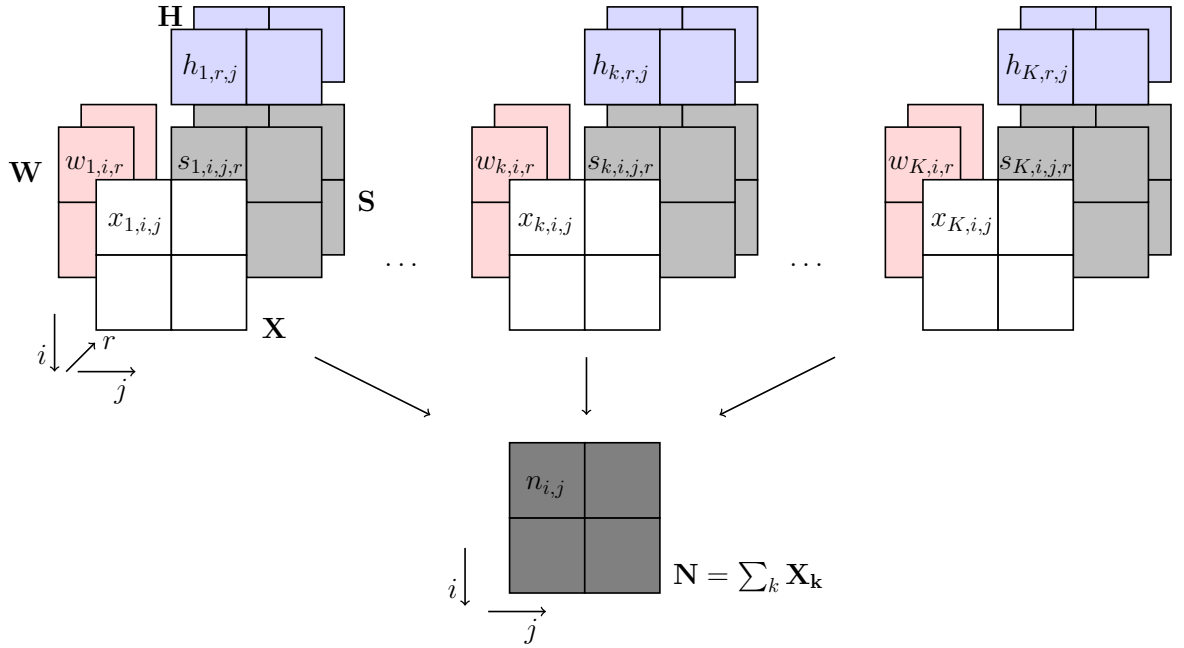
Figure 3.1. A schematic description of SCPF model.

In SCPF, each entry $x_{k,i,j}$ is still modeled as sum of Poisson random variables of $s_{k,i,j,1:R_k}$. However, conditioned on $n_{i,j}$, the random variables $x_{:,i,j}$ becomes coupled across the k variable thanks to the well known property of Poisson distributions [43]: conditioned on $n_{i,j}$, the joint posterior $p(x_{1,i,j}, \ldots, x_{K,i,j}|n_{i,j})$ is multinomial with the k'th cell probability given as μ_k/μ where μ_k is the intensity parameter of $x_{k,i,j}$.

The generative model of SCPF can be given as follows:

$$w_{k,i,r} \sim \mathcal{G}(w_{k,i,r}; a^w, b^w/a^w) \qquad\qquad h_{k,r,j} \sim \mathcal{G}(h_{k,r,j}; a^h, b^h/a^h)$$

$$s_{k,i,j,r} \sim \mathcal{PO}(s_{k,i,j,r}; w_{k,i,r} \times h_{k,r,j}) \qquad\qquad x_{k,i,j} = \sum_{r=1}^{R_k} s_{k,i,j,r}$$

$$n_{i,j} = \sum_{k=1}^{K} x_{k,i,j}$$

where a^w and b^w/a^w are the shape and scale parameters of Gamma distributions used for the variables in W while a^h and b^h/a^h are the shape and scale parameters of Gamma distributions used for the variables in H. Note that this parameter choice makes b^w and b^h mean parameters of Gamma distributions.

The cardinality matrix N is assumed to be always known such that $n_{i,j}$ is equal to the cardinality of the discrete variable $x_{:,i,j}$. Furthermore, for one-hot encoding schema, K is the number of categories. With setting N and K, one can model data with Bernoulli, Categorical, Binomial and Multinomial random variables.

Consider modeling a binary matrix. We set K to 2 and each entry $n_{i,j}$ to 1. The conditional distribution $p(x_{1,i,j}|n_{i,j}, w_{:,i,:}, h_{:,:,j})$ becomes a Bernoulli distribution such that:

$$p(x_{1,i,j}|n_{i,j}, w_{:,i,:}, h_{:,:,j}) = \mathcal{BE}(x_{1,i,j}; \frac{w_{1,i,:}h_{1,:,j}}{w_{1,i,:}h_{1,:,j} + w_{2,i,:}h_{2,:,j}})$$

Similarly, other distributions can also be derived. For a categorical distribution, we set K to the number of categories and let $n_{i,j} = 1$, leading to the following distribution:

$$p(x_{k,i,j}|n_{i,j}, w_{:,i,:}, h_{:,:,j}) = \text{Cat}(x_{k,i,j}; \frac{w_{k,i,:}h_{k,:,j}}{\sum_{k=1}^{K} w_{k,i,:}h_{k,:,j}})$$

For a binomial random variable with a range of $\{0, \ldots, n\}$, we set K to 2 and let $n_{i,j} = n$. For a multinomial distribution with a range of $\{0, \ldots, n\}$, we let $K > 2$ and $n_{i,j} = n$. Table 3.1 summarizes the parameter choices for modeling with Bernoulli, Categorical, Binomial and Multinomial random variables.

Table 3.1. Parameters for modeling multinomial data in SCPF.

	K	$n_{i,j}$
Bernoulli	2	1
Categorical	# categories	1
Binomial	2	>2
Multinomial	# categories	>2

For modeling missing data, we introduce a mask tensor M whose each entry $m_{k,i,j}$ becomes 1 or 0 if $x_{k,i,j}$ is observed or missing, respectively. This allows us to define a residual matrix $\tilde{N}$ where $\tilde{n}_{i,j} = n_{i,j} - \sum_{k=1}^{K} x_{k,i,j} m_{k,i,j}$. The unobserved entries in X models the residual matrix $\tilde{N}$ which can be seen as the rest of data.

Suppose we are given an ordinal data matrix Y such as movie ratings whose each entry takes values in $\{1, \ldots, n\}$. To model Y with SCPF, we can assign Y to $x_{1,:,:}$, first component of X, such that $x_{1,i,j} = Y_{i,j}$. In such a case, each entry of $m_{1,:,:}$, first component of mask tensor M, needs to be set to 1/0 for observed/missing entries. Also, let $K = 2$ and $n_{i,j} = n$ for all variables. For the special case of $K = 2$, $x_{2,:,:}$, second component of X, can also be inferred since $x_{2,i,j} = n - x_{1,i,j}$. Here, the matrices $x_{1,:,:}$ and $x_{2,:,:}$ can be interpreted as measures of likes and dislikes given by users to movies. Note that when there is no missing entries in the original matrix Y, the model would reduce to two independent PF models. However, missing entries makes corresponding entry $x_{1,i,k}$ a Binomial random variable whose predictive distribution can be given as:

$$p(x_{1,i,j} | n_{i,j}, w_{:,i,:}, h_{:,:,j}) = \mathcal{BI}(x_{1,i,j}; n_{i,j}, \frac{w_{1,i,:} h_{1,:,j}}{w_{1,i,:} h_{1,:,j} + w_{2,i,:} h_{2,:,j}})$$

We can also observe that, in this ordinal data case, the choice of $K > 2$ prevents us from inferring other variables $x_{2:K,i,j}$ even if $x_{1,i,k}$ is observed.

4. INFERENCE

For inference, we have investigated two methods: Expectation-Maximization (EM) and Gibbs sampler. Initially, EM is described for fully observed data where no missing entries exist so that the analogies to PF model can be easily established. We then adopt the inference for missing entries in which the novelty of our model comes into the play. Furthermore, we give an efficient implementation of the algorithm. In the second section, an MCMC method named Gibbs sampler, is derived and compared with EM in the Chapter 5.

4.1. The Expectation-Maximization Algorithm

In this section, we present important aspects of the EM derivations for inference in SCPF. However, interested reader can find more details in Appendix A. In SCPF, the observed variables are denoted by X and N. The latent variables of the model is represented by S. The unknown parameters are shown as W and H. Hence, the objective function of EM becomes the following:

$$Q(W, H | W^{(t)}, H^{(t)}) = \mathbb{E}\left[\log p(N, X, S | W, H) \right]_{p(S|N,X,W,H)} \tag{4.1}$$

where the posterior distribution of the latent variables can be calculated through:

$$p(S|N, X, W, H) = p(N, X, S | W, H) / p(N, X | W, H) \tag{4.2}$$

4.1.1. Derivations

In this section, we first derive EM equations for SCPF which are very similar to PF in cases where no missing data exists. Then, we present the derivations for missing data to show the difference between two models.

4.1.1.1. Fully Observed Data. The marginal log-likelihood, can be derived by marginalising out the latent variable S [4]:

$$
\begin{aligned}
\log p(N, X | W, H) &= \log \sum_S p(N, X, S | W, H) \\
&= \log \sum_S p(N|X) p(X|S) p(S|W, H) \\
&= \log \prod_{k,i,j} p(x_{k,i,j} | w_{k,i,:}, h_{k,:,j}) \\
&= \log \prod_{k,i,j} \mathcal{PO}(x_{k,i,j}; \sum_r w_{k,i,r}, h_{k,r,j}) \quad (4.3)
\end{aligned}
$$

which is a result of superposition property of Poisson random variables [43], meaning that sum of Poisson random variables s_1, s_2, ..., s_K with intensity parameters λ_1, λ_2, ..., λ_K is also a Poisson random variable whose intensity parameter is $\lambda = \sum_{k=1}^K \lambda_k$.

Note that at the second line the Equation 4.3, $p(n_{i,j} | x_{:,i,j})$ vanishes due to $n_{i,j} = \sum_{k=1}^K x_{k,i,j}$.

The posterior distribution of the latent variables consists of Multinomial distributions as shown in Appendix A.1:

$$
\log p(S | N, X, W, H) = \sum_{k,i,j} \log \mathcal{M}(s_{k,i,j,:}; x_{k,i,j}, p_{k,i,j,:})
$$

where the cell probability $p_{k,i,j,r}$ is equal to $\frac{w_{k,i,r} h_{k,r,j}}{\sum_r w_{k,i,r}, h_{k,r,j}}$. Since $s_{k,i,j,:}$ is shown to be a Multinomial random variable, the expectation of the latent variable $s_{k,i,j,r}$ can be easily calculated as:

$$
\mathbb{E}\big[s_{k,i,j,r} | w_{k,i,:}, h_{k,:,j}\big] = x_{k,i,j} p_{k,i,j,r} \quad (4.4)
$$

where expectation becomes a fraction of the observation $x_{k,i,j}$.

The objective function in Equation 4.1 can be written using Equation A.1:

$$
\mathbb{E}\Big[\log p(N, X, S|W, H)\Big]_{p(S|N,X,W,H)} = \sum_{k,i,j} \Bigg(\sum_{r} \Big(\mathbb{E}\big[s_{k,i,j,r}|w_{k,i,:}, h_{k,:,j}\big]
$$

$$
\log w_{k,i,r} h_{k,r,j} - w_{k,i,r} h_{k,r,j}
$$

$$
- \mathbb{E}\Big[\log \Gamma(s_{k,i,j,r} + 1)\Big]\Big)
$$

$$
+ \mathbb{E}\Big[\log \delta(x_{k,i,j} - \sum_{r} s_{k,i,j,r})\Big]
$$

$$
+ \log \delta(n_{i,j} - \sum_{k} x_{k,i,j}) \Bigg) \qquad (4.5)
$$

where the last three terms are merely constant for the maximization w.r.t. W and H.

