Dependent Indian Buffet Process-based Sparse Nonparametric Nonnegative Matrix Factorization

Junyu Xuan, Jie Lu, Senior Member, IEEE, Guangquan Zhang, Richard Yi Da Xu, Xiangfeng Luo, Member, IEEE

Abstract—Nonnegative Matrix Factorization (NMF) aims to factorize a matrix into two optimized nonnegative matrices appropriate for the intended applications. The method has been widely used for unsupervised learning tasks, including recommender systems (rating matrix of users by items) and document clustering (weighting matrix of papers by keywords). However, traditional NMF methods typically assume the number of latent factors (i.e., dimensionality of the loading matrices) to be fixed. This assumption makes them inflexible for many applications. In this paper, we propose a nonparametric NMF framework to mitigate this issue by using dependent Indian Buffet Processes (dIBP). In a nutshell, we apply a correlation function for the generation of two stick weights associated with each pair of columns of loading matrices, while still maintaining their respective marginal distribution specified by IBP. As a consequence, the generation of two loading matrices will be column-wise (indirectly) correlated. Under this same framework, two classes of correlation function are proposed (1) using Bivariate beta distribution and (2) using Copula function. Both methods allow us to adopt our work for various applications by flexibly choosing an appropriate parameter settings. Compared with the other state-of-the-art approaches in this area, such as using Gaussian Process (GP)-based dIBP, our work is seen to be much more flexible in terms of allowing the two corresponding binary matrix columns to have greater variations in their non-zero entries. Our experiments on the real-world and synthetic datasets show that three proposed models perform well on the document clustering task comparing standard NMF without predefining the dimension for the factor matrices, and the Bivariate beta distribution-based and Copula-based models have better flexibility than the GP-based model.

Index Terms—Nonnegative matrix factorization, Probability graphical model, Bayesian nonparametric learning, Indian Buffet Process

1 INTRODUCTION

ONNEGATIVE Matrix Factorization (NMF) is a renowned tool for factor analysis and unsupervised learning which has been used in many research areas [1]–[4]. In text mining area, for example, factorization on the document-keyword matrix can discover the hidden topics from documents, and these topics can assist document clustering or browsing; In the recommender system area, factorization on the user-movie matrix can help find the genres of users and movies. Based on these genres, the more accurate recommendation can be generated.

Sparse nonnegative matrix factorization [5] is favoured by researchers in several areas due to its output: sparse representation of data. Sparse representation discovers a limited number of components to represent data, which is an important research problem [6]. This sparse representation is generally desirable because it can aid human understanding (e.g., with gene expression data), reduce computational costs and obtain better generalization in learning algorithms. Sparsity is closely related to feature selection and certain generalizations in machine learning algorithms [7], [8].

Although NMF has experienced a boom, with significant development in recent years, the assumption that number of dimensions of factor matrices need to be predefined is blocking its usage in real-world applications. Normally, this number (i.e., topic number or genre number) is assigned by experts with domain knowledge, but it seems that inaccurate assignment will impact on the performance of NMF on such applied tasks as document clustering or recommender systems. Furthermore, increase of the amount of data and complexity of the tasks means that even experts are inadequate for this job. Therefore, it is more reasonable and practical to automatically discover the topic number from the data.

Stochastic processes, such as the Dirichlet process [9] and Indian Buffet Process (IBP) [10], [11], have been used to replace ordinary distributions in nonparametric learning [12] for providing priors of infinite vectors or matrices. In this context, IBP can
be seen as the prior of a sparse matrix with an infinite number of columns. Therefore, an intuitive idea would be to give two factor matrices two IBPs priors, thus avoiding the need to predefine the dimensions for the factor matrices. However, the problem is how to ensure the two IBPs will generate the same dimension number for the two factor matrices if they are given to them separately. A close related approach was seen to use GP-based dIBP [11]. This model works in the following fashion: We let $Z_j^{(1)}$ and $Z_j^{(2)}$ to be the $j$th corresponding columns of two identical sized binary matrices $Z^{(1)}$ and $Z^{(2)}$. The method assumes that both $Z_j^{(1)}$ and $Z_j^{(2)}$ share the same stick weight $\mu_j$, making the probability of generating the number of non-zero entries identical for both $Z_j^{(1)}$ and $Z_j^{(2)}$. The GP is merely used to control the correlations between individual entry pairs WITHIN $Z_j^{(1)}$ and $Z_j^{(2)}$, by thresholding a Gaussian CDF to make the entry to be either one or zero. Although this model improves upon the traditional NMF, it nonetheless is inadequate in many matrix factorisation scenarios as the assumption that the total number of non-zero entries are distributed identically does not hold universally. For example, the number of non-zero factors of a User column may be drastically different to the number of non-zero factors of a corresponding Item column in a recommender system.

In order to better and more flexibly describe the correlations between corresponding IBP columns, we instead propose a framework in which the correlation function is used for the generation of the latent weights pairs $\mu_j^{(1)}$ and $\mu_j^{(2)}$ which are responsible for the generation of $Z_j^{(1)}$ and $Z_j^{(2)}$ respectively. Compared with [11], our work allow both $Z_j^{(1)}$ and $Z_j^{(2)}$ to have much greater flexibility and variations in terms of their non-zero entries. To be specific, we propose two new dIBPs based on bi-variate beta distribution and copula. Instead of correlating two IBPs at the binary matrices level, the two proposed dIBPs correlate with the two IBPs at the very beginning (at the beta random variable level). This strategy results in the implementation of the nonparametric NMF framework being based on new dIBPs with a simple model structure. Another advantage of the new dIBPs is that the data correlation is directly modeled by the parameters of the bi-variate beta distribution and copula. Nevertheless, introducing bi-variate beta distribution and copula presents a challenge for the model inference. We have given three designed inference algorithms for three implementations of the nonparametric NMF framework: GP-based dIBP model, bi-variate beta distribution-based dIBP model, and copula-based dIBP model. The experiments show that the proposed models perform well on the document clustering task without explicitly predefining the dimension number for the factor matrices, and the models based on new dIBPs have better convergence rates than GP-based dIBP.

The contributions of this paper are:

- Two new dIBPs (i.e., bi-variate beta distribution-based and copula-based) with simpler model structure and direct correlation modeling capability are proposed as alternatives to the existing GP-based dIBP;
- Three dIBP-based nonparametric nonnegative matrix factorization models are proposed to remove the assumption of the traditional NMF that the number of factors needs to be fixed in advance.

The rest of this paper is organized as follows. Preliminary details of NMF and IBP are briefly introduced in Section 2. Section 3 reviews related work. Our dIBP-based nonparametric NMF framework is proposed with three implementations in Section 4, and Gibbs samplers are designed for the three models in Section 5. Section 6 conducts a set of experiments on real-world tasks. Lastly, Section 7 concludes the study and discusses further work.