Hence, one needs to maximize the following objective function:

$$
Q(W, H|W^{(t)}, H^{(t)}) = \sum_{k,i,j,r} \Bigg(\mathbb{E}\big[s_{k,i,j,r}^{(t)}|w_{k,i,:}^{(t)}, h_{k,:,j}^{(t)}\big] \log w_{k,i,r} h_{k,r,j} - w_{k,i,r} h_{k,r,j} \Bigg) \qquad (4.6)
$$

where the expectation of the latent variable $s_{k,i,j,r}^{(t)}$ is a result of Equation A.4:

$$
\mathbb{E}\big[s_{k,i,j,r}^{(t)}|w_{k,i,:}^{(t)}, h_{k,:,j}^{(t)}\big] = x_{k,i,j} p_{k,i,j,r}^{(t)}
$$

$$
= x_{k,i,j} \frac{w_{k,i,r}^{(t)} h_{k,r,j}^{(t)}}{\sum_{r} w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}}
$$

Finally, we present the fixed point equations for maximizing the objective function $Q(W, H|W^{(t)}, H^{(t)})$ whose details can be found in Appendix A.1:

$$
w_{k,i,r}^{(t+1)} = \frac{\sum_{j} \mathbb{E}\big[s_{k,i,j,r}^{(t)}|w_{k,i,:}^{(t)}, h_{k,:,j}^{(t)}\big]}{\sum_{j} h_{k,r,j}^{(t)}} = \frac{\sum_{j} x_{k,i,j} \frac{w_{k,i,r}^{(t)} h_{k,r,j}^{(t)}}{\sum_{r} w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}}}{\sum_{j} h_{k,r,j}^{(t)}}
$$

$$
= \frac{w_{k,i,r}^{(t)}}{\sum_{j} h_{k,r,j}^{(t)}} \sum_{j} x_{k,i,j} \frac{h_{k,r,j}^{(t)}}{\sum_{r} w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}} \qquad (4.7)
$$

$$h_{k,r,j}^{(t+1)} = \frac{\sum_i \mathbb{E}\big[s_{k,i,j,r}^{(t)}|w_{k,i,:}^{(t)}, h_{k,:,j}^{(t)}\big]}{\sum_i w_{k,i,r}^{(t)}} = \frac{\sum_i x_{k,i,j} \frac{w_{k,i,r}^{(t)} h_{k,r,j}^{(t)}}{\sum_r w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}}}{\sum_i w_{k,i,r}^{(t)}}$$

$$= \frac{h_{k,r,j}^{(t)}}{\sum_i w_{k,i,r}^{(t)}} \sum_i x_{k,i,j} \frac{w_{k,i,r}^{(t)}}{\sum_r w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}} \tag{4.8}$$

<u>4.1.1.2. Missing Data.</u> The derivations presented below follows a similar procedure used in the previous subsubsection. For handling missing data, we make use of Mask tensor M whose each entry $m_{k,i,j}$ becomes 1 or 0 when $x_{k,i,j}$ is observed or missing, respectively. Hence, the SCPF marginal log-likelihood given in Equation 4.3 becomes as follows:

$$\log p(N, X|W, H) = \log \sum_S p(N, X, S|W, H)$$

$$= \log \sum_S p(N|X)p(X|S)p(S|W, H)$$

$$= \log \left[\left(\prod_{k,i,j} p(x_{k,i,j}|w_{k,i,:}, h_{k,:,j})^{m_{k,i,j}} \right) \right.$$

$$\left. \left(\prod_{i,j} p(\tilde{n}_{i,j}|w_{:,i,:}, h_{:,:,j}, m_{:,i,j}) \right) \right]$$

$$= \sum_{k,i,j} m_{k,i,j} \log(\mathcal{PO}(x_{k,i,j}; \sum_r w_{k,i,r} h_{k,r,j}))$$

$$+ \sum_{i,j} \log(\mathcal{PO}(\tilde{n}_{i,j}; \sum_k (1 - m_{k,i,j}) \sum_r w_{k,i,r} h_{k,r,j}) \tag{4.9}$$

where $\tilde{n}_{i,j} = n_{i,j} - \sum_{k=1}^K x_{k,i,j} m_{k,i,j}$.

Accordingly, Equation A.2 changes as the calculation of the posterior distribution of the latent variables in S is updated:

$$\log p(S|N, X, W, H) = \log p(N, X, S|W, H) - \log p(N, X|W, H)$$

$$= \sum_{k,i,j} m_{k,i,j} \log \mathcal{M}(s_{k,i,j,:}; x_{k,i,j}, p_{k,i,j,:})$$

$$+ \sum_{i,j} \log \mathcal{M}(s_{:,i,j,:}; \tilde{n}_{i,j}, p_{:,i,j,:}, m_{:,i,j}) \qquad (4.10)$$

where $\tilde{n}_{i,j}$ is shared among the latent variables in $s_{:,i,j,:}$ for which $m_{k,i,j} = 0$.

Recall that when $x_{k,i,j}$ is not missing, i.e. $m_{k,i,j} = 1$, the latent random variables in $s_{k,i,j,:}$ were shown to be a Multinomial random variable so that the expectation of $s_{k,i,j,r}$ is calculated easily. However, when $x_{k,i,j}$ is missing, i.e. $m_{k,i,j} = 0$, the latent variable $s_{k,i,j,r}$ becomes coupled with others in $s_{:,i,j,:}$ for which $m_{k,i,j} = 0$. This, in turn, results in a Multinomial random variable as well which allows calculating the expectation of the latent variable $s_{k,i,j,r}$.

Hence, the expectation of the latent variable $s_{k,i,j,r}$ can be given as a piecewise function as follows:

$$\mathbb{E}\big[s_{k,i,j,r}|\tilde{n}_{i,j}, x_{k,i,j}, w_{k,i,1:R_k}, h_{k,1:R_k,j}\big] = \begin{cases} x_{k,i,j} p_{k,i,j,r}, & \text{if } m_{k,i,j} = 1, \\ \tilde{n}_{i,j} q_{k,i,j,r}, & \text{if } m_{k,i,j} = 0, \end{cases}$$

where $p_{k,i,j,r} = \frac{w_{k,i,r} h_{k,r,j}}{\sum_r w_{k,i,r}, h_{k,r,j}}$ and $q_{k,i,j,r} = \frac{w_{k,i,r} h_{k,r,j}}{\sum_k (1 - m_{k,i,j}) \sum_r w_{k,i,r}, h_{k,r,j}}$. This can be rewritten as:

$$\mathbb{E}\big[s_{k,i,j,r}^{(t)}|\tilde{n}_{i,j}, x_{k,i,j}, w_{k,i,:}^{(t)}, h_{k,:,j}^{(t)}\big] = \frac{m_{k,i,j} w_{k,i,r} h_{k,r,j} x_{k,i,j}}{\sum_r w_{k,i,r}, h_{k,r,j}} + \frac{(1 - m_{k,i,j}) w_{k,i,r} h_{k,r,j} \tilde{n}_{i,j}}{\sum_k (1 - m_{k,i,j}) \sum_r w_{k,i,r}, h_{k,r,j}}$$

One can easily derive the fixed point equations for maximizing the objective function $Q(W, H|W^{(t)}, H^{(t)})$ by using the expectation of the latent variable $s_{k,i,j,r}$ given

above:

$$w_{k,i,r}^{(t+1)} = \frac{\sum_j \mathbb{E}\big[s_{k,i,j,r}^{(t)}|\tilde{n}_{i,j}, x_{k,i,j}, w_{k,i,:}^{(t)}, h_{k,:,j}^{(t)}\big]}{\sum_j h_{k,r,j}^{(t)}}$$

$$= \frac{\sum_j \frac{m_{k,i,j} w_{k,i,r}^{(t)} h_{k,r,j}^{(t)} x_{k,i,j}}{\sum_r w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}} + \frac{(1-m_{k,i,j})w_{k,i,r}^{(t)} h_{k,r,j}^{(t)} \tilde{n}_{i,j}}{\sum_k (1-m_{k,i,j})\sum_r w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}}}{\sum_j h_{k,r,j}^{(t)}}$$

$$= \frac{w_{k,i,r}^{(t)}}{\sum_j h_{k,r,j}^{(t)}} \sum_j \frac{m_{k,i,j} h_{k,r,j}^{(t)} x_{k,i,j}}{\sum_r w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}} + \frac{(1-m_{k,i,j})h_{k,r,j}^{(t)} \tilde{n}_{i,j}}{\sum_k (1-m_{k,i,j})\sum_r w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}} \quad (4.11)$$

$$h_{k,r,j}^{(t+1)} = \frac{\sum_i \mathbb{E}\big[s_{k,i,j,r}^{(t)}|\tilde{n}_{i,j}, x_{k,i,j}, w_{k,i,:}^{(t)}, h_{k,:,j}^{(t)}\big]}{\sum_i w_{k,i,r}^{(t)}}$$

$$= \frac{\sum_i \frac{m_{k,i,j} w_{k,i,r}^{(t)} h_{k,r,j}^{(t)} x_{k,i,j}}{\sum_r w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}} + \frac{(1-m_{k,i,j})w_{k,i,r}^{(t)} h_{k,r,j}^{(t)} \tilde{n}_{i,j}}{\sum_k (1-m_{k,i,j})\sum_r w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}}}{\sum_i w_{k,i,r}^{(t)}}$$

$$= \frac{h_{k,r,j}^{(t)}}{\sum_i w_{k,i,r}^{(t)}} \sum_i \frac{m_{k,i,j} w_{k,i,r}^{(t)} x_{k,i,j}}{\sum_r w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}} + \frac{(1-m_{k,i,j})w_{k,i,r}^{(t)} \tilde{n}_{i,j}}{\sum_k (1-m_{k,i,j})\sum_r w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}} \quad (4.12)$$

Please observe that the update equations of SCPF and PF are exactly same when data is not missing, i.e. $m_{k,i,j} = 1$ as the second components vanishes. However, in the opposite case where data entries are missing, they differ because of the second component given in Table 4.2.

Table 4.1. Update rules for Poisson Factorization with observation, template and excitation matrices denoted by X, W and H, respectively.

Parameter	Update rule
$w_{i,r}$	$\frac{w_{i,r}^{(t)}}{\sum_j h_{r,j}^{(t)}} \sum_j \frac{h_{r,j}^{(t)} x_{i,j}}{\sum_r w_{i,r}^{(t)}, h_{r,j}^{(t)}}$
$h_{r,j}$	$\frac{h_{r,j}^{(t)}}{\sum_i w_{i,r}^{(t)}} \sum_i \frac{w_{i,r}^{(t)} x_{i,j}}{\sum_r w_{i,r}^{(t)}, h_{r,j}^{(t)}}$

Table 4.2. Update rules for Sum Conditioned Poisson Factorization with observation, template and excitation tensors denoted by X, W and H, respectively.

Parameter	Update rule
$w_{k,i,r}$	$\dfrac{w_{k,i,r}^{(t)}}{\sum_j h_{k,r,j}^{(t)}} \sum_j \left[\dfrac{m_{k,i,j} h_{k,r,j}^{(t)} x_{k,i,j}}{\sum_r w_{k,i,r}^{(t)} h_{k,r,j}^{(t)}} + \dfrac{(1-m_{k,i,j}) h_{k,r,j}^{(t)} \tilde{n}_{i,j}}{\sum_k (1-m_{k,i,j}) \sum_r w_{k,i,r}^{(t)} h_{k,r,j}^{(t)}} \right]$
$h_{k,r,j}$	$\dfrac{h_{k,r,j}^{(t)}}{\sum_i w_{k,i,r}^{(t)}} \sum_i \left[\dfrac{m_{k,i,j} w_{k,i,r}^{(t)} x_{k,i,j}}{\sum_r w_{k,i,r}^{(t)} h_{k,r,j}^{(t)}} + \dfrac{(1-m_{k,i,j}) w_{k,i,r}^{(t)} \tilde{n}_{i,j}}{\sum_k (1-m_{k,i,j}) \sum_r w_{k,i,r}^{(t)} h_{k,r,j}^{(t)}} \right]$

4.1.2. Implementation

We have derived EM update equations for inference in SCPF which might seem complicated to implement. Here, our goal is to present an efficient implementation of the algorithm by using matrix multiplications, similarly to [4].

Figure 4.1.2 allows us to update the unknown parameters of the model in a straightforward manner. We denote element-wise arithmetic operations of summation, multiplication and division with $.+$, $\odot$ and $\oslash$, respectively. $A * B$ is used for the inner product of two matrices A and B. $(A)^T$ symbolizes the transpose operation applied to a matrix A. Finally, 1 is a ones matrix of size $I \times J$.

4.2. Gibbs sampler

We have described a Gibbs sampler in Section 2.3. Here, we adopt the algorithm for inference in SCPF. Whilst other variants of Gibbs sampler could also be developed, we use a very simple sampling scheme. As the latent variables and unknown parameters in the model are W, H and S, samples are drawn from the following full conditional distributions at each iteration:

$$h_{k,r,j}^{(t)} \sim p(h_{k,r,j}|W = W^{(t-1)}, H_{-h_{k,r,j}} = H_{-h_{k,r,j}}^{(t-1)}, S = S^{(t-1)}, X = X^{(t-1)}, N = N)$$

Initialize the parameters: $W^{(0)}$ and $H^{(0)}$;

Set MX and $\hat{\tilde{N}}$ to zeros matrices of sizes $I \times J$;

for $k = 1$ to K **do**

   $MX \leftarrow MX \ .+ \ (M_k \odot X_k)$;

end for

$\tilde{N} \leftarrow N - MX$;

for $t = 1$ to T **do**

  for $k = 1$ to K **do**

    $\hat{X}_k \leftarrow W_k^{(t)} * H_k^{(t)}$;

    $\hat{\tilde{N}} \leftarrow \hat{\tilde{N}} \ .+ \ (1 - M_k) \odot \hat{X}_k$;

    $Q_k^x \leftarrow M_k \ \odot X_k \oslash \hat{X}_k$

    $Q_k^{xH} \leftarrow Q_k^x * (H_k^{(t)})^T$;

    $Q^{xW} \leftarrow (W_k^{(t)})^T * Q_k^x$;

  end for

  for $k = 1$ to K **do**

    $Q_k^n \leftarrow ((1 - M_k) \odot \tilde{N}) \oslash \hat{\tilde{N}}$;

    $Q_k^{nH} \leftarrow Q_k^n * (H_k^{(t)})^T$;

    $Q_k^{nW} \leftarrow (W_k^{(t)})^T * Q_k^n$;

    $1_k^W \leftarrow 1 * (H_k^{(t)})^T$;

    $1_k^H \leftarrow (W_k^{(t)})^T * 1$;

  end for

  $W^{(t+1)} \leftarrow (W^{(t)} \odot (Q^{xH} + Q^{nH})) \oslash 1^W$;

  $H^{(t+1)} \leftarrow (H^{(t)} \odot (Q^{xW} + Q^{nW})) \oslash 1^H$;

end for

Figure 4.1. Pseudo-code for Expectation-Maximization (EM) Algorithm.

$$w_{k,i,r}^{(t)} \sim p(w_{k,i,r}|W_{-w_{k,i,r}} = W_{-w_{k,i,r}}^{(t-1)}, H = H^{(t)}, S = S^{(t-1)}, X = X^{(t-1)}, N = N)$$

$$s_{k,i,j,r}^{(t)} \sim p(s_{k,i,j,r}|W = W^{(t)}, H = H^{(t)}, S_{-s_{k,i,j,r}} = S_{-s_{k,i,j,r}}^{(t-1)}, X = X^{(t-1)}, N = N)$$

At each iteration, we first sample from full conditional distribution of each entry of W. Then, sampling is done for variables in H. Finally, we draw samples for each variable in S.

4.2.1. Derivations

The complete derivations are presented in Appendix B. Here, we give sampling schemes for the latent variables. The samples for the latent variables in W and H are drawn from Gamma distributions whose parameters are updated with the previous values of corresponding parameters and observations. Note that the derivations appear very similar for W and H. However, the derivations of the full conditional distributions for the latent variables in S differ from others. For these Poisson random variables, we present a sampling procedure in which Multinomial distributions are used.

4.2.1.1. H Variables.

$$p(h_{k,r,j}^{(t)}|W^{(t-1)}, H_{-h_{k,r,j}}^{(t-1)}, S^{(t-1)}, X^{(t-1)}, N) \sim \mathcal{G}\left(h_{k,r,j}^{(t)}; a_{k,r,j}^h + \sum_{i=1}^{I} s_{k,i,j,r}^{(t-1)},\right.$$
$$\left.\left(a_{k,r,j}^h/b_{k,r,j}^h + \sum_{i=1}^{I} w_{k,i,r}^{(t-1)}\right)^{-1}\right)$$

4.2.1.2. *W* Variables.

$$p(w_{k,i,r}^{(t)}|W_{-w_{k,i,r}^{(t-1)}}^{(t-1)}, H^{(t)}, S^{(t-1)}, X^{(t-1)}, N) \sim \mathcal{G}\left(w_{k,i,r}^{(t)}; a_{k,i,r}^w + \sum_{j=1}^{J} s_{k,i,j,r}^{(t-1)}, \right.$$
$$\left.\left(a_{k,i,r}^w/b_{k,i,r}^w + \sum_{j=1}^{J} w_{k,i,r}^{(t)}\right)^{-1}\right)$$

4.2.1.3. *S* Variables.

Previously in Subsection 4.1.1.2, we have showed that the posterior distribution of the latent variables in S can be written as Multinomial distributions, giving the full conditional distributions, as well. Here, we omit the details of this derivation and follow the result presented in Equation A.13.

Note that parameters of Multinomial random variables depend on the mask tensor M as it denotes whether an observation variable $x_{k,i,j}$ is missing or not. In case of an observed variable $x_{k,i,j}$, i.e. $m_{k,i,j} = 1$, the variables $s_{k,i,j,:}$ can be sampled from a Multinomial distribution with the parameter vector $p_{k,i,j,:}$ where $p_{k,i,j,r} = \frac{w_{k,i,r}h_{k,r,j}}{\sum_r w_{k,i,r},h_{k,r,j}}$. Here, the observed value of distribution is given by $x_{k,i,j}$. For missing variables in $x_{:,i,j}$, we have a similar procedure in which observed value is replaced with $\tilde{n}_{i,j}$. This allows us to sample the latent variables in $s_{:,i,j,:}$ which correspond to the missing variables in $x_{:,i,j}$ from a Multinomial distribution, as well. However, the parameters need to be calculated differently. Entries of parameters, denoted by q, becomes as $q_{k,i,j,r} = \frac{w_{k,i,r}h_{k,r,j}}{\sum_k(1-m_{k,i,j})\sum_r w_{k,i,r},h_{k,r,j}}$.

Hence, if $x_{k,i,j}$ is observed, the following is used to sample:

$$s_{k,i,j,:}^{(t)} \sim \mathcal{M}(s_{k,i,j,:}^{(t)}; x_{k,i,j}^{(t)}, p_{k,i,j,:}^{(t)})$$

where $p_{k,i,j,r}^{(t)} = \frac{w_{k,i,r}^{(t)}h_{k,r,j}^{(t)}}{\sum_r w_{k,i,r}^{(t)},h_{k,r,j}^{(t)}}$.

For latent variables which correspond to the missing variables, one needs to use the following sampling scheme:

$$s_{:,i,j,:}^{(t)} \sim \mathcal{M}(s_{:,i,j,:}^{(t)}; \tilde{n}_{i,j}, q_{:,i,j,:}^{(t)})$$

where $q_{k,i,j,r}^{(t)} = \frac{w_{k,i,r}^{(t)} h_{k,r,j}^{(t)}}{\sum_k (1 - m_{k,i,j}) \sum_r w_{k,i,r}^{(t)} h_{k,r,j}^{(t)}}$.

The samples generated after burn-in period, $T_{burn-in}$, can be used for prediction based on the estimation of the latent variables:

$$\mathbb{E}\big[s_{k,i,j,r} | n_{i,j}, x_{k,i,j}, w_{k,i,r}, h_{k,r,j}\big] \approx \frac{1}{T - T_{burn-in}} \sum_{t=T_{burn-in}}^{T} s_{k,i,j,r}^{(t)}$$

where $s_{k,i,j,r}^{(t)}$ are samples at t^{th} iteration.

Hence, the observation variables can be estimated by normalizing the sum of expectations of the latent variables:

$$\mathbb{E}\big[x_{k,i,j} | n_{i,j}, s_{k,i,j,r}, w_{k,i,r}, h_{k,r,j}\big] = \frac{\sum_{r=1}^{R_k} \mathbb{E}\big[s_{k,i,j,r} | n_{i,j}, x_{k,i,j}, w_{k,i,r}, h_{k,r,j}\big]}{\sum_{k=1}^{K} \sum_{r=1}^{R_k} \mathbb{E}\big[s_{k,i,j,r} | n_{i,j}, x_{k,i,j}, w_{k,i,r}, h_{k,r,j}\big]}$$

5. EXPERIMENTS AND RESULTS

In this chapter, various experiments are conducted to compare the inference algorithms and show the properties of the SCPF model: interpretability and predictive performance. First experiment is desgined on a synthetic data set simply for comparing EM and Gibbs sampler. For interpretability, a binary data set is used and the model is compared with its canonical form alternative: Logistic Matrix Factorization. The predictive performance of the model is measured with an ordinal data set where the canonical form alternative, Ordinal Matrix Factorization, is also used.

The data sets, experiment setups and results are described in the following sections.

5.1. Simulated Data

In this experiment, our aim is to compare the performance of Gibbs sampler and EM algorithms for prediction of missing variables X. The synthetic data set used is generated from SCPF model with the parameters $K = 2$, $R = \{5, 5\}$, $N = 1$, $I = 9$, $J = 100$ meaning two components of X are formed with rank of 5. Also, the number of samples are 100 where each sample is a binary valued vector of length 9. Hence, the observations of the first component, X_1, can be shown as in Figure 5.1.
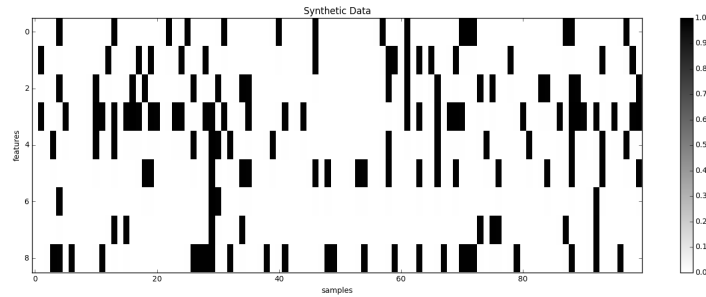

Figure 5.1. Binary valued data set generated from the model synthetically.

Since we have used $K = 2$ and $N = 1$, each entry of $x_{k,i,j}$ becomes a Bernoulli random variable. In EM, the expected value of $x_{k,i,j}$ is estimated as follows:

$$\mathbb{E}\big[x_{k,i,j}|n_{i,j}, s_{1:K,i,j,1:R_k}, w_{k,i,1:R_k}, h_{k,1:R_k,j}\big] = n_{i,j} \times$$
$$\frac{\sum_{r=1}^{R_k} \mathbb{E}\big[s_{k,i,j,r}|n_{i,j}, x_{k,i,j}, w_{k,i,r}, h_{k,r,j}\big]}{\sum_{k=1}^{K} \sum_{r=1}^{R_k} \mathbb{E}\big[s_{k,i,j,r}|n_{i,j}, x_{k,i,j}, w_{k,i,r}, h_{k,r,j}\big]}$$

where the expectation becomes a fraction of $n_{i,j}$.

In Gibbs sampler, we approximate to the expectation of $x_{k,i,j}$ by averaging the corresponding samples for the observation variable $x_{k,i,j}$:

$$\mathbb{E}\big[x_{k,i,j}|n_{i,j}, s_{k,i,j,1:R_k}, w_{k,i,1:R_k}, h_{k,1:R_k,j}\big] \approx \frac{1}{T - T_{burn-in}} \sum_{t=T_{burn-in}}^{T} x_{k,i,j}^{(t)}$$

Figure 5.2 denotes the data set used with X_1, the first component of the observation variables and the mask matrix M_1 whose each entry $m_{1,i,j}$ equals to 0 for missing variables and 1 for observed ones.
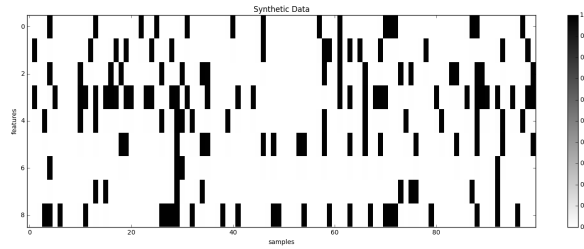
(a) Data matrix denoted by X_1.
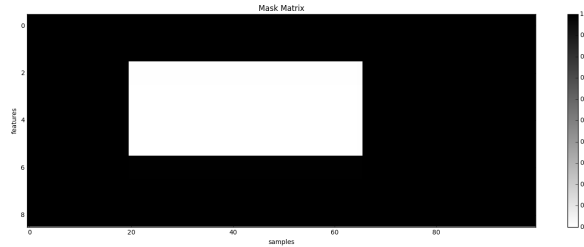
(b) Mask matrix denoted by M_1.

Figure 5.2. (a,b) Subfigures denote the data matrix X_1 and corresponding mask matrix M_1.

The expected value of each entry $x_{1,i,j}$ are plotted in the Figure 5.3 which shows that Gibbs sampler performs better than EM in restoration of missing variables. This suggests that EM might be over-fitting the data as it uses point estimates of the unknown parameters and latent variables. On the other hand, Gibbs sampler uses the full conditional distribution of the variables, which might be the reason of having more accurate estimations.
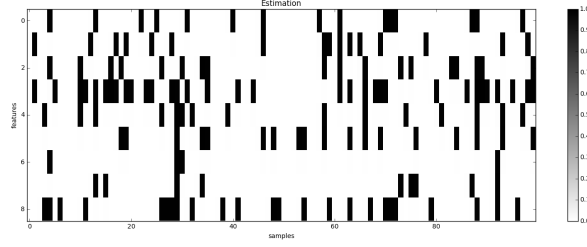
(a) Gibbs sampler.
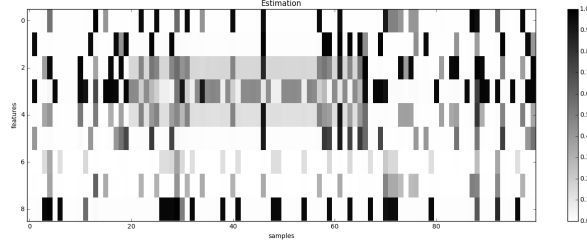
(b) EM.

Figure 5.3. (a,b) Subfigures represent the expected values for each entry of the data set given in Figure 5.2.

In order to show the performance of two algorithms for recovering the missing variables in X, we have plotted a randomly selected subset of the samples and corresponding expectations in Figure 5.4 and Figure 5.5, respectively.
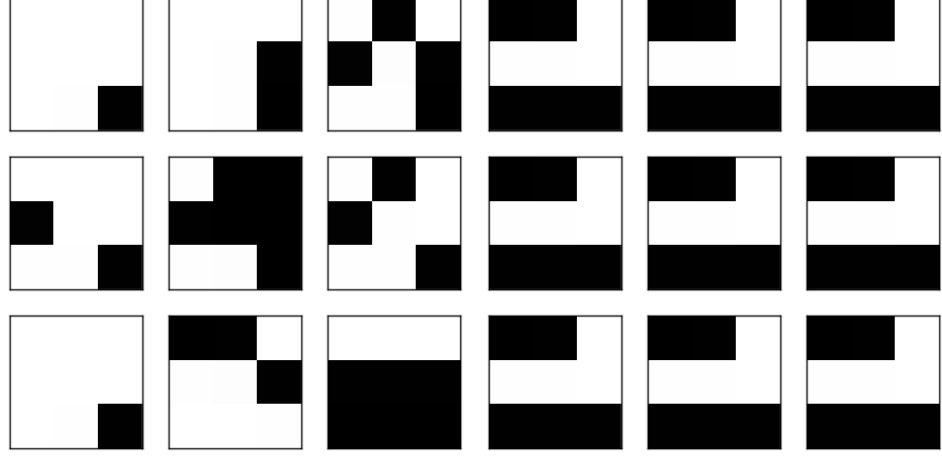
(a) A subset of data. (b) A subset of mask matrix.

Figure 5.4. (a) Each sample $x_{1:I,j}$ is reshaped to a square of size 3×3 and plotted in the given subplots. Corresponding mask matrix entries $m_{1:I,j}$ are also given in (b).