2 Preliminary Knowledge

2.1 Sparse Nonnegative Matrix Factorization

Given a nonnegative matrix $Y_{m \times n}$ (extended to semi-nonnegative by [13]), Nonnegative Matrix Factorization (NMF) aims to find two matrices $A_{m \times k}$ and $X_{n \times k}$ to minimize the following cost function,

$$J = \|Y - AX\|_F^2 + \|A\|_1 + \|X\|_1$$

where $\|\cdot\|_F$ is the Frobenius norm, $\|\cdot\|_1$ is the $\ell_1$ norm and the elements of $A$ and $X$ are also nonnegative. The $\ell_1$ norm in the cost function is used for the sparseness constraint.

NMF is widely used in many different research areas. Take the recommender system as an example. The input $Y_{m \times n}$ denotes the ratings given by $m$ users on $n$ movies. The $A_{m \times k}$ denotes the users’ interests, and $X_{n \times k}$ denotes the movies’ genres. All users and movies are projected into the same $k$-dimensional space by NMF. Based on $A_{m \times k}$ and $A_{m \times k}$, a more accurate recommendation can be achieved.

One problem of NMF is to determine $k$. Normally, this variable is experimentally adjusted within a range. The problem of NMF without predefined $k$ is called Nonparametric Nonnegative Matrix Factorization.

2.2 Indian Buffet Process

The Indian Buffet Process (IBP) [10], [11] is defined as a prior for the binary matrices with an infinite number of columns. The graphical model is shown in Fig. 1(a). A stick-breaking construction for IBP [14] is proposed...
as follows:
\[
\nu_j \sim \text{Beta}(\alpha, 1), \quad \mu_k = \prod_{j=1}^{k} \nu_j, \quad z_{n,k} \sim \text{Bernoulli}(\mu_k)
\] (2)

where \(z_{n,k}\) form a matrix \(Z_{N \times K}\), \(\nu_j\) is the variable with a beta distribution, and \(\alpha\) is the parameter of beta distribution. \(\mu_k\) is the stick weight of column \(k\). The bigger \(\mu_k\) is, the more ‘ones’ there are in the column \(k\) of the binary matrix \(Z_{N \times K}\). The final number of \(K\) is determined by the data and the parameter \(\alpha\) of IBP.

An intuitive idea of applying IBP for Nonparametric Nonnegative Matrix Factorization is to separately set two IBPs as priors of \(A\) and \(X\). One problem with this idea is that the number of columns \(K_1\) and \(K_2\) generated by the two IBPs may be different, which violates the requirement of NMF.

3 RELATED WORK

The related work mainly falls into two categories: one concerns research on the Nonnegative Matrix Factorization and the other concerns about the Indian Buffet Process. These state-of-the-art researches inspire our idea of using a dependent Indian Buffet Process for nonparametric nonnegative matrix factorization.

3.1 Nonparametric Nonnegative Matrix Factorization

Types of research on the Nonnegative Matrix Factorization include supervised or semi-supervised extension [13], [15], the convergence rate [16], Sparse [17], Nonparametric [18] and more.

There are also some researches on Bayesian extension of NMF which aim to model the NMF using the distributions. In one example, LDA [19] and correlated LDA [20] ideas are transferred to the Bayesian parametric NMF and correlated NMF [21]. These extensions are still parametric, however, which means the dimension number still need to be predefined.

The nonparametric extension of NMF mainly relies on the machinery of stochastic processes. The Beta process is used as the prior of one factor matrix in [18]. The Gamma process is used to generate the coefficients for the combination of corresponding elements in factor matrices rather than the prior of the factor matrices [22]. Both have been successfully applied for music analysis. IBP is used as the prior of one factor matrix and another factor matrix is drawn from Gaussian distribution; an efficient inference method (Power-EP) is proposed for this model [23]. These processes can be considered as the extension of the latent feature factor model [11]. However, there is no work to place the priors for two factor matrices simultaneously.

3.2 IBP and dIBP

The idea of dependent nonparametric processes was first proposed by MacEachern [24]. Seven classes of dependence are summarized [25]. The basic IBP is proposed in [10], [11], and is actually a marginalization of the Beta-Bernoulli process. Its widespread popularity is due to its power to generate a binary matrix with infinite columns. The dIBP was first proposed in [26] based on Gaussian process, and can be used for the nonparametric NMF after being embedded in our proposed framework which is discussed in Section 4. The Kernel Beta process [27] is another dependent beta process model that models the dependence between data points with different covariants, such as the time tags of documents, geographic locations of people or GDPs of countries. Hierarchial beta process is proposed so that the different beta processes share a common base discrete measure [28]. The Phylogenetic Indian Buffet Process [29] considers the tree structure of the data points, which can be seen as a supervised IBP. A coupled IBP [30] is proposed for collaborative filtering, which links two IBPs through a factor matrix. Therefore, this coupling does not guarantee that the factor matrices have the same dimension number which is important for the NMF. dimension number of two infinite factor matrices . However, there is no work on using or constructing dIBP for nonparametric NMF.

4 NONPARAMETRIC NMF FRAMEWORK AND THREE IMPLEMENTATIONS

To set a prior for the infinite matrices \(A\) and \(X\) in NMF and make sure the two matrices have the same

| Symbol | meaning in this paper |
|--------|-----------------------|
| \(M\)  | the row number of \(Y\) |
| \(N\)  | the column number of \(Y\) |
| \(K\)  | the latent factor number |
| \(Y\)  | data matrix with size \(M \times N\) |
| \(y_{m,n}\) | the element of \(Y\) at \(m\) row and \(n\) column |
| \(A\)  | factor matrix with size \(M \times K\) |
| \(X\)  | factor matrix with size \(N \times K\) |
| \(a_{m,k}\) | the element of \(A\) at \(m\) row and \(k\) column |
| \(x_{n,k}\) | the element of \(X\) at \(n\) row and \(k\) column |
| \(Z^{(1)}\) | mask matrix for \(A\) |
| \(Z^{(2)}\) | mask matrix for \(X\) |
| \(z_{m,k}^{(1)}\) | the mask binary variable for element of \(A\) at \(m\) row and \(k\) column |
| \(z_{n,k}^{(2)}\) | the mask binary variable for element of \(X\) at \(n\) row and \(k\) column |
| \(V^{(1)}\) | loading matrix for \(A\) |
| \(V^{(2)}\) | loading matrix for \(X\) |
| \(v_{m,k}^{(1)}\) | the loading variable for element of \(A\) at \(m\) row and \(k\) column |
| \(v_{n,k}^{(2)}\) | the loading variable for element of \(X\) at \(n\) row and \(k\) column |
| \(\nu_k^{(1)}\) | \(k\) stick weight of IBP for \(A\) |
| \(\nu_k^{(2)}\) | \(k\) stick weight of IBP for \(X\) |
| \(\theta\) | model parameters for Bivariate beta distribution or copula |
number of columns, we apply the dependent Indian Buffet process as the prior for two matrices.