As can be seen from the Figure 5.5, Gibbs sampler provides closer estimates for the missing parts of the data given in the Figure 5.4. However, we believe that the full Bayesian treatment of the model would allow higher predictive performance in estimating the missing parts. Hence, an important future work is to adapt Gibbs sampler for estimating the hyperparameters of the model, as well.
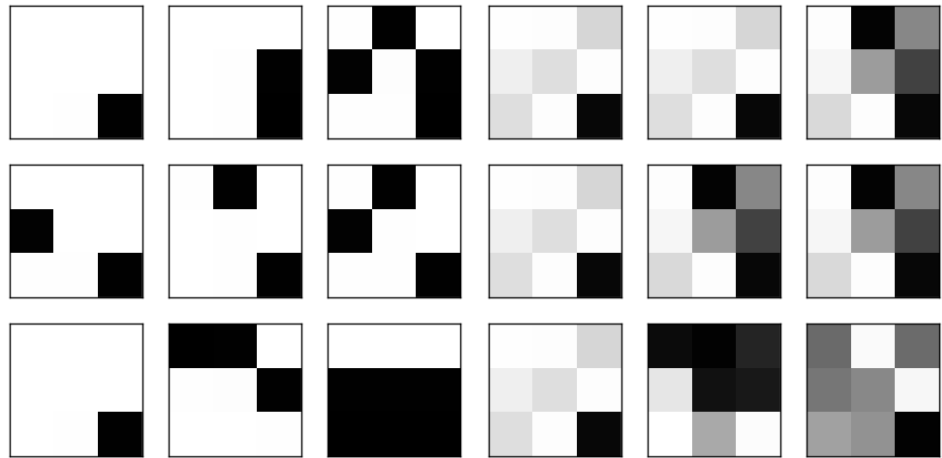
(a) Gibbs sampler. (b) EM.

Figure 5.5. Each image is the expected value of observation variables in X provided in the Figure 5.1.

5.2. Binary Data

As a binary data set, we used the Swimmer data set which is a collection of synthetically generated binary images. Each image consists of four "limbs" each of which can be in one of four positions. Figure 5.6 shows the data matrix, which is used as the observation matrix of the first component of X. In Figure 5.7, we see samples from the data set where each figure represents a reshaped version of a sample.
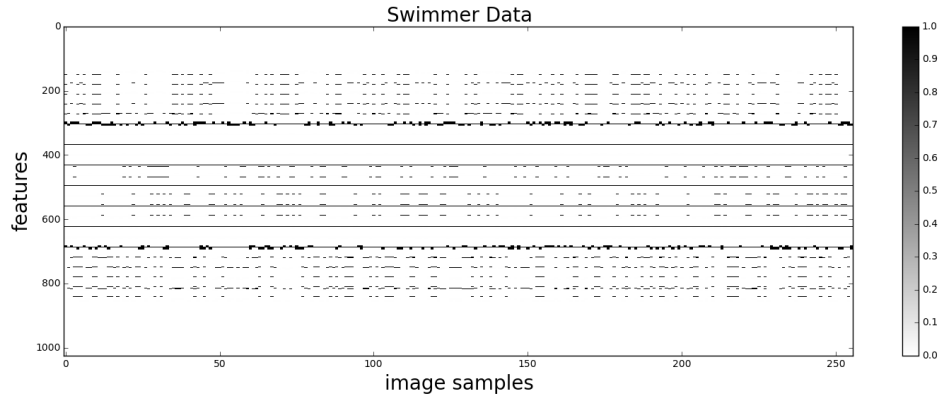
Figure 5.6. The figure shows the data matrix, which is used as the observation matrix of the first component of X.
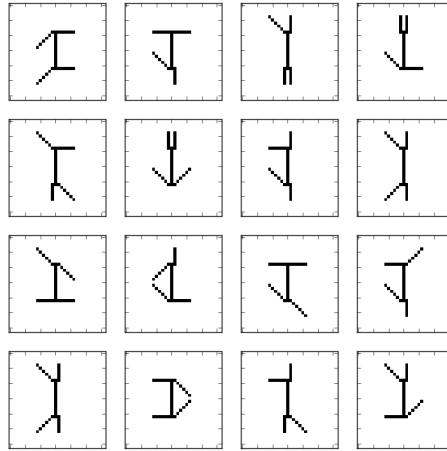
Figure 5.7. Each figure is a sample from the Swimmer data set.

We set the SCPF components as $K = 2$, $x_{1,i,j} = Y(i,j)$ and $x_{2,i,j} = 1 - x_{1,i,j}$ where Y denotes the data matrix given in Figure 5.6. As the data is binary valued, cardinality matrix N becomes an ones matrix whose each entry is 1. With a similar

notation, we can show that LMF models the data as follows:

$$Y(i,j) \sim \mathcal{BE}(\sigma(\sum_k W(i,k)H(k,j))$$

where $\sigma(\cdot)$ is the sigmoid function and $\mathcal{BE}$ is Bernoulli distribution.

Two alternatives are used to factorize the Swimmer data set. In order to analyze the models in terms of interpretability, we consider the template matrices inferred by LMF and the first component of SCPF, named X_1. Basis vectors in the template matrices are reshaped into 2D images and plotted in Figure 5.8. This brings us to the observation that the moment parametrization with SCFP allows a more interpretable template matrix. Similar to PF and NMF with multiplicative update rules, SCPF also learns by parts as each basis vector of the template matrix corresponds to a specific characteristics in the data.

(a) LMF.   (b) SCPF.
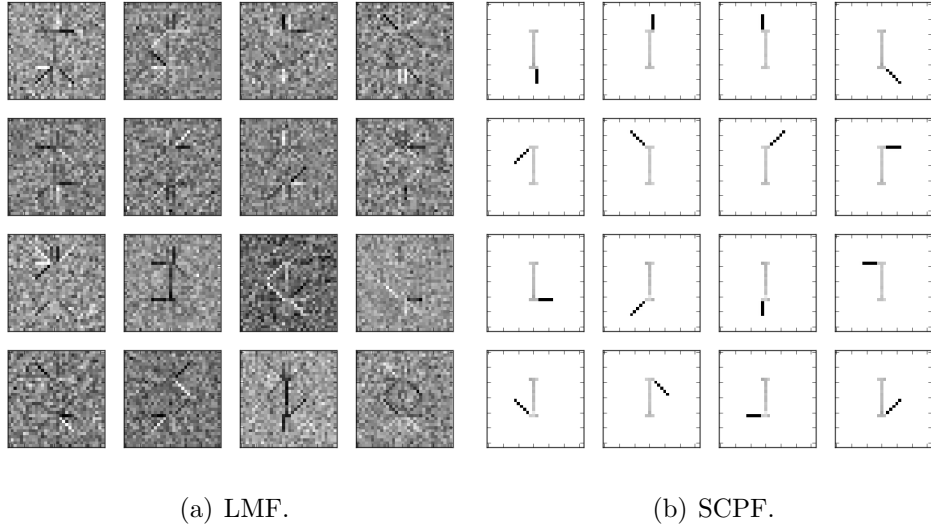
Figure 5.8. Each image is constructed by reshaping a basis vector in the template matrix (the first one in the SCPF case) inferred by the models.

5.3. Ordinal Data

For ordinal data analysis, we use the MovieLens 100-K data set consisting of 100-K ratings from 943 users on 1,682 movies. Each rating can take a value from 1 to

5 where 0 is used to show the absence of that rating.

The model can be used on ordinal data with the following setting: $x_{1,i,j} = Y(i,j)$ for observed $Y(i,j)$ and $n_{i,j} = 5$. We use the models with $K = 2$ and $K = 3$ to test the effect of component K.

The canonical form alternative of SCPF for ordinal data is Ordinal Matrix Factorization (OMF) which is constructed with probit function. OMF is described in more detail at [26]. We also compare the models with Gaussian [44] and Poisson Matrix Factorizations [4].

5.3.1. Experiment Setup

In experiments, 5-fold cross validation is used where one splits data set randomly into a training and a test set with 80% and 20% ratings. We repeat each experiment for the latent ranks $R \in \{20, 50, 100\}$. The shape A and mean parameter B of SCPF are fixed to 10^3 and $1/R$, respectively. The parameters of Ordinal Matrix Factorization (OMF) are kept as in the original paper [26] except that the latent rank is changed according to the experiment setting. We set the maximum iteration number to 1000 for each algorithm. Burn-in period of Gibbs sampler for OMF is given as 500.

We present the results with the maximum a-posteriori estimates for SCPF. Parameter estimation in LMF and Gaussian Matrix Factorization (GMF) is carried out through Stochastic Gradient Descent (SGD) with regularization. The Gibbs sampler provided for OMF is used in the experiments.

5.3.1.1. Metrics. We measure the performance of the models with Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and top-K recommendation for test ratings. Top-K recommendation performance is evaluated with the standard IR metrics

Mean Average Precision (MAP) and Recall@k, where

$$avg - precision_i = \sum_{j \in test_i} Precision(rank(i,j))/|test_i|$$

$$Recall@k_i = \sum_{j \in test_i} 1[rank(i,j) \leq k]/min(k,|test_i|)$$

Here, $test_i$ includes the movies that user i rated 5. and $rank(i,j)$ denotes the position of the item j in the recommendation list for user i.

5.3.2. Results

The results are presented in the Table 5.1 where we observe that SCPF gives a higher average precision, indicating a higher ranking performance of the model. This supports the idea of using the second component X_2 which represents user's' dislikes rather than using X_1 only as in PF.

Note that we calculate MAP through the ranked recommend list regardless of its length. But with recall@k, we also consider the length of recommendation lists. As can be seen from the Table 5.1, SCPF results in higher values of recall@k in short lists of length 10 and 20 only. Fortunately, this might be more useful for users since they are more likely to observe only a short list.

Setting $K > 2$ does not bring an additional benefit in our experiments. In SCPF, we factorize $\tilde{N}$ with $X_{2:K}$, the components of X other than the first one. Hence, increasing K actually correspond to a higher-rank factorization of $\tilde{N}$, which might be resulting in over-fitting. The lower performance with $K = 3$, can be explained by this over-fit to training data sets. In order to find the correct number of factorizations K and their latent dimensions $R_{1:K}$, a model order selection procedure can be developed, possibly leading to a higher performance of the model.

Table 5.1. MovieLens 100-K experiment results. SCPF-K denotes the proposed
model with K components. R@k is abbreviation for Recall@k.

	Model	RMSE	MAE	MAP	R@10	R@20	R@50
R=20	SCPF-2	0.961	0.743*	0.083*	0.113	0.165*	0.240*
	SCPF-3	0.978	0.753	0.080	0.123*	0.150	0.219
	PF	1.330	0.957	0.019	0.009	0.018	0.045
	GMF	0.940*	0.744	0.071	0.098	0.132	0.239
	OMF	0.980	0.762	0.041	0.058	0.086	0.174
R=50	SCPF-2	0.973	0.750	0.086*	0.133*	0.175*	0.255
	SCPF-3	0.999	0.763	0.075	0.122	0.162	0.223
	PF	1.393	1.031	0.020	0.018	0.036	0.099
	GMF	0.926*	0.736*	0.076	0.112	0.165	0.285*
	OMF	0.986	0.766	0.043	0.060	0.092	0.175
R=100	SCPF-2	0.997	0.763	0.084*	0.133*	0.180*	0.265
	SCPF-3	1.016	0.771	0.074	0.121	0.167	0.266
	PF	1.325	1.011	0.031	0.040	0.067	0.150
	GMF	0.924*	0.734*	0.068	0.099	0.154	0.276*
	OMF	0.999	0.778	0.037	0.054	0.080	0.158

5.4. Piano Roll Data

The experiments here aim to show the usefulness of the model in modeling time
series. Recall that the template matrices were previously shown to be interpretable
in Section 5.2. Now, we show that they can also be used as parameters of prior
distributions in time series analysis. For this, we use a piano roll data, a binary valued
data which we have extracted from Bach Chorales. A piano roll data X represents a
music such that $x_{i,j}$ denotes whether i^{th} note is active at time index j or not.

The model is compared with PF in order to investigate the possible benefits of
using SCPF over PF. Consider the data presented in Figure 5.9 where x-axis and y-axis

denote time and notes, respectively.
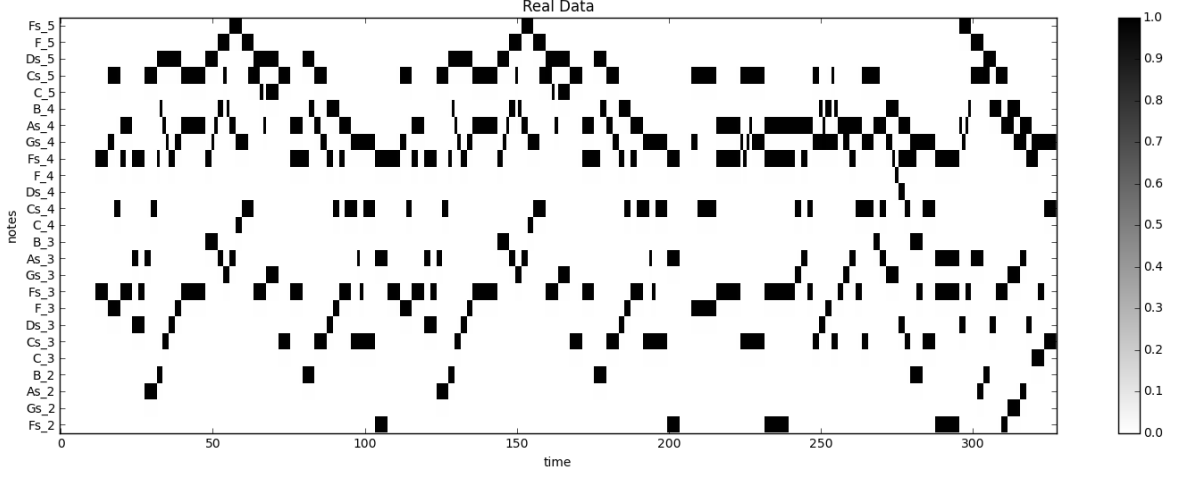
Figure 5.9. A piano roll data extracted from Bach Chorales.