Nonparametric NMF Framework, the data matrix is modeled as,

\[ Y = A \ast X^T = (V^{(1)} \odot Z^{(1)}) \ast (V^{(2)} \odot Z^{(2)})^T \]

where \( Z^{(1)} \) and \( Z^{(2)} \) are two binary matrices, \( V^{(1)} \) and \( V^{(2)} \) are loading matrices, and \( \odot \) denotes the Hadamard product. A dIBP is used as the prior for \( Z^{(1)} \) and \( Z^{(2)} \). The likelihood of nonnegative matrix factorization is,

\[
\begin{align*}
    a_{m,k} &= v_{m,k}^{(1)} \cdot z_{m,k}^{(1)}, v_{m,k}^{(1)} \sim \text{gamma}(1, \tau_1) \\
    x_{n,k} &= v_{n,k}^{(2)} \cdot z_{n,k}^{(2)}, v_{n,k}^{(2)} \sim \text{gamma}(1, \tau_2) \\
    y_{m,n} | a_{m,k}, x_{n,k} \sim \text{Exp}(y_{m,n}; \sum_k a_{m,k} \cdot x_{n,k} + \epsilon)
\end{align*}
\]

where \( \text{Exp}() \) denotes the exponential distribution, the selection of the gamma and exponential distributions for \( V \) is used to guarantee the nonnegativity of \( A \) and \( X \), and each element of \( Y \) satisfies an exponential distribution with support \([0, +\infty)\). The special parameters for the distributions are to retain the desired expectations of these distributions. For example, the expectation of distribution \( y_{m,n} \) is \( \sum_k a_{m,k} \cdot x_{n,k} + \epsilon \).

The implementation of nonnegative matrix factorization is only conducted to select a prior for \( Z^{(1)} \) and \( Z^{(2)} \). In the following subsections, we will introduce three implementations of this framework through three dIBPs.
4.1 Implementation by GP-based dIBP

The first dIBP is proposed based on Gaussian Process (GP) [26]. In this GP-based dIBP model, each stick weight \( \mu_k \) is used to generated a different number of columns of binary matrices. For the NMF, we can use this model as the prior for the matrices \( A \) and \( X \). The graphical model is shown in Fig. 1(b), and the generative process is as follows,

\[
\nu_j \sim \text{Beta}(\alpha, 1), \quad \mu_k = \prod_{j=1}^{k} \nu_j
\]

where the \( \mu_k \) are IBP sticks as in Eq. (2). This set of sticks is shared by two binary matrices through

\[
g_k \sim GP(0, \Sigma_k) \\
h_{n1,k}^{(1)} \sim GP(g_k, r_{n1,k}^{(1)}) \\
h_{n2,k}^{(2)} \sim GP(g_k, r_{n2,k}^{(2)}) \\
\Sigma_k(t, t') = \sigma^2 \exp(-\frac{(t-t')^2}{s^2})
\]

where \( \Sigma_k \) is the kernel function for the GP. The model details can be found in [26]. Since there are only two binary matrices, the GP is degenerated to a two-dimensional Gaussian distribution, and \( \Sigma_k \) is a \( 2 \times 2 \) matrix.

4.2 Implementation by Bivariate Beta Distribution-based dIBP

There are different ways to link two IBPs. The GP-based IBP uses the same set of sticks for different binary matrices. Here, we propose another method that links the initials of two IBPs, \( \nu \). In the original IBP, \( \nu \) satisfies a beta distribution with parameter \((\alpha, 1)\). Therefore, we use a joint distribution \((\nu^{(1)}, \nu^{(2)})\) with beta distributions \(\text{Beta}(\alpha_1, 1)\) and \(\text{Beta}(\alpha_2, 1)\) as marginal distributions. Following this idea, the intuitive candidate would be Dirichlet distribution. However, there is a strictly negative relation, \( \nu^1 + \nu^2 = 1 \), between the samples of the Dirichlet distribution, but we hope to preserve the freedom of two \((\nu^{(1)}, \nu^{(2)})\).

Instead of the Dirichlet distribution, a bivariate beta distribution [31] is adopted. The probability density function is defined as,

\[
p(x, y) = \frac{x^{a-1}y^{b-1}(1-x)^{b+c-1}(1-y)^{a+c-1}}{B(a, b, c)(1-xy)^{a+b+c}} \\
\text{s.t.}, \quad 0 \leq x, y \leq 1, \quad a, b, c > 0
\]

where \(a, b, c\) are three parameters of this distribution. One merit of this bivariate beta distribution is that two marginal distributions are,

\[
x \sim \text{Beta}(a, c), \quad y \sim \text{Beta}(b, c)
\]

Another metric is that this distribution models positive correlation between \((\mu^{(1)}, \mu^{(2)})\) with range \([0, 1]\) adjusted by the three parameters \((a, b, c)\) compared to the Dirichlet distribution. Here, we set \(c\) of the bivariate beta distribution to 1. The reason is because we must ensure that the marginal distribution of each \(\nu\) is a beta distribution with parameter \((\alpha, 1)\). This condition is for the distribution of binary matrices generated by the model that satisfies the IBPs. Considering Eq. (7), we give \(c\) a fixed value. Even with a fixed value for \(c\), the bivariate beta distribution in Eq. (6) is still able to model different correlations of two variables. For example, when \(a = 2.5\) and \(b = 4\), the correlation between \(x\) and \(y\) is 0.978; when \(a = 0.05\) and \(b = 0.1\), the correlation between \(x\) and \(y\) is 0.08.

With the desired bivariate distribution in hand, we build the model as follows,

\[
(\nu^{(1)}_k, \nu^{(2)}_k) \sim \text{biBeta}(\theta), \quad \theta : \{a, b, c = 1\} > 0 \\
\nu^{(1)}_{a,k} \sim \text{Bernoulli}(\mu^{(1)}_{k}), \quad \mu^{(1)}_{k} = \prod_{j=1}^{k} \nu^{(1)}_{j} \\
\nu^{(2)}_{a,k} \sim \text{Bernoulli}(\mu^{(2)}_{k}), \quad \mu^{(2)}_{k} = \prod_{j=1}^{k} \nu^{(2)}_{j}
\]

where \(\text{biBeta}(\cdot)\) denotes the bivariate beta distribution in Eq. (6) and \(\theta\) denotes the parameters of the distribution. The graphical model is shown in Fig. 1(c).