For these experiments, we set the number of components in SCPF, K, to 2. Since the data is binary valued, N naturally becomes 1. I is set to 25 as piano roll data is acquired by ignoring the notes which are inactive during the whole play. Since the time step is taken as 1 sec., the length of data J becomes 328 in this example. Finally, rank is taken as each value in $\{10, 15\}$. Hence, the parameter settings for the experiments are as follows: $K = 2$, $N = 1$, $I = 25$, $J = 328$ and $R = \{10, 15\}$.

For inference, we have used EM algorithm whose convergences are shown in Figure 5.10 where log-likelihood of SCPF converges to a higher value.
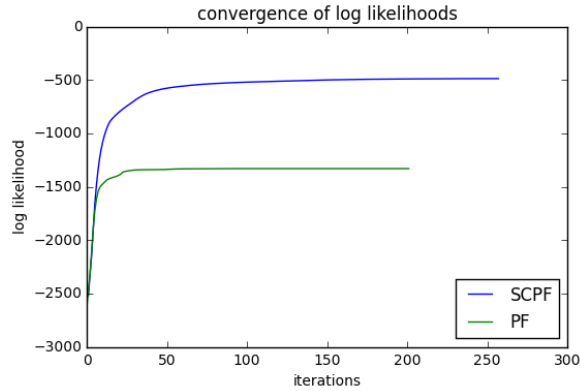
Figure 5.10. Convergence of log-likelihood of EM.

In Figure 5.11, we observe that both models, PF and SCPF, leads to estimations with similar patterns. However, SCPF restricts estimates to the interval of $[0, 1]$ while PF does not have such a constraint.
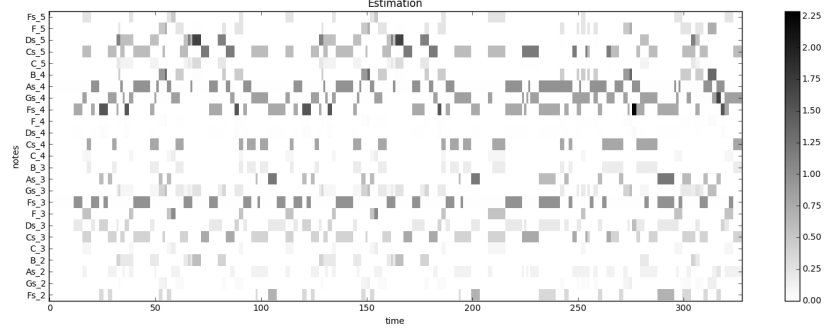
(a) Estimation for X by PF.
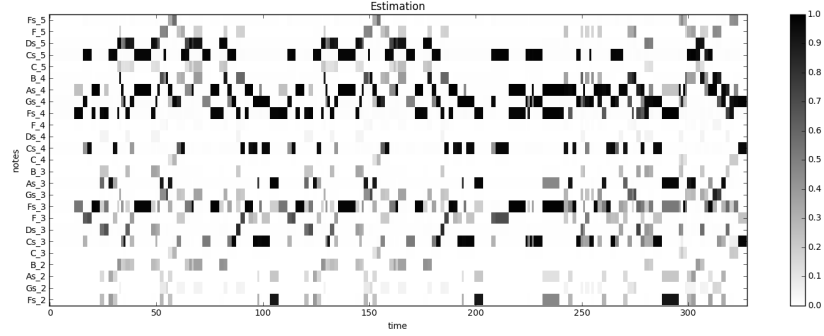
(b) Estimation for X_1 by SCPF.

Figure 5.11. (a,b) The model estimates where inference is carried out via EM algorithm.

The inferred basis vectors can be treated as prior parameters for modeling time series. Consider modeling data presented in Figure 5.9. The corresponding template matrices inferred by two models are given in Figure 5.12. A simple method can be modeling these observations as mixtures of the basis vectors in the template matrix given in Figure 5.12. Compared to the case where one starts from a randomly selected parameter value in state space, a faster convergence may be attained thanks to the prior beliefs extracted with the model.
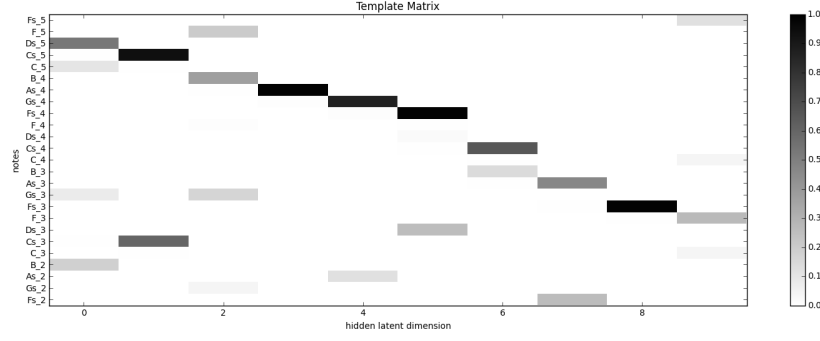
(a) Template Matrix by PF.
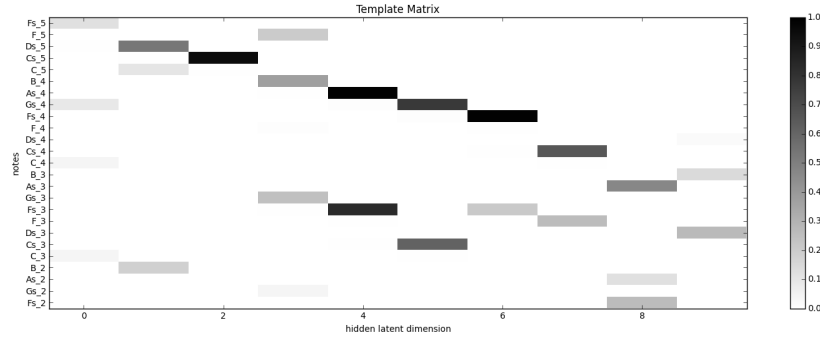
(b) Template Matrix by SCPF.

Figure 5.12. (a,b) Template matrices inferred by PF and SCPF with rank 10.

As can be seen from Figure 5.12, almost each basis vector contains 2 or 3 notes co-occurring some of which correspond to simple chord structures that can be found in the piano roll data. Note that this is not the case with PF.

In order to analyze the effect of the rank on expressiveness of basis vectors, we have also conducted experiments with $R = 15$ and $R = 5$. In this case, the basis vectors inferred are shown in Figure 5.13 and Figure 5.14.

We observe that as the number of latent dimensions increase the basis vectors correspond to smaller parts of the data. For $R = 15$, we see that the basis vectors mostly correspond to a single note. This is perhaps not surprising as the rank of the data is also not very high. For generalization to different data sets, the selection of R can be automatized by model order selection.

Table 5.2. Co-occurring notes found in each basis vector of the template matrix when rank is equal to 10.

Basis id	Notes
0	F5,B4,G#3
1	A#3,A#2,F#2
2	D#5,C5,B2
3	A#4,F#3
5	G#4,C#3
6	F#4,F#3
7	B3,D#3
8	C#4,F3

(a) Template Matrix by PF.
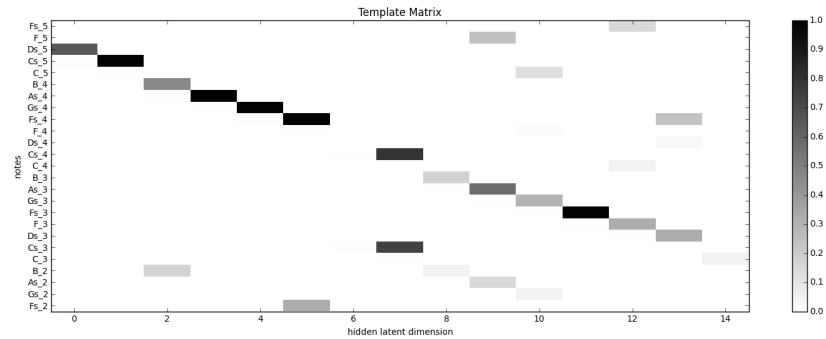
(b) Template Matrix by SCPF.

Figure 5.13. (a,b) Template matrices inferred by PF and SCPF with rank 15.

In Table 5.3, we have observed that each basis vector consists of 1 or 2 notes. This might be a result of using a higher rank for factorization.

Table 5.3. Co-occurring notes found in each basis vector of the template matrix when rank is equal to 15.

Basis id	Notes
0	B3
1	F#5
2	C#5
3	A#4
4	C#3
5	D#5,C5
6	G#3
7	F5,B4
8	F#3
9	C#4,F3
10	D#5,B2
11	F#4,D#3
12	A#3,A#2
13	G#4,F3
14	F#4,F#2

In contrast to $R = 15$, the choice of lower rank of $R = 5$ result in the loss of discovery for chord structures as notes start to mix in the basis vectors of the template matrix.

To sum up, the experiments with varying ranks suggest that there is an optimum value for the model to find the chord structures in the data. As PF did not infer chord structures in our experiments, the proposed model can be a good choice for extracting these musical structures, which might be useful in modeling music.

(a) Template Matrix by PF.

(b) Template Matrix by SCPF.

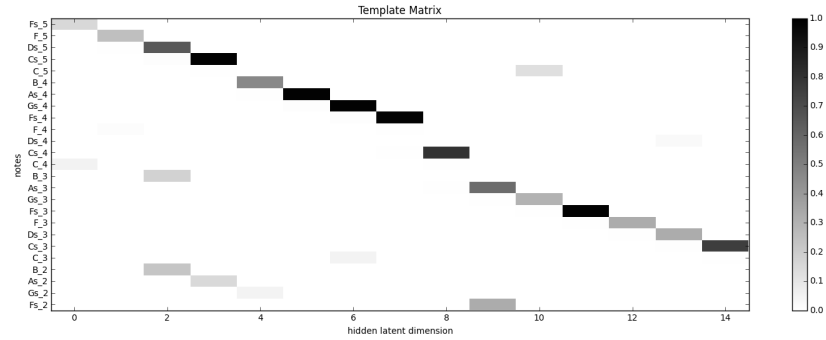Figure 5.14. (a,b) Template matrices inferred by PF and SCPF with rank 5.

Table 5.4. Co-occurring notes found in each basis vector of the template matrix when rank is equal to 5.

Basis id	Notes
0	F5,F#4,A#3,F#2
1	C#5,A#4,F#3
2	C#5,G#4,C#4,C#3
3	D#5,F#4,B3,D#3,B2
4	D#5,B4,G#3,F3

6. CONCLUSION

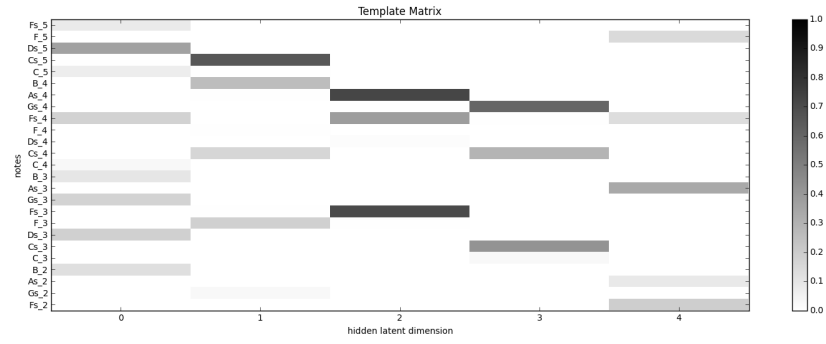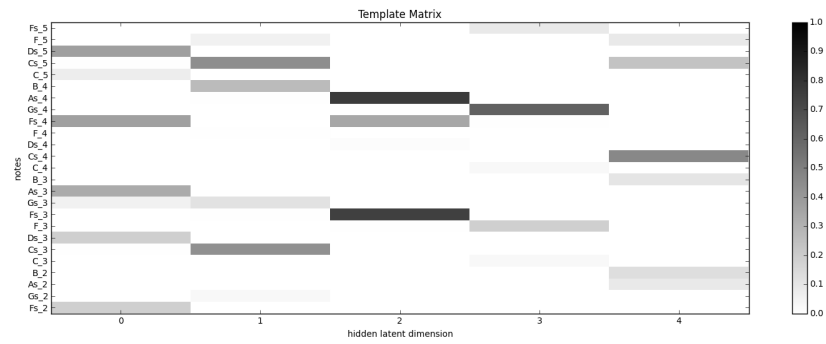In this work, we propose and investigate a model called Sum Conditioned Poisson Factorization for modeling bounded data such as Bernoulli, Binomial, Categorical and Multinomial. The model extends Poisson Factorization by conditioning multiple Poisson Factorization models on their sum. This allows to overcome the problem of modeling observations with Poisson distribution whose support is unbounded.

While the proposed model still enjoys the interpretability property of Poisson Factorization, it performs better in prediction tasks. We have shown interpretability of the model on a binary data set, Swimmer data set, such that basis vectors in inferred template matrix becomes explanatory. For testing its predictive performance, we have conducted experiments with Movie-Lens 500-K, which have demonstrated promising results.

We have also derived two algorithms for inference in the model: Gibbs sampler and Expectation-Maximization. They are compared in a simple experiment on a data set generated from the model synthetically. The results showed that Gibbs sampler performs better than EM as the latter may be suffering from over-fitting due to the use of point estimate for the variables. On the other hand, Gibbs sampler uses the full conditional distributions of the variables, which might be the reason for giving more information about the variables.

In the last experiment, we have employed the model on a piano roll data extracted from Bach Chorales. The goal was to show the use of the model in finding prior parameters that can be used for modeling time series. We have also showed that the model allows discovering the chord structures in piano roll data thanks to the basis vectors in the template matrix. In this sense, the model is compared to Poisson Factorization, which did not provide similar chord structures. However, the optimal value for the rank of factorization in Sum Conditioned Poisson Factorization may depend on the data of interest.

The experiments have demonstrated promising results for the proposed model. Therefore, we believe that further studies on the inference and applications are crucial. In that sense, one future work is to investigate the full Bayesian treatment of the model with approximate methods such as Variational Bayes or Gibbs sampler, where the hyperparameters of the model are also inferred. Another step is to develop variants of the model, i.e. Coupled Sum Conditioned Poisson Factorization such that unobserved properties of the model may be revealed.

In order to explore the model and its properties, we believe that applications should be extended, as well. For instance, the model can be used with different data sets in order to show its interpretability property. Additionally, we think that it would be more convincing for demonstrating the performance of the model by extending the experiments for the Collaborative Filtering problem. In this sequel, one may want to employ the model on data sets such as Movie-Lens 1M, EachMovie, and NetFlix. However, these are typically large-scale data sets, for which more scalable inference algorithms need to be developed [31].

REFERENCES

1. Berry, M. W., M. Browne, A. N. Langville, V. P. Pauca and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization", *Computational statistics & data analysis*, Vol. 52, No. 1, pp. 155–173, 2007.

2. Kim, J. and H. Park, "Toward faster nonnegative matrix factorization: A new algorithm and comparisons", *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pp. 353–362, IEEE, 2008.

3. Lee, D. D. and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization", *Nature*, Vol. 401, No. 6755, pp. 788–791, 1999.

4. Cemgil, A. T., "Bayesian inference for nonnegative matrix factorisation models", *Computational Intelligence and Neuroscience*, 2009.

5. Shahnaz, F., M. W. Berry, V. P. Pauca and R. J. Plemmons, "Document clustering using nonnegative matrix factorization", *Information Processing & Management*, Vol. 42, No. 2, pp. 373–386, 2006.

6. Xu, W., X. Liu and Y. Gong, "Document clustering based on non-negative matrix factorization", *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 267–273, ACM, 2003.

7. Virtanen, T., A. T. Cemgil and S. Godsill, "Bayesian extensions to non-negative matrix factorisation for audio signal modelling", *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 1825–1828, IEEE, 2008.

8. Smaragdis, P. and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription", *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*, pp. 177–180, IEEE, 2003.

9. Virtanen, T., "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria", *IEEE transactions on audio, speech, and language processing*, Vol. 15, No. 3, pp. 1066–1074, 2007.

10. Mohammadiha, N., P. Smaragdis and A. Leijon, "Prediction based filtering and smoothing to exploit temporal dependencies in NMF", *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 873–877, IEEE, 2013.

11. Ozerov, A. and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation", *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 18, No. 3, pp. 550–563, 2010.

12. Schmidt, M. N., J. Larsen and F.-T. Hsiao, "Wind noise reduction using nonnegative sparse coding", *Machine Learning for Signal Processing, 2007 IEEE Workshop on*, pp. 431–436, IEEE, 2007.

13. Nikunen, J., T. Virtanen and M. Vilermo, "Multichannel audio upmixing based on non-negative tensor factorization representation", *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2011 IEEE Workshop on*, pp. 33–36, IEEE, 2011.

14. Cooper, M. and J. Foote, "Summarizing video using non-negative similarity matrix factorization", *Multimedia Signal Processing, 2002 IEEE Workshop on*, pp. 25–28, IEEE, 2002.

15. Essid, S. and C. Févotte, "Smooth nonnegative matrix factorization for unsupervised audiovisual document structuring", *IEEE Transactions on Multimedia*, Vol. 15, No. 2, pp. 415–425, 2013.

16. Bucak, S. S. and B. Gunsel, "Video content representation by incremental nonnegative matrix factorization", *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, Vol. 2, pp. II–113, IEEE, 2007.

17. Masurelle, A., S. Essid and G. Richard, "Gesture recognition using a NMF-based representation of motion-traces extracted from depth silhouettes", *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pp. 1275–1279, IEEE, 2014.

18. Brunet, J.-P., P. Tamayo, T. R. Golub and J. P. Mesirov, "Metagenes and molecular pattern discovery using matrix factorization", *Proceedings of the national academy of sciences*, Vol. 101, No. 12, pp. 4164–4169, 2004.

19. Carmona-Saez, P., R. D. Pascual-Marqui, F. Tirado, J. M. Carazo and A. Pascual-Montano, "Biclustering of gene expression data by non-smooth non-negative matrix factorization", *BMC bioinformatics*, Vol. 7, No. 1, p. 78, 2006.

20. Gao, Y. and G. Church, "Improving molecular cancer class discovery through sparse non-negative matrix factorization", *Bioinformatics*, Vol. 21, No. 21, pp. 3970–3975, 2005.

21. Greene, D., G. Cagney, N. Krogan and P. Cunningham, "Ensemble non-negative matrix factorization methods for clustering protein–protein interactions", *Bioinformatics*, Vol. 24, No. 15, pp. 1722–1728, 2008.

22. Wang, F., T. Li, X. Wang, S. Zhu and C. Ding, "Community discovery using nonnegative matrix factorization", *Data Mining and Knowledge Discovery*, Vol. 22, No. 3, pp. 493–521, 2011.

23. Yang, J. and J. Leskovec, "Overlapping community detection at scale: a nonnegative matrix factorization approach", *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 587–596, ACM, 2013.

24. Gu, Q., J. Zhou and C. Ding, "Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs", *Proceedings of the 2010 SIAM International Conference on Data Mining*, pp. 199–210, SIAM, 2010.

25. Zhang, S., W. Wang, J. Ford and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization", *Proceedings of the 2006 SIAM International Conference on Data Mining*, pp. 549–553, SIAM, 2006.

26. Paquet, U., B. Thomson and O. Winther, "A hierarchical model for ordinal matrix factorization", *Statistics and Computing*, Vol. 22, No. 4, pp. 945–957, 2012.

27. Houlsby, N., J. M. Hernández-Lobato and Z. Ghahramani, "Cold-start Active Learning with Robust Ordinal Matrix Factorization.", *ICML*, pp. 766–774, 2014.

28. Collins, M., S. Dasgupta and R. E. Schapire, "A generalization of principal component analysis to the exponential family", *Advances in Neural Information Processing Systems*, MIT Press, 2001.

29. Tomé, A., R. Schachtner, V. Vigneron, C. Puntonet and E. Lang, "A logistic non-negative matrix factorization approach to binary data sets", *Multidimensional Systems and Signal Processing*, Vol. 26, No. 1, pp. 125–143, 2015.

30. Paatero, P. and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values", *Environmetrics*, Vol. 5, No. 2, pp. 111–126, 1994.

31. Gopalan, P., J. M. Hofman and D. M. Blei, "Scalable recommendation with poisson factorization", *arXiv preprint arXiv:1311.1704*, 2013.

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

33. Charlin, L., R. Ranganath, J. McInerney and D. M. Blei, "Dynamic poisson factorization", *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 155–162, ACM, 2015.

34. 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)*, pp. 1–38, 1977.

35. Caflisch, R. E., "Monte carlo and quasi-monte carlo methods", *Acta numerica*, Vol. 7, pp. 1–49, 1998.

36. Liu, J. S., *Monte Carlo strategies in scientific computing*, Springer Science & Business Media, 2008.

37. Spall, J. C., "Estimation via markov chain monte carlo", *IEEE control systems*, Vol. 23, No. 2, pp. 34–45, 2003.

38. Gilks, W. R., S. Richardson and D. Spiegelhalter, *Markov chain Monte Carlo in practice*, CRC press, 1995.

39. MacKay, D. J., *Information theory, inference and learning algorithms*, Cambridge university press, 2003.

40. Geman, S. and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images", *IEEE Transactions on pattern analysis and machine intelligence*, Vol. PAMI-6, No. 6, pp. 721–741, 1984.

41. Casella, G. and E. I. George, "Explaining the Gibbs sampler", *The American Statistician*, Vol. 46, No. 3, pp. 167–174, 1992.

42. Yıldırım, S., *Bayesian Methods For Deconvolution of Sparse Processes*, Master's Thesis, Bogazici University, 2009.

43. Kingman, J. F. C., *Poisson Processes*, Oxford Science Publications, 1993.

44. Salakhutdinov, R. and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo", *Proceedings of the 25th international conference on Machine learning*, pp. 880–887, ACM, 2008.

APPENDIX A: EM DERIVATIONS

A.1. Fully Observed Data

In order to calculate the posterior distribution in Equation 4.2, we can rewrite the terms on the right side in log-domain. The first term, the SCPF log-likelihood function, is as follows:

$$
\begin{aligned}
\log p(N, X, S | W, H) &= \log \Big(p(N|X)p(X|S)p(S|W,H) \Big) \\
&= \sum_{k,i,j} \Bigg(\sum_r \log \mathcal{PO}(s_{k,i,j,r}; w_{k,i,r} h_{k,r,j}) \\
&\quad + \log \delta(x_{k,i,j} - \sum_r s_{k,i,j,r}) \\
&\quad + \log \delta(n_{i,j} - \sum_k x_{k,i,j}) \Bigg) \\
&= \sum_{k,i,j} \Bigg(\sum_r \Big(s_{k,i,j,r} \log w_{k,i,r} h_{k,r,j} - w_{k,i,r} h_{k,r,j} \\
&\quad - \log \Gamma(s_{k,i,j,r} + 1) \Big) + \log \delta(x_{k,i,j} - \sum_r s_{k,i,j,r}) \\
&\quad + \log \delta(n_{i,j} - \sum_k x_{k,i,j}) \Bigg)
\end{aligned}
\tag{A.1}
$$

Combining Equations 4.2, A.1 and 4.3 leads to the following:

$$\log p(S|N, X, W, H) = \log p(N, X, S|W, H) - \log p(N, X|W, H)$$

$$= \sum_{k,i,j} \left(\sum_r \log \mathcal{PO}(s_{k,i,j,r}; w_{k,i,r} h_{k,r,j}) \right.$$

$$+ \log \delta(x_{k,i,j} - \sum_r s_{k,i,j,r})$$

$$\left. + \log \delta(n_{i,j} - \sum_k x_{k,i,j}) \right)$$

$$- \sum_{k,i,j} \log \mathcal{PO}(x_{k,i,j}; \sum_r w_{k,i,r}, h_{k,r,j})$$

$$= \sum_{k,i,j} \left(\sum_r \log \mathcal{PO}(s_{k,i,j,r}; w_{k,i,r} h_{k,r,j}) \right.$$

$$- \log \mathcal{PO}(x_{k,i,j}; \sum_r w_{k,i,r}, h_{k,r,j})$$

$$+ \log \delta(x_{k,i,j} - \sum_r s_{k,i,j,r})$$

$$\left. + \log \delta(n_{i,j} - \sum_k x_{k,i,j}) \right) \tag{A.2}$$

Poisson distributions in Equation A.2 can be rewritten in terms of their parameters:

$$\sum_r \log \mathcal{PO}(s_{k,i,j,r}; w_{k,i,r} h_{k,r,j}) - \log \mathcal{PO}(x_{k,i,j}; \sum_r w_{k,i,r}, h_{k,r,j})$$

$$= \left(\sum_r \left(s_{k,i,j,r} \log w_{k,i,r} h_{k,r,j} - w_{k,i,r} h_{k,r,j} - \log \Gamma(s_{k,i,j,r} + 1) \right) \right)$$

$$- \left(x_{k,i,j} \log(\sum_r w_{k,i,r}, h_{k,r,j}) - (\sum_r w_{k,i,r}, h_{k,r,j}) - \log \Gamma(x_{k,i,j} + 1) \right)$$

$$= \left(\sum_r \left(s_{k,i,j,r} \log w_{k,i,r} h_{k,r,j} - w_{k,i,r} h_{k,r,j} - \log \Gamma(s_{k,i,j,r} + 1) \right) \right)$$

$$- \left(\left(\sum_r s_{k,i,j,r} \right) \log \left(\sum_r w_{k,i,r}, h_{k,r,j} \right) - \left(\sum_r w_{k,i,r}, h_{k,r,j} \right) - \log \Gamma(x_{k,i,j} + 1) \right)$$

$$= \left(\sum_r s_{k,i,j,r} \log \frac{w_{k,i,r} h_{k,r,j}}{\sum_r w_{k,i,r}, h_{k,r,j}} - \log \Gamma(s_{k,i,j,r} + 1) \right) + \log \Gamma(x_{k,i,j} + 1) \qquad \text{(A.3)}$$

By incorporating the result of Equation A.3, we can observe that the posterior distribution of the latent variables consists of Multinomial distributions:

$$\log p(S|N, X, W, H) = \sum_{k,i,j} \left(\sum_r \log \mathcal{PO}(s_{k,i,j,r}; w_{k,i,r} h_{k,r,j}) \right.$$

$$- \log \mathcal{PO}(x_{k,i,j}; \sum_r w_{k,i,r}, h_{k,r,j})$$

$$+ \log \delta(x_{k,i,j} - \sum_r s_{k,i,j,r})$$

$$\left. + \log \delta(n_{i,j} - \sum_k x_{k,i,j}) \right)$$

$$= \sum_{k,i,j} \left(\sum_r s_{k,i,j,r} \log \frac{w_{k,i,r} h_{k,r,j}}{\sum_r w_{k,i,r}, h_{k,r,j}} \right.$$

$$\left. - \log \Gamma(s_{k,i,j,r} + 1) \right)$$

$$+ \log \Gamma(x_{k,i,j} + 1) + \log \delta(x_{k,i,j} - \sum_r s_{k,i,j,r})$$

$$+ \log \delta(n_{i,j} - \sum_k x_{k,i,j})$$

$$= \sum_{k,i,j} \log \mathcal{M}(s_{k,i,j,:}; x_{k,i,j}, p_{k,i,j,:})$$

where the cell probability $p_{k,i,j,r}$ is equal to $\frac{w_{k,i,r} h_{k,r,j}}{\sum_r w_{k,i,r}, h_{k,r,j}}$.