The likelihood function is the same as Eq. (4). Although the bivariate beta distribution-based dIBP has extended the freedom of the relation between \(\nu^{(1)}\) and \(\nu^{(2)}\), the relation is restricted to positive relations in bivariate beta distribution. We use the copula to capture more freedom of relations.

4.3 Implementation by Copula-based dIBP

Copula [32] is another alternative for linking two variables with given marginal distributions, and is used to define a joint distribution for variables with known marginal distributions. Here, we select the Farlie-Gumbel-Morgenstern (FGM) Copula [33] as an example.

The definition of the FGM Copula is,

\[
C_{\rho}(u, v) = uv + \rho uv(1-u)(1-v), \quad \rho \in [-1, 1] \\
c_{\rho}(u, v) = 1 + \rho(2u - 1)(2v - 1)
\]

where

\[
u = F(\nu^{(1)}) \sim (\nu^{(1)})^{\alpha_1}, \quad f(\nu^{(1)}) \sim \alpha_1 \cdot (\nu^{(1)})^{\alpha_1-1} \\
v = F(\nu^{(2)}) \sim (\nu^{(2)})^{\alpha_2}, \quad f(\nu^{(2)}) \sim \alpha_2 \cdot (\nu^{(2)})^{\alpha_2-1}
\]

and \(C_{\rho}(u, v)\) is the cumulative distribution function, \(c_{\rho}(u, v)\) is the probability density function, and \(\rho\) is the parameter of the FGM copula. \(u\) and \(v\) are two marginal distributions that are known in advance. For our dIBP model, these marginal distributions are beta distribution with parameters \((\alpha_1, 1)\) and \((\alpha_2, 1)\).
The correlation or dependence of $\nu^{(1)}$ and $\nu^{(2)}$ is modeled or reflected by the value of $\rho$. As shown in Fig. 2, the different correlation (i.e., positive or negative) can be captured by the value of $\rho$. In particular, if $\rho = 0$, there is no correlation between two marginal distributions.

FGM-based dIBP is defined by replacing the bivariate beta distribution-based graphical model with a copula-based one. The graphical model is the same as the bivariate beta distribution but not the exact form of the probability density function of the beta distributions:

$$p(\nu^{(1)}_k, \nu^{(2)}_k) = c_\rho(\theta), \quad \theta \in \{\rho \in [-1, 1], \alpha_1 > 0, \alpha_2 > 0\} \tag{11}$$

The graphical model is the same as the bivariate beta distribution in Eq. (8) with joint distribution defined by the FGM copula,

Comparing the bivariate beta distribution, the advantage of copulas is that we can easily obtain the cumulative distribution function and probability density function of $(\nu^{(1)}_k, \nu^{(2)}_k)$, because we only know the exact form of probability density function of the bivariate beta distribution but not the exact form of the cumulative distribution function. This will impact on the model inference, which will be introduced later.

One advantage of bivariate beta distribution-based and copula-based dIBP compared with GP-based dIBP that less latent variables are involved. This is easily observed through the graphical models in Fig. 1. More latent variables will slow the convergence of the Gibbs sampler.

5 Model Inference

With data $Y$ in hand, the objective of this section is to estimate the hidden variables by a properly designed MCMC sampler for their posterior distribution, $p(\mu, Z, V, \theta | Y)$. It is difficult to perform posterior inference under infinite mixtures, thus a common work-around solution in nonparametric Bayesian learning is to use a truncation method. The truncation method, which uses a relatively big $K$ as the (potential) maximum number of factors, is widely accepted.

Note that GP-based dIBP is not our contribution. Our contribution is to link the GP-based dIBP with the nonnegative matrix factorization likelihood. The inference of this model, which is not our contribution, is given in the Appendix for the self-contained purpose. The details can be found in [26].

5.1 Update $\mu$

When updating $\mu_k$, we need to find the conditional distribution of $\mu_k$ given $\mu_{k-1}$ and $\mu_{k+1}$, because the order of $\mu$ must be maintained to make the marginal distributions of $Z$ satisfy two IBPs. The acceptance ratio of the M-H sampler for $\mu_{(k)} = [\mu^{(1)}_{(k)}, \mu^{(2)}_{(k)}]$ is,

$$\min \left(1, \frac{p(Z | \mu^{(1)}_{(k)}) \cdot p(\mu^{(1)}_{(k)} | \mu_{(k-1)}, \mu_{(k-1)})}{p(Z | \mu^{(2)}_{(k)}) \cdot p(\mu^{(2)}_{(k)} | \mu_{(k-1)}, \mu_{(k-1)})} \times \frac{q(\mu^{(2)}_{(k)})}{q(\mu^{(1)}_{(k)})} \right) \tag{12}$$

where $p(Z | \mu_{(k)})$ is the likelihood of $\mu_{(k)}$ to generate $k$th column of binary matrix $Z$.

The $p(\mu^{(1)}_{(k)} | \mu_{(k-1)}, \mu_{(k-1)})$ in Eq. (12) is the conditional probability density function of $\mu^{(1)}_{(k)}$ within the range of $[\mu_{(k-1)}, \mu_{(k-1)}]$. This will be different when different strategies are used to link $\mu^{(1)}_{(k)}$ and $\mu^{(2)}_{(k)}$. Next, we derive the conditional probability density function of $\mu_{(k)}$ as follows:

For the first column,

$$\mu^{(1)}_1 = \nu^{(1)}_1, \quad \mu^{(1)}_2 = \nu^{(1)}_2 \tag{13}$$

and

$$p(\mu^{(1)}_1, \mu^{(2)}_1) = p(\nu^{(1)}_1, \nu^{(2)}_1) \tag{14}$$

For the second column,

$$\mu^{(2)}_1 = \nu^{(2)}_1 \mu^{(1)}_1, \quad \mu^{(2)}_2 = \nu^{(2)}_2 \mu^{(1)}_2 \tag{15}$$
and
\[ p(\mu_1^{(2)}, \mu_2^{(2)}) = p(\nu_1^{(2)} \cdot \mu_1^{(1)}, \nu_2^{(2)} \cdot \mu_2^{(1)}) \]
\[ = \frac{p(\nu_1^{(2)}, \nu_2^{(2)})}{|J^{(1)}|} \]
\[ p(\mu_1^{(2)}, \mu_2^{(2)}) = \frac{p(\mu_1^{(2)}, \mu_2^{(2)})}{\mu_1^{(1)} \cdot \mu_2^{(1)}} \]
where \( J^{(1)} \) is the Jacobian matrix,
\[ J^{(1)} = \begin{bmatrix} \mu_1^{(1)} & 0 \\ 0 & \mu_2^{(1)} \end{bmatrix} \]
For the \( n \)th column,
\[ \mu_1^{(n)} = \nu_1^{(n)} \mu_1^{(n-1)}, \quad \mu_2^{(n)} = \nu_2^{(n)} \mu_2^{(n-1)} \]
and
\[ p(\mu_1^{(n)}, \mu_2^{(n)}) = p(\nu_1^{(n)} \cdot \mu_1^{(n-1)}, \nu_2^{(n)} \cdot \mu_2^{(n-1)}) \]
\[ = \frac{p(\nu_1^{(n)}, \nu_2^{(n)})}{|J^{(n-1)}|} \]
\[ = \frac{p(\frac{\mu_1^{(n)}}{\mu_1^{(n-1)}}, \frac{\nu_2^{(n)}}{\mu_2^{(n-1)}})}{\mu_1^{(n-1)} \cdot \mu_2^{(n-1)}} \]
where \( J^{(n-1)} \) is the Jacobian matrix,
\[ J^{(n-1)} = \begin{bmatrix} \mu_1^{(n-1)} & 0 \\ 0 & \mu_2^{(n-1)} \end{bmatrix} \]
To summarize, the conditional density of \( \mu^{(n)} \) is
\[ p(\mu^{(n)} | \mu^{(n-1)}, \mu^{(n+1)}) = \frac{p(\mu_1^{(n-1)}, \mu_2^{(n-1)})}{p(\mu_1^{(n-1)} \cdot \mu_2^{(n-1)})} \]
The different methods of linking \( \nu_1 \) with \( \nu_2 \) will lead to a different joint probability density function for them. As introduced in Section 4, we have given two specific forms of their joint probability in Eq. (6) and Eq. (9).

The proposal distribution, \( q(\cdot) \), is selected as the product of two independent truncated beta distributions,
\[ q(\mu_k) = \text{beta}(\mu_k^{(1)}, \alpha_1, 1) \cdot \text{beta}(\mu_k^{(2)}, \alpha_2, 1) \]
\[ \mu_k^{(1)} \sim \text{beta}(\frac{\alpha_1}{\alpha}, 1), \quad \mu_k^{(2)} \sim \text{beta}(\frac{\alpha_2}{\alpha}, 1) \]
where \( \alpha = \sum_{k=1}^{K} \alpha_k \)

5.2 Update \( Z \)
Two binary matrices, \( Z : \{Z^{(1)}, Z^{(2)}\} \), can be updated separately, since each element in two matrices satisfies a Bernoulli distribution with the following conditional probabilities,
\[ p(z_{m,k}^{(1)} = 1) \]
\[ \propto \mu_k^{(1)} \prod_m \text{Exp}(y_{m,n} \cdot \sum_l a_{m,l} \cdot x_{n,l} + \epsilon) \]
\[ p(z_{m,k}^{(2)} = 1) \]
\[ \propto \mu_k^{(2)} \prod_m \text{Exp}(y_{m,n} \cdot \sum_l v_{m,l} z_{m,l} v_{n,l} z_{n,l} + \epsilon) \]
\[ p(z_{m,k}^{(1)} = 0) \]
\[ \propto (1 - \mu_k^{(1)}) \prod_m \text{Exp}(y_{m,n} \cdot \sum_l a_{m,l} \cdot x_{n,l} + \epsilon) \]
\[ p(z_{m,k}^{(2)} = 0) \]
\[ \propto (1 - \mu_k^{(2)}) \prod_m \text{Exp}(y_{m,n} \cdot \sum_l v_{m,l} z_{m,l} v_{n,l} z_{n,l} + \epsilon) \]
where \( \epsilon \) is small positive number.

5.3 Update \( V \)
Since the prior for \( V \) is Gamma distribution and the likelihood is exponential distribution, the conditional distribution for \( V \) is,
\[ p(v_{m,k}^{(1)} | z_{m,k}^{(1)} a_{m,k}) \]
\[ \propto \text{gam}(v_{m,k}^{(1)} \cdot \alpha_1, \sum_k a_{m,k} \cdot x_{n,k} + \epsilon) \]
\[ \text{gam}(v_{m,k}^{(1)} \cdot \alpha_1, \sum_k a_{m,k} \cdot x_{n,k} + \epsilon) \]
\[ \propto \text{gam}(v_{m,k}^{(2)} \cdot \alpha_2, \sum_k v_{m,l} z_{m,l} v_{l,n} z_{n,l} + \epsilon) \]
\[ \text{gam}(v_{m,k}^{(2)} \cdot \alpha_2, \sum_k v_{m,l} z_{m,l} v_{l,n} z_{n,l} + \epsilon) \]

5.4 Update \( \theta \)
In Fig. 1(c), the graphical model has a parameter \( \theta \). For the different strategies to link \((\nu^{(1)}, \nu^{(2)})\), the parameters must be different. Therefore, we design corresponding update methods for the proposed two strategies: bivariate beta distribution and copula.
Algorithm 1: Gibbs Sampler for GP-dIBP-NMF

Input: $Y$
Output: $A, X$
initialization;
while $i \leq \text{max}_{\text{iter}}$ do
// latent variables of dIBP
Update $\mu$ by Eq. (32);
Update $Z$ by Eq. (39) and (40);
Update $g$ by Eq. (34);
Update $h$ by Eq. (35);
Update $s$ by Eq. (38);
// latent variables of data of NMF
Update $V$ by Eq. (25) and (26);
i + +;
Select the sample with largest likelihood;
return $A$ and $X$;

Algorithm 2: Gibbs Sampler for BB-dIBP-NMF

Input: $Y$
Output: $A, X$
initialization;
while $i \leq \text{max}_{\text{iter}}$ do
// latent variables of dIBP
Update $\mu$ by Eq. (12);
Update $Z$ by Eq. (23) and (24);
Update $a$ and $b$ by Eq. (27);
// latent variables of data of NMF
Update $V$ by Eq. (25) and (26);
i + +;
Select the sample with largest likelihood;
return $A$ and $X$;

Algorithm 3: Gibbs Sampler for C-dIBP-NMF

Input: $Y$
Output: $A, X$
initialization;
while $i \leq \text{max}_{\text{iter}}$ do
// latent variables of dIBP
Update $\mu$ by Eq. (12);
Update $Z$ by Eq. (23) and (24);
Update $\alpha_1$ and $\alpha_2$ by Eq. (28);
Update $\rho$ by Eq. (29);
// latent variables of data of NMF
Update $V$ by Eq. (25) and (26);
i + +;
Select the sample with largest likelihood;
return $A$ and $X$;

5.4.1 Bivariate Beta Distribution

The parameters of bivariate beta distribution, $\theta : \{a, b\}$, are given two gamma priors. The conditional distributions are,

$$p([a, b]; \cdot) \propto \text{gam}([a, b]; hp) \prod_{k=1}^{K} p(\mu_k^{(1)}, \mu_k^{(2)} | a, b)$$ (27)

where $hp$ is the hyper-parameter of the prior for $a$ and $b$.