Since $s_{k,i,j,:}$ is shown to be a Multinomial random variable, the expectation of the latent variable $s_{k,i,j,r}$ can be easily calculated as:

$$\mathbb{E}\big[s_{k,i,j,r}|w_{k,i,:}, h_{k,:,j}\big] = x_{k,i,j} p_{k,i,j,r} \tag{A.4}$$

where expectation becomes a fraction of the observation $x_{k,i,j}$.

The objective function in Equation 4.1 can be written using Equation A.1:

$$\mathbb{E}\Big[\log p(N, X, S|W, H)\Big]_{p(S|N,X,W,H)} = \sum_{k,i,j} \Bigg(\sum_{r} \Big(\mathbb{E}\big[s_{k,i,j,r}|w_{k,i,:}, h_{k,:,j}\big]$$

$$\log w_{k,i,r} h_{k,r,j} - w_{k,i,r} h_{k,r,j}$$

$$- \mathbb{E}\Big[\log \Gamma(s_{k,i,j,r} + 1)\Big]\Big)$$

$$+ \mathbb{E}\Big[\log \delta(x_{k,i,j} - \sum_{r} s_{k,i,j,r})\Big]$$

$$+ \log \delta(n_{i,j} - \sum_{k} x_{k,i,j})\Bigg) \tag{A.5}$$

where the last three terms are merely constant for the maximization w.r.t. W and H.

Hence, one needs to maximize the following objective function:

$$Q(W, H|W^{(t)}, H^{(t)}) = \sum_{k,i,j,r} \Big(\mathbb{E}\big[s_{k,i,j,r}^{(t)}|w_{k,i,:}^{(t)}, h_{k,:,j}^{(t)}\big] \log w_{k,i,r} h_{k,r,j}$$

$$- w_{k,i,r} h_{k,r,j}\Big) \tag{A.6}$$

where the expectation of the latent variable $s_{k,i,j,r}^{(t)}$ is a result of Equation A.4:

$$\mathbb{E}\big[s_{k,i,j,r}^{(t)}|w_{k,i,:}^{(t)}, h_{k,:,j}^{(t)}\big] = x_{k,i,j} p_{k,i,j,r}^{(t)}$$

$$= x_{k,i,j} \frac{w_{k,i,r}^{(t)} h_{k,r,j}^{(t)}}{\sum_{r} w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}}$$

Finally, we conclude the fixed point equations for maximizing the objective function $Q(W, H|W^{(t)}, H^{(t)})$:

$$
\begin{aligned}
\frac{\partial Q(W, H|W^{(t)}, H^{(t)})}{\partial w_{k,i,r}} &= \sum_j \left(\mathbb{E}\big[s_{k,i,j,r}^{(t)}|w_{k,i,:}^{(t)}, h_{k,:,j}^{(t)}\big] \frac{1}{w_{k,i,r}} - h_{k,r,j}^{(t)} \right) \\
&= \left(\sum_j \mathbb{E}\big[s_{k,i,j,r}^{(t)}|w_{k,i,:}^{(t)}, h_{k,:,j}^{(t)}\big] \frac{1}{w_{k,i,r}} \right) - \left(\sum_j h_{k,r,j}^{(t)} \right) \\
&= 0
\end{aligned}
$$

$$
\begin{aligned}
w_{k,i,r}^{(t+1)} &= \frac{\sum_j \mathbb{E}\big[s_{k,i,j,r}^{(t)}|w_{k,i,:}^{(t)}, h_{k,:,j}^{(t)}\big]}{\sum_j h_{k,r,j}^{(t)}} = \frac{\sum_j x_{k,i,j} \frac{w_{k,i,r}^{(t)} h_{k,r,j}^{(t)}}{\sum_r w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}}}{\sum_j h_{k,r,j}^{(t)}} \quad\quad (A.7) \\
&= \frac{w_{k,i,r}^{(t)}}{\sum_j h_{k,r,j}^{(t)}} \sum_j x_{k,i,j} \frac{h_{k,r,j}^{(t)}}{\sum_r w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial Q(W, H|W^{(t)}, H^{(t)})}{\partial h_{k,r,j}} &= \sum_i \left(\mathbb{E}\big[s_{k,i,j,r}^{(t)}|w_{k,i,:}^{(t)}, h_{k,:,j}^{(t)}\big] \frac{1}{h_{k,r,j}} - w_{k,i,r}^{(t)} \right) \\
&= \left(\sum_i \mathbb{E}\big[s_{k,i,j,r}^{(t)}|w_{k,i,:}^{(t)}, h_{k,:,j}^{(t)}\big] \frac{1}{h_{k,r,j}} \right) - \left(\sum_i w_{k,i,r}^{(t)} \right) \\
&= 0
\end{aligned}
$$

$$
\begin{aligned}
h_{k,r,j}^{(t+1)} &= \frac{\sum_i \mathbb{E}\big[s_{k,i,j,r}^{(t)}|w_{k,i,:}^{(t)}, h_{k,:,j}^{(t)}\big]}{\sum_i w_{k,i,r}^{(t)}} = \frac{\sum_i x_{k,i,j} \frac{w_{k,i,r}^{(t)} h_{k,r,j}^{(t)}}{\sum_r w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}}}{\sum_i w_{k,i,r}^{(t)}} \\
&= \frac{h_{k,r,j}^{(t)}}{\sum_i w_{k,i,r}^{(t)}} \sum_i x_{k,i,j} \frac{w_{k,i,r}^{(t)}}{\sum_r w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}} \quad\quad (A.8)
\end{aligned}
$$

A.2. Missing Data

The derivations presented below follows a similar procedure used in the previous subsubsection. For handling missing data, we make use of Mask tensor M whose each entry $m_{k,i,j}$ becomes 1 or 0 when $x_{k,i,j}$ is observed or missing, respectively. Hence, the

SCPF marginal log-likelihood given in Equation 4.3 becomes as follows:

$$
\begin{aligned}
\log p(N, X|W, H) &= \log \sum_S p(N, X, S|W, H) \\
&= \log \sum_S p(N|X)p(X|S)p(S|W, H) \\
&= \log \left[\left(\prod_{k,i,j} p(x_{k,i,j}|w_{k,i,:}, h_{k,:,j})^{m_{k,i,j}} \right) \right. \\
&\qquad \left. \left(\prod_{i,j} p(\tilde{n}_{i,j}|w_{:,i,:}, h_{:,:,j}, m_{:,i,j}) \right) \right] \\
&= \sum_{k,i,j} m_{k,i,j} \log(\mathcal{PO}(x_{k,i,j}; \sum_r w_{k,i,r} h_{k,r,j})) \\
&\qquad + \sum_{i,j} \log(\mathcal{PO}(\tilde{n}_{i,j}; \sum_k (1 - m_{k,i,j}) \sum_r w_{k,i,r} h_{k,r,j}) \quad \text{(A.9)}
\end{aligned}
$$

where $\tilde{n}_{i,j} = n_{i,j} - \sum_{k=1}^{K} x_{k,i,j} m_{k,i,j}$.

Accordingly, Equation A.2 changes as the calculation of the posterior distribution of the latent variables in S is updated:

$$
\begin{aligned}
\log p(S|N, X, W, H) &= \log p(N, X, S|W, H) - \log p(N, X|W, H) \\
&= \sum_{k,i,j} \left(\sum_r \log \mathcal{PO}(s_{k,i,j,r}; w_{k,i,r} h_{k,r,j}) \right. \\
&\qquad + \log \delta(x_{k,i,j} - \sum_r s_{k,i,j,r}) \\
&\qquad \left. + \log \delta(n_{i,j} - \sum_k x_{k,i,j}) \right) \\
&\quad - \sum_{k,i,j} m_{k,i,j} \log(\mathcal{PO}(x_{k,i,j}; \sum_r w_{k,i,r} h_{k,r,j})) \\
&\quad - \sum_{i,j} \log(\mathcal{PO}(\tilde{n}_{i,j}; \sum_k (1 - m_{k,i,j}) \sum_r w_{k,i,r} h_{k,r,j})
\end{aligned}
$$

$$
\begin{aligned}
= \sum_{k,i,j} \Bigg(& \sum_r (m_{k,i,j} + 1 - m_{k,i,j}) \\
& \log \mathcal{PO}(s_{k,i,j,r}; w_{k,i,r} h_{k,r,j}) \\
& + \log \delta(x_{k,i,j} - \sum_r s_{k,i,j,r}) \\
& + \log \delta(n_{i,j} - \sum_k x_{k,i,j}) \Bigg) \\
- & \sum_{k,i,j} m_{k,i,j} \log(\mathcal{PO}(x_{k,i,j}; \sum_r w_{k,i,r} h_{k,r,j})) \\
- & \sum_{i,j} \log(\mathcal{PO}(\tilde{n}_{i,j}; \sum_k (1 - m_{k,i,j}) \sum_r w_{k,i,r} h_{k,r,j})) \\
= \sum_{k,i,j} \Bigg(& \sum_r m_{k,i,j} \log \mathcal{PO}(s_{k,i,j,r}; w_{k,i,r} h_{k,r,j}) \\
& + \sum_r (1 - m_{k,i,j}) \log \mathcal{PO}(s_{k,i,j,r}; w_{k,i,r} h_{k,r,j}) \\
& + \log \delta(x_{k,i,j} - \sum_r s_{k,i,j,r}) \\
& + \log \delta(n_{i,j} - \sum_k x_{k,i,j}) \Bigg) \\
- & \sum_{k,i,j} m_{k,i,j} \log(\mathcal{PO}(x_{k,i,j}; \sum_r w_{k,i,r} h_{k,r,j})) \\
- & \sum_{i,j} \log(\mathcal{PO}(\tilde{n}_{i,j}; \sum_k (1 - m_{k,i,j}) \sum_r w_{k,i,r} h_{k,r,j}))
\end{aligned}
\tag{A.10}
$$

$$
\begin{aligned}
= \sum_{k,i,j} \Bigg(& \sum_r m_{k,i,j} \log \mathcal{PO}(s_{k,i,j,r}; w_{k,i,r} h_{k,r,j}) \\
& - m_{k,i,j} \log(\mathcal{PO}(x_{k,i,j}; \sum_r w_{k,i,r} h_{k,r,j})) \\
& + \log \delta(x_{k,i,j} - \sum_r s_{k,i,j,r}) + \log \delta(n_{i,j} - \sum_k x_{k,i,j}) \Bigg) \\
+ \sum_{i,j} \Bigg(& \sum_{k,r} (1 - m_{k,i,j}) \log \mathcal{PO}(s_{k,i,j,r}; w_{k,i,r} h_{k,r,j}) \\
& - \left(\log(\mathcal{PO}(\tilde{n}_{i,j}; \sum_k (1 - m_{k,i,j}) \sum_r w_{k,i,r} h_{k,r,j})) \right) \Bigg)
\end{aligned}
\tag{A.11}
$$

Poisson distributions in Equation A.11 can be rewritten in terms of their parameters:

$$
\begin{aligned}
= \sum_{k,i,j} m_{k,i,j} & \left(\left(\sum_r s_{k,i,j,r} \log \frac{w_{k,i,r} h_{k,r,j}}{\sum_r w_{k,i,r}, h_{k,r,j}} - \log \Gamma(s_{k,i,j,r} + 1) \right) \right. \\
& \left. + \log \Gamma(x_{k,i,j} + 1) \right) + \log \delta(x_{k,i,j} - \sum_r s_{k,i,j,r}) \\
& + \log \delta(n_{i,j} - \sum_k x_{k,i,j}) \\
+ \sum_{i,j} & \left(\left(\sum_{k,r} (1 - m_{k,i,j}) s_{k,i,j,r} \log \frac{w_{k,i,r} h_{k,r,j}}{\sum_k (1 - m_{k,i,j}) \sum_r w_{k,i,r}, h_{k,r,j}} \right. \right. \\
& \left. \left. - (1 - m_{k,i,j}) \log \Gamma(s_{k,i,j,r} + 1) \right) + \log \Gamma(\tilde{n}_{i,j} + 1) \right)
\end{aligned}
\tag{A.12}
$$

$$
\begin{aligned}
= & \sum_{k,i,j} m_{k,i,j} \log \mathcal{M}(s_{k,i,j,:}; x_{k,i,j}, p_{k,i,j,:}) \\
& + \sum_{i,j} \log \mathcal{M}(s_{:,i,j,:}; \tilde{n}_{i,j}, p_{:,i,j,:}, m_{:,i,j})
\end{aligned}
\tag{A.13}
$$

where $\tilde{n}_{i,j}$ is shared among the latent variables in $s_{:,i,j,:}$ for which $m_{k,i,j} = 0$.

Recall that when $x_{k,i,j}$ is not missing, i.e. $m_{k,i,j} = 1$, the latent random variables in $s_{k,i,j,:}$ were shown to be a Multinomial random variable so that the expectation of

$s_{k,i,j,r}$ is calculated easily. However, when $x_{k,i,j}$ is missing, i.e. $m_{k,i,j} = 0$, the latent variable $s_{k,i,j,r}$ becomes coupled with others in $s_{:,i,j,:}$ for which $m_{k,i,j} = 0$. This, in turn, results in a Multinomial random variable as well which allows calculating the expectation of the latent variable $s_{k,i,j,r}$.