5.4.2 Copula

There are three parameters for each copula, $\theta : \{\rho, \alpha_1, \alpha_2\}$. Their conditional distributions are,

$$p([\alpha_1, \alpha_2]; \cdot) \propto \text{gam}([\alpha_1, \alpha_2]; hp) \prod_{k=1}^{K} c(\mu_k^{(1)}, \mu_k^{(2)} | \rho)$$ (28)

We give the $\rho$ of the FGM copula a uniform distribution on $[-1, 1]$,

$$p(\rho; \cdot) \propto \prod_{k=1}^{K} c(\mu_k^{(1)}, \mu_k^{(2)} | \rho)$$ (29)

After introducing the update methods, we summarize the inference for the three models in Algorithm 1 for the GP-based dIBP NMF (GP-dIBP-NMF) model, Algorithm 2 for the Bivariate beta distribution-based dIBP NMF (BB-dIBP-NMF) model, and Algorithm 3 for the Copula-based dIBP NMF (C-dIBP-NMF) model.

6 Experiments

In this section, we first express the ability to conduct matrix factorization and hidden factor number learning of the proposed algorithms through synthetic examples, followed by the performance of the algorithms on real-world tasks.

6.1 Synthetic dataset

We randomly generate a matrix $Y_{20 \times 30}$ with elements in $\{0, 1\}$. The proposed algorithms: GP-dIBP-NMF, BB-dIBP-NMF and C-dIBP-NMF are used for sparse nonnegative matrix factorization on this matrix. The results are shown in Fig. 4, Fig. 5 and Fig. 6, respectively. In each figure, we have visualized the generated matrix and its constructed version through the learned matrices $A$ and $X$ which are both visualized in the figure. It can be observed that the (dark) positions with zero values are reconstructed well by all three algorithms comparing when the original $Y$ and the reconstructed $Y$ are compared. Another observation is the sparsity of the factor matrices $A$ and $X$ whose number of columns will change with the iteration. The last but not least observation is the value log-likelihoods from the three algorithms. A bigger log-likelihood value means that the model fits the data better. The comparison shows that the BB-dIBP-NMF has similar performance to C-dIBP-NMF but is better than GP-dIBP-NMF. Except for log-likelihood, the convergence rate of GP-dIBP-NMF is the worst of the three models, as can be observed from the convergence curve in List L in Fig. 4, Fig. 5 and Fig.
6. It takes about 200 for BB-dIBP-NMF to converge, and 500 for C-dIBP-NMF to converge, but GP-dIBP-NMF needs more than 1500 iterations to reach convergence. This is due to the complex model structure of the GP-based dIBP. As well as the visualization, the reconstruction error is also quantitatively compared. The reconstruction error is defined as

$$\text{error} = \|Y - A \ast X^T\|_1$$  \hspace{1cm} (30)

The comparison is shown in Fig. 3. The solid (light blue) line in the figure shows the sparse NMF with different numbers of factors as the input. As shown in the figure, the worst error value is around 25. Since the proposed algorithms do not need to predefine the factor number, there are three lines in Fig. 3 to indicate the errors of the three algorithms.

In order to show the flexibility of our proposed dIBP comparing with GP-based dIBP, we firstly have designed a metric to measure this flexibility by comparing the number of non-zero entries of two loading matrices (A and X) from the models. As we claimed, our proposed model is with more flexibility to allow the numbers of non-zero entries of loading matrices more different from each other comparing GP-based dIBP. Therefore, we have evaluated the mean of the differences between the corresponding columns of A and X as

$$\frac{1}{K} \sum_{k=1}^{K} |n_k^{(1)} - n_k^{(2)}|$$

where $K$ is the number of columns of both matrices, $n_k^{(1)}$ is the number of non-zero entries of $k$-th column of $A$, and $n_k^{(2)}$ is the number of non-zero entries of $k$-th column of $X$. We have randomly generated ten matrices with same size: $20 \times 30$, and run three models on ten matrices. The designed metric has been evaluated by on the outputs of models. As shown in Fig. 7, we can see that the BB-dIBP-NMF and C-dIBP-NMF are with more fluctuations and larger non-zero entry number differences comparing with GP-dIBP-NMF. It demonstrates that the proposed dIBPs are more flexible than GP-based dIBP.

6.2 Real-world Dataset

In this subsection, we apply the proposed algorithms on two real-world tasks: document clustering and recommender system. For each task, we will compare the proposed algorithms with sparse NMF with different numbers of factors.

6.2.1 Document Clustering

The real-world datasets used for the document clustering task are:

- **Cora Dataset**\(^1\) The Cora dataset consists of 2708 scientific publications classified into one of seven classes. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 1433 unique words.

- **Citeseer Dataset** The CiteSeer dataset consists of 3312 scientific publications. Each publication

1. http://linqs.cs.umd.edu/projects/projects/lbc/
Fig. 5. Results on synthetic dataset from BB-dIBP-NMF.

in the dataset is also described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 3703 unique words. The labels of these papers are set as their research areas, such as AI (Artificial Intelligence), ML (Machine Learning), Agents, DB (Database), IR (Information Retrieval) and HCI (Human-Computer Interaction).

The evaluation metrics of document clustering are Jaccard Coefficient (JC), Folkes&Mallows (FM) and F1 measure (F1). Given a clustering result,

- \( a \) is the number of two points that are in the same cluster of both benchmark results and clustering results;
- \( b \) is the number of two points that are in the same cluster of benchmark results but in different clusters of clustering results;
- \( c \) is the number of two points that are not in the same cluster of the two benchmark results but are in the same cluster of clustering results.

and three metrics are computed the equations in Table 2 (bigger means better).

When document clustering is selected as the task, the output of the algorithms (\( A \) is the document-factor matrix) is used as the input of the spectral clustering algorithm. Since the common clustering algorithm is used, the clustering performance of different NMF algorithms is only determined by the factors \( A \). The clustering results evaluated by the metrics in Table 2 on two different datasets are shown in Fig. 8 and Fig. 9. In each figure, we run the sparse nonnegative matrix factorization in Eq. 1 with different factor numbers (the x-axis). All the algorithms have 1000 iterations. The results of the proposed three algorithms are also shown in the figure through three horizontal

Fig. 6. Results on synthetic dataset from C-dIBP-NMF.

Fig. 7. Results on synthetic dataset to show the flexibility of the different models. The x-axis denotes the trial IDs (order is irrelevant)
Table 2: Evaluation metrics of document clustering

| Metric          | Formula                                                                 |
|-----------------|--------------------------------------------------------------------------|
| Jaccard Coefficient | \[ JC = \frac{a}{a + b + c} \] |
| Folkes & Mallows | \[ FM = \left( \frac{a}{a + b} \cdot \frac{a}{a + c} \right)^{1/2} \] |
| F1 measure      | \[ F1 = \frac{2ac}{2ac + a^2 + b^2} \] |

The learned factor numbers are 10 (from BB-dIBP-NMF), 12 (from C-dIBP-NMF) and 12 (from GP-dIBP-NMF) for the Citeseer dataset; 17 (from BB-dIBP-NMF), 15 (from C-dIBP-NMF) and 22 (from GP-dIBP-NMF) for the Cora dataset. The reason why the horizontal lines are used is that the proposed algorithms do not need the factor number as an input, so the clustering results are not dependent of the x-axis (factor number). To summarize, without the prior knowledge of the number of factors, the proposed algorithms achieve relatively good performance on the document clustering task. Of the three algorithms, BB-dIBP-NMF achieves the best performance. C-dIBP-NMF and GP-dIBP-NMF achieve similar performances on two datasets only except the F1 on the Cora dataset.

6.2.2 Recommender System

The dataset for this task is MovieLens 100K. This data set consists of 100,000 ratings (1-5) from 943 users on 1682 movies, and each user has rated at least 20 movies. In the following experiment, we use a 5-fold method for cross validation (20% ratings are kept as the test data and the remaining 80% ratings are used as the training data).

When used for the recommender system, the output of the algorithms A (user-factor matrix) and X (movie-factor matrix) reconstruct the rating matrix \( Y_r = A \ast X^T \) on the retained test ratings. The quantitative evaluation is

\[ MAE = \| Y_r - Y_{test} \|_1 \]  

(31)

Here, we do not normalize \( MAE \) since it does not impact on the comparison. Note that the sparse NMF in Eq. 1 needs to be revised to ignore the test ratings. The weighted NMF [34], [35] can be adopted and a mask matrix with ones on the training ratings and zeros on the test ratings is constructed. Ignoring the test ratings can be easily achieved by only considering the training ratings in the updates for \( Z \) in Eq. 23 and Eq. 24 and updates for \( V \) in Eq. 25 and Eq. 26.

The recommendation results are shown in Fig. 10. Considering the 5-fold cross test, the mean of the five groups is plotted in the figure with a solid line, and the standard deviations are shown in the figure by the gray area around the mean plot. Three subfigures in Fig. 10 denote the comparison of the sparse NMF with the proposed three algorithms, for which the results are plotted by dashed lines. Similarly, the standard deviations are also given by the area around the dashed lines. It can be observed that the SNMF has better performance than the proposed algorithms only when the factor number is smaller than 20. The proposed algorithms outweigh the SNMF (smaller MAE values) for the remaining choices for the factor number. We want to highlight again that the factor number is unknown before running SNMF. There is a wide possible range for this number. Within this wide range, there are only approximately 20 choices that give better performance than the proposed algorithms. Three algorithms have similar performance to one another. The BB-dIBP-NMF algorithm is the best, with relatively smaller MAE, and C-dIBP-NMF is the worst. The C-dIBP-NMF, however, has the smallest variance. The small variance means that this algorithm achieves similar performance on different test data. In other words, an algorithm with small variance will be stable on different test data. We can see that SNMF is not stable compared with the proposed algorithms.

7 Conclusion and Further Study

The renowned nonnegative matrix factorization is advantageous for many machine learning tasks, but the assumption that the dimension of the factors is known in advance makes NMF impractical for many applications. To resolve this issue, we have proposed a nonparametric NMF framework based on dIBP to remove the assumption. First, a model is built by implementing this framework using GP-based dIBP, which successfully removes the assumption but suffers from larger model complexity and less flexibility. Then, we have proposed two new dIBPs through a bi-variate beta distribution and a copula. One advantage of the models based on new dIBPs is that they have simpler model structures than models with GP-based dIBP. Lastly, three inference algorithms have been designed for the proposed three models, respectively. The experiments on the synthetic and real-world datasets demonstrate the capability of the proposed models to perform NMF without predefining the dimension number, and the models based on the new dIBPs have better convergence rates and more flexibility than the model based on GP-based dIBP.

The bivariate beta distribution-based model and copula-based model have achieved comparative performances both on document clustering and recommender system. We here give hints for their usage: 1) if the data correlation is within the range \([-0.3, 0.3]\), the FGM copula-based model will have better flexibility than the bivariate beta distribution-based model; 2) if the data correlation is outside the range \([-0.3, 0.3]\), the bivariate beta distribution-based model is more reasonable than the FGM copula-based model.

One possible future study for this work is the aspect of efficiency. Current Gibbs sampling inference is not

2. http://grouplens.org/datasets/movielens/
Fig. 8. Document clustering comparisons on \textit{Citeseer} dataset between GP-dIBP-NMF (gIBP-NMF), BB-dIBP-NMF (bIBP-NMF), C-dIBP-NMF (cIBP-NMF), and Sparse NMF (SNMF).

Fig. 9. Document clustering comparisons on \textit{Cora} dataset between GP-dIBP-NMF (gIBP-NMF), BB-dIBP-NMF (bIBP-NMF), C-dIBP-NMF (cIBP-NMF), and Sparse NMF (SNMF).

Fig. 10. Recommendation comparisons on \textit{MovieLens-100K} dataset between GP-dIBP-NMF (gIBP-NMF), BB-dIBP-NMF (bIBP-NMF), C-dIBP-NMF (cIBP-NMF), and Sparse NMF (SNMF).
efficient enough for big data. Our further study will focus on the efficiency of the inference of the proposed models using the variational inference strategy.

Acknowledgment

The research work reported in this paper was partly supported by the Australian Research Council (ARC) under discovery grant DP140101366 and the China Scholarship Council. This work was jointly supported by the National Science Foundation of China under grant no. 61471232.

References

[1] R. Sandler and M. Lindenbaum, “Nonnegative matrix factorization with earth mover’s distance metric for image analysis,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 33, no. 8, pp. 1590–1602, Aug 2011.

[2] Z. He, S. Xie, R. Zhunek, G. Zhou, and C. Chichocki, “Symmetric non-negative matrix factorization: Algorithms and applications to probabilistic clustering,” Neural Networks, IEEE Transactions on, vol. 22, no. 12, pp. 2117–2131, Dec 2011.

[3] J. Choo, C. Lee, C. Reddy, and H. Park, “Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization,” Visualization and Computer Graphics, IEEE Transactions on, vol. 19, no. 12, pp. 1992–2001, Dec 2013.

[4] X. Luo, M. Zhou, X. Xia, and Q. Zhu, “An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems,” Industrial Informatics, IEEE Transactions on, vol. 10, no. 2, pp. 1273–1284, May 2014.