Hence, the expectation of the latent variable $s_{k,i,j,r}$ can be given as a piecewise function as follows:

$$\mathbb{E}\big[s_{k,i,j,r}\big] = \begin{cases} x_{k,i,j}p_{k,i,j,r}, & \text{if } m_{k,i,j} = 1, \\ \tilde{n}_{i,j}q_{k,i,j,r}, & \text{if } m_{k,i,j} = 0, \end{cases}$$

where $p_{k,i,j,r} = \frac{w_{k,i,r}h_{k,r,j}}{\sum_r w_{k,i,r},h_{k,r,j}}$ and $q_{k,i,j,r} = \frac{w_{k,i,r}h_{k,r,j}}{\sum_k (1-m_{k,i,j})\sum_r w_{k,i,r},h_{k,r,j}}$. This can be rewritten as:

$$\mathbb{E}\big[s^{(t)}_{k,i,j,r}|w^{(t)}_{k,i,:}, h^{(t)}_{k,:,j}\big] = \frac{m_{k,i,j}w_{k,i,r}h_{k,r,j}x_{k,i,j}}{\sum_r w_{k,i,r},h_{k,r,j}} + \frac{(1-m_{k,i,j})w_{k,i,r}h_{k,r,j}\tilde{n}_{i,j}}{\sum_k(1-m_{k,i,j})\sum_r w_{k,i,r},h_{k,r,j}}$$

One can easily derive the fixed point equations for maximizing the objective function $Q(W, H|W^{(t)}, H^{(t)})$ by using the expectation of the latent variable $s_{k,i,j,r}$ given above:

$$
\begin{aligned}
w^{(t+1)}_{k,i,r} &= \frac{\sum_j \mathbb{E}\big[s^{(t)}_{k,i,j,r}\big]}{\sum_j h^{(t)}_{k,r,j}} \\
&= \frac{\sum_j \frac{m_{k,i,j}w^{(t)}_{k,i,r}h^{(t)}_{k,r,j}x_{k,i,j}}{\sum_r w^{(t)}_{k,i,r},h^{(t)}_{k,r,j}} + \frac{(1-m_{k,i,j})w^{(t)}_{k,i,r}h^{(t)}_{k,r,j}\tilde{n}_{i,j}}{\sum_k(1-m_{k,i,j})\sum_r w^{(t)}_{k,i,r},h^{(t)}_{k,r,j}}}{\sum_j h^{(t)}_{k,r,j}} \\
&= \frac{w^{(t)}_{k,i,r}}{\sum_j h^{(t)}_{k,r,j}}\sum_j \frac{m_{k,i,j}h^{(t)}_{k,r,j}x_{k,i,j}}{\sum_r w^{(t)}_{k,i,r},h^{(t)}_{k,r,j}} + \frac{(1-m_{k,i,j})h^{(t)}_{k,r,j}\tilde{n}_{i,j}}{\sum_k(1-m_{k,i,j})\sum_r w^{(t)}_{k,i,r},h^{(t)}_{k,r,j}} \quad \text{(A.14)}
\end{aligned}
$$

$$h_{k,r,j}^{(t+1)} = \frac{\sum_i \mathbb{E}\left[s_{k,i,j,r}^{(t)}\right]}{\sum_i w_{k,i,r}^{(t)}}$$

$$= \frac{\sum_i \frac{m_{k,i,j} w_{k,i,r}^{(t)} h_{k,r,j}^{(t)} x_{k,i,j}}{\sum_r w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}} + \frac{(1-m_{k,i,j}) w_{k,i,r}^{(t)} h_{k,r,j}^{(t)} \tilde{n}_{i,j}}{\sum_k (1-m_{k,i,j}) \sum_r w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}}}{\sum_i w_{k,i,r}^{(t)}}$$

$$= \frac{h_{k,r,j}^{(t)}}{\sum_i w_{k,i,r}^{(t)}} \sum_i \frac{m_{k,i,j} w_{k,i,r}^{(t)} x_{k,i,j}}{\sum_r w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}} + \frac{(1-m_{k,i,j}) w_{k,i,r}^{(t)} \tilde{n}_{i,j}}{\sum_k (1-m_{k,i,j}) \sum_r w_{k,i,r}^{(t)}, h_{k,r,j}^{(t)}} \tag{A.15}$$

APPENDIX B: GIBBS SAMPLER

B.1. Update for $h_{k,r,j}^{(t)}$

$$p(h_{k,r,j}^{(t)}|W^{(t-1)}, H_{-h_{k,r,j}}^{(t-1)}, S^{(t-1)}, X^{(t-1)}, N) \propto p(h_{k,r,j}^{(t)}, W^{(t-1)}, H_{-h_{k,r,j}}^{(t-1)}, S^{(t-1)},$$

$$X^{(t-1)}, N)$$

$$\propto p(h_{k,r,j}^{(t)})p(s_{k,1:I,j,r}^{(t-1)}|h_{k,r,j}^{(t)}, w_{k,1:I,r}^{(t-1)})$$

$$= p(h_{k,r,j}^{(t)}) \prod_{i=1}^{I} p(s_{k,i,j,r}^{(t-1)}|w_{k,i,r}^{(t-1)}, h_{k,r,j}^{(t)})$$

Taking the logarithm leads to the following:

$$\log(p(h_{k,r,j}^{(t)}|W^{(t-1)}, H_{-h_{k,r,j}^{(t)}}^{(t-1)}, S^{(t-1)}, X^{(t-1)}, N)) \propto \log p(h_{k,r,j}^{(t)})$$

$$+ \sum_{i=1}^{I} \log p(s_{k,i,j,r}^{(t-1)}|w_{k,i,r}^{(t-1)}, h_{k,r,j}^{(t)})$$

We now continue by plugging the distributions into the equation:

$$.. = \log \mathcal{G}(h_{k,r,j}^{(t)}; a_{k,r,j}^h, b_{k,r,j}^h/a_{k,r,j}^h) + \sum_{i=1}^{I} \log \mathcal{PO}(s_{k,i,j,r}^{(t-1)}; w_{k,i,r}^{(t-1)} h_{k,r,j}^{(t)})$$

$$= (a_{k,r,j}^h - 1) \log h_{k,r,j}^{(t)} - h_{k,r,j}^{(t)}(a_{k,r,j}^h/b_{k,r,j}^h) - \log \Gamma(a_{k,r,j}^h)$$

$$+ a_{k,r,j}^h \log(b_{k,r,j}^h/a_{k,r,j}^h)$$

$$+ \left[\sum_{i=1}^{I} -(w_{k,i,r}^{(t-1)} h_{k,r,j}^{(t)}) + s_{k,i,j,r}^{(t-1)} \log(w_{k,i,r}^{(t-1)} h_{k,r,j}^{(t)}) - s_{k,i,j,r}^{(t-1)}! \right]$$

$$=^+ (a^h_{k,r,j} - 1) \log h^{(t)}_{k,r,j} - h^{(t)}_{k,r,j}(a^h_{k,r,j}/b^h_{k,r,j}) + \left[\sum_{i=1}^{I} -w^{(t-1)}_{k,i,r} h^{(t)}_{k,r,j} \right.$$

$$\left. + s^{(t-1)}_{k,i,j,r} \log h^{(t)}_{k,r,j} \right]$$

$$= (a^h_{k,r,j} + \sum_{i=1}^{I} s^{(t-1)}_{k,i,j,r} - 1) \log h^{(t)}_{k,r,j} - h^{(t)}_{k,r,j}(a^h_{k,r,j}/b^h_{k,r,j} + \sum_{i=1}^{I} w^{(t-1)}_{k,i,r})$$

$$\propto \log \mathcal{G}(h^{(t)}_{k,r,j}; a^h_{k,r,j} + \sum_{i=1}^{I} s^{(t-1)}_{k,i,j,r}, \left(a^h_{k,r,j}/b^h_{k,r,j} + \sum_{i=1}^{I} w^{(t-1)}_{k,i,r} \right)^{-1})$$

B.2. Update for $w^{(t)}_{k,i,r}$

$$p(w^{(t)}_{k,i,r}|W^{(t-1)}_{-w^{(t-1)}_{k,i,r}}, H^{(t)}, S^{(t-1)}, X^{(t-1)}, N) \propto p(w^{(t)}_{k,i,r}, W^{(t-1)}_{-w^{(t-1)}_{k,i,r}}, H^{(t)}, S^{(t-1)},$$

$$X^{(t-1)}, N)$$

$$\propto p(w^{(t)}_{k,i,r})p(s^{(t-1)}_{k,i,1:J,r}|w^{(t)}_{k,i,r} h^{(t)}_{k,r,1:J})$$

$$= p(w^{(t)}_{k,i,r}) \prod_{j=1}^{J} p(s^{(t-1)}_{k,i,j,r}|w^{(t)}_{k,i,r}, h^{(t)}_{k,r,1:J})$$

Similarly, we take the logarithm:

$$\log(p(w^{(t)}_{k,i,r}|W^{(t-1)}_{-w^{(t-1)}_{k,i,r}}, H^{(t)}, S^{(t-1)}, X^{(t-1)}, N)) \propto \log p(w^{(t)}_{k,i,r})$$

$$+ \sum_{j=1}^{J} \log p(s^{(t-1)}_{k,i,j,r}|w^{(t)}_{k,i,r}, h^{(t)}_{k,r,j})$$

We now continue by plugging the distributions into the equation:

$$
\begin{aligned}
.. &= \log \mathcal{G}(w_{k,i,r}^{(t)}; a_{k,i,r}^{w}, b_{k,i,r}^{w}/a_{k,i,r}^{w}) + \sum_{j=J}^{I} \log \mathcal{PO}(s_{k,i,j,r}^{(t-1)}; w_{k,i,r}^{(t)} h_{k,r,j}^{(t)}) \\
&= (a_{k,i,r}^{w} - 1)\log w_{k,i,r}^{(t)} - w_{k,i,r}^{(t)}(a_{k,i,r}^{w}/b_{k,i,r}^{w}) - \log \Gamma(a_{k,i,r}^{w}) \\
&\quad + a_{k,i,r}^{w}\log(b_{k,i,r}^{w}/a_{k,i,r}^{w}) \\
&\quad + \left[\sum_{j=1}^{J} -(w_{k,i,r}^{(t)} h_{k,r,j}^{(t)}) + s_{k,i,j,r}^{(t-1)}\log(w_{k,i,r}^{(t)} h_{k,r,j}^{(t)}) - s_{k,i,j,r}^{(t-1)}! \right] \\
&=^{+} (a_{k,i,r}^{w} - 1)\log w_{k,i,r}^{(t)} - h_{k,r,j}^{(t)}(a_{k,i,r}^{w}/b_{k,i,r}^{w}) + \left[\sum_{j=1}^{J} -w_{k,i,r}^{(t)} h_{k,r,j}^{(t)} \right. \\
&\quad \left. + s_{k,i,j,r}^{(t-1)}\log w_{k,i,r}^{(t)} \right] \\
&= (a_{k,i,r}^{w} + \sum_{j=1}^{J} s_{k,i,j,r}^{(t-1)} - 1)\log w_{k,i,r}^{(t)} - h_{k,r,j}^{(t)}(a_{k,i,r}^{w}/b_{k,i,r}^{w} + \sum_{j=1}^{J} w_{k,i,r}^{(t)}) \\
&\propto \log \mathcal{G}(w_{k,i,r}^{(t)}; a_{k,i,r}^{w} + \sum_{j=1}^{J} s_{k,i,j,r}^{(t-1)}, \left(a_{k,i,r}^{w}/b_{k,i,r}^{w} + \sum_{j=1}^{J} w_{k,i,r}^{(t)} \right)^{-1})
\end{aligned}
$$

B.3. Update for $s_{k,i,j,r}^{(t)}$

Previously in Subsection 4.1.1.2, we have showed that the posterior distribution of the latent variables in S can be written as Multinomial distributions, giving the full conditional distributions, as well. Here, we omit the details of this derivation and follow the result presented in Equation A.13.

Note that parameters of Multinomial random variables depend on the mask tensor M as it denotes whether an observation variable $x_{k,i,j}$ is missing or not. In case of an observed variable $x_{k,i,j}$, i.e. $m_{k,i,j} = 1$, the variables $s_{k,i,j,:}$ can be sampled from a Multinomial distribution with the parameter vector $p_{k,i,j,:}$ where $p_{k,i,j,r} = \frac{w_{k,i,r}h_{k,r,j}}{\sum_r w_{k,i,r,}h_{k,r,j}}$. Here, the observed value of distribution is given by $x_{k,i,j}$. For missing variables in $x_{:,i,j}$, we have a similar procedure in which observed value is replaced with $\tilde{n}_{i,j}$. This allows us to sample the latent variables in $s_{:,i,j,:}$ which correspond to the missing variables

in $x_{:,i,,j}$ from a Multinomial distribution, as well. However, the parameters need to be calculated differently. Entries of parameters, denoted by q, becomes as $q_{k,i,j,r} = \frac{w_{k,i,r}h_{k,r,j}}{\sum_k(1-m_{k,i,j})\sum_r w_{k,i,r,}h_{k,r,j}}$.

Hence, if $x_{k,i,j}$ is observed, the following is used to sample:

$$s_{k,i,j,:}^{(t)} \sim \mathcal{M}(s_{k,i,j,:}^{(t)}; x_{k,i,j}^{(t)}, p_{k,i,j,:}^{(t)})$$

where $p_{k,i,j,r}^{(t)} = \frac{w_{k,i,r}^{(t)}h_{k,r,j}^{(t)}}{\sum_r w_{k,i,r,}^{(t)}h_{k,r,j}^{(t)}}$.

For latent variables which correspond to the missing variables, one needs to use the following sampling scheme:

$$s_{:,i,j,:}^{(t)} \sim \mathcal{M}(s_{:,i,j,:}^{(t)}; \tilde{n}_{i,j}, q_{:,i,j,:}^{(t)})$$

where $q_{k,i,j,r}^{(t)} = \frac{w_{k,i,r}^{(t)}h_{k,r,j}^{(t)}}{\sum_k(1-m_{k,i,j})\sum_r w_{k,i,r,}^{(t)}h_{k,r,j}^{(t)}}$.