[5] M. Ye, Y. Qian, and J. Zhou, “Multitask sparse nonnegative matrix factorization for joint spectral-spatial hyperspectral imagery denoising,” Geoscience and Remote Sensing, IEEE Transactions on, vol. 53, no. 5, pp. 2621–2639, May 2015.

[6] M. Heiler and C. Schnör, “Learning sparse representations by non-negative matrix factorization and sequential cone programming,” The Journal of Machine Learning Research, vol. 7, pp. 1385–1407, 2006.

[7] J. Kim and H. Park, “Sparse nonnegative matrix factorization for clustering,” Technical Report, 2008.

[8] Z. Yang, G. Zhou, S. Xie, S. Ding, J.-M. Yang, and J. Zhang, “Blind spectral unmixing based on sparse nonnegative matrix factorization,” Image Processing, IEEE Transactions on, vol. 20, no. 4, pp. 1112–1125, April 2011.

[9] R. M. Neal, “Markov chain sampling methods for dirichlet process mixture models,” Journal of Computational and Graphical Statistics, vol. 9, no. 2, pp. 249–265, 2000.

[10] T. L. Griffiths and Z. Ghahramani, “Infinite latent feature models and the indian buffet process,” in NIPS, vol. 18, 2005, pp. 475–482.

[11] ——, “The Indian buffet process: An introduction and review,” The Journal of Machine Learning Research, vol. 12, pp. 1185–1224, Jan. 2011.

[12] S. J. Gershman and D. M. Blei, “A tutorial on bayesian nonparametric models,” Journal of Mathematical Psychology, vol. 56, no. 1, pp. 1–12, 2012.

[13] C. H. Q. Ding, T. Li, and M. I. Jordan, “Convex and semi-nonnegative matrix factorizations,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 32, no. 1, pp. 45–55, Jan. 2010.

[14] Y. W. Teh, D. Görür, and Z. Ghahramani, “Stick-breaking construction for the indian buffet process,” in AISTATS, 2007, pp. 556–563.

[15] S. Huh, M. Das Gupta, and J. Xiao, “Supervised nonnegative matrix factorization,” 2013, uS Patent 8,498,949.

[16] C.-J. Lin, “On the convergence of multiplicative update algorithms for nonnegative matrix factorization,” Neural Networks, IEEE Transactions on, vol. 18, no. 6, pp. 1589–1596, Nov 2007.

[17] Y. Yuan, X. Li, Y. Pang, X. Lu, and D. Tao, “Binary sparse nonnegative matrix factorization,” Circuits and Systems for Video Technology, IEEE Transactions on, vol. 19, no. 5, pp. 772–777, May 2009.

APPENDIX A

The Conditional Distributions for the GP-based dBiP NMF

The conditional distributions are:

**Sampling μ**

\[
p(μ_k | \cdots) \propto \mu_k^g \prod_{t=1}^{2N} \left( \gamma_k \right)^{\gamma_k} \left( 1 - \gamma_k \right)^{-\gamma_k} \]

where

\[
\gamma_k = F(F^{-1}(μ_k | 0, Σ_k) + η)^2 - g_k(0, η)^2
\]

**Sampling g**

\[
p(g_k | \cdots) \propto N(g_k | 0, Σ_k) \prod_{t=1}^{N} N(h_k \cdot g_k | \gamma_k, η^2)
\]

where N() denotes normal distribution.
Sampling \( h \)
\[
p(h_{n,k}^t | \cdots) \propto \mathcal{N}(g_k, \eta^2)
\]
(35)
with support
\[
\begin{align*}
\begin{cases}
    h_{n,k}^t \in (-\infty, \tilde{\mu}_k^t] & \text{if } z_{n,k}^t = 1 \\
    h_{n,k}^t \in [\tilde{\mu}_k^t, +\infty) & \text{if } z_{n,k}^t = 0
\end{cases}
\end{align*}
\]
(36)
where
\[
\tilde{\mu}_k^t = F^{-1}(\mu_k|0, \Sigma_k^{(f,t)} + \eta^2)
\]
Sampling \( s \)
\[
P(s| \cdots) \propto \text{gamma}(s; h_s, 1) \prod_k \mathcal{N}(g_k|0, \Sigma_k)
\]
(38)

Sampling \( Z \)
\[
\begin{align*}
    &p(z_{m,k}^{(1)} = 1| \cdots) \propto \gamma_k^1 \\
    &\prod_n \text{Exp}(y_{m,n}, \sum_l y_{m,l}^{(1)} z_{m,l}^{(1)} z_{n,l}^{(2)} + \epsilon) \\
    &p(z_{m,k}^{(1)} = 0| \cdots) \propto (1 - \gamma_k^1) \\
    &\prod_n \text{Exp}(y_{m,n}, \sum_l y_{m,l}^{(1)} z_{m,l}^{(1)} z_{n,l}^{(2)} + \epsilon)
\end{align*}
\]
(39)
and
\[
\begin{align*}
    &p(z_{n,k}^{(2)} = 1| \cdots) \\
    &\propto \gamma_k^2 \prod_m \text{Exp}(y_{m,n}, \sum_l y_{m,l}^{(1)} z_{m,l}^{(1)} z_{n,l}^{(2)} + \epsilon) \\
    &p(z_{n,k}^{(2)} = 0| \cdots) \\
    &\propto (1 - \gamma_k^2) \prod_m \text{Exp}(y_{m,n}, \sum_l y_{m,l}^{(1)} z_{m,l}^{(1)} z_{n,l}^{(2)} + \epsilon)
\end{align*}
\]
(40)
where \( \gamma_k^1 \) is same as in Eq. (33).
Xiangfeng Luo is a professor in the School of Computers, Shanghai University, China. Currently, he is a visiting professor in Purdue University. He received the master’s and PhD degrees from the Hefei University of Technology in 2000 and 2003, respectively. He was a postdoctoral researcher with the China Knowledge Grid Research Group, Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), from 2003 to 2005. His main research interests include Web Wisdom, Cognitive Informatics, and Text Understanding. He has authored or co-authored more than 50 publications and his publications have appeared in IEEE Trans. on Automation Science and Engineering, IEEE Trans. on Systems, Man, and Cybernetics-Part C, IEEE Trans. on Learning Technology, Concurrency and Computation: Practice and Experience, and New Generation Computing, etc. He has served as the Guest Editor of ACM Transactions on Intelligent Systems and Technology. Dr. Luo has also served on the committees of a number of conferences/workshops, including Program Co-chair of ICWL 2010 (Shanghai), WISM 2012 (Chengdu), CTUW2011 (Sydney) and more than 40 PC members of conferences and workshops.