Doubly Nonparametric Sparse Nonnegative Matrix Factorization Based on Dependent Indian Buffet Processes

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

***Abstract*— Sparse nonnegative matrix factorization (SNMF) aims to factorize a data matrix into two optimized nonnegative sparse factor matrices, which could benefit many tasks, such as document-word co-clustering. However, the traditional SNMF typically assumes the number of latent factors (i.e., dimensionality of the factor matrices) to be fixed. This assumption makes it inflexible in practice. In this paper, we propose a doubly sparse nonparametric NMF framework to mitigate this issue by using dependent Indian buffet processes (dIBP). We apply a correlation function for the generation of two stick weights associated with each column pair of factor matrices while still maintaining their respective marginal distribution specified by IBP. As a consequence, the generation of two factor matrices will be columnwise correlated. Under this framework, two classes of correlation function are proposed: 1) using bivariate Beta distribution and 2) using Copula function. Compared with the single IBP-based NMF, this paper jointly makes two factor matrices nonparametric and sparse, which could be applied to broader scenarios, such as co-clustering. This paper is seen to be much more flexible than Gaussian process-based and hierarchial Beta process-based dIBPs in terms of allowing the two corresponding binary matrix columns to have greater variations in their nonzero entries. Our experiments on synthetic data show the merits of this paper compared with the state-of-the-art models in respect of factorization efficiency, sparsity, and flexibility. Experiments on real-world data sets demonstrate the efficiency of this paper in document-word co-clustering tasks.**

***Index Terms*— Co-clustering, nonnegative matrix factorization, probability graphical model, text mining.**

I. Introduction

SPARSE nonnegative matrix factorization (SNMF) [1] aims to factorize a matrix into two optimized sparse nonnegative factor matrices, which is recognized as an efficient tool for unsupervised learning in many research areas, such as image processing [2], gene analysis, and recommender systems [3]. The improvement of SNMF over classical NMF is due to its sparse data representation [4], which is generally desirable, because it assists human understanding (e.g., with gene expression data), reduces computational costs, and obtains better generalization in learning algorithms [5]. When two factor matrices from NMF have the sparsity constraint, they are referred to as *doubly sparse NMF*.

A motivating example that could benefit from *doubly sparse NMF* is the document-word co-clustering task [6], [7], which clusters documents and words simultaneously given a document-word count matrix. Document clustering is useful for organizational purposes or browsing, and word clustering is useful for the automatic construction of a statistical thesaurus or the enhancement of queries [8]. Traditional separative clustering methods do not utilize the relationship between documents and words well, whereas co-clustering achieves much better performance in terms of discovering the structure of data [9]. With the new sparse representations from *doubly sparse NMF*, documents and words could be simultaneously and accurately clustered by considering their interior relations.

The assumption that the dimensionality of factor matrices from doubly sparse NMF needs to be predefined, however, prevents its use in many real-world applications. Normally, this dimensionality is assigned by experts with domain knowledge, but inaccurate assignment will impact the performance of doubly sparse NMF on such applied tasks as document-word co-clustering. Furthermore, the increase in the amount of data and the complexity of the tasks means that even experts are inadequate for this job. Therefore, it is more reasonable and practical to automatically learn the dimensionality from the data (which is known as *nonparametricity*). An intuitive solution from Bayesian nonparametric learning [10], which utilizes stochastic processes [11], [12] as the tools for data analysis, is to use the Indian buffet process (IBP) [13], [14] as the prior for the factor matrices. Unfortunately, IBP can only enable one factor matrix *sparsity* and *nonparametricity* (called single nonparametric sparse NMF [sIBP-NMF] in this paper). Despite its success on some tasks (e.g., factor analysis), sIBP-NMF fails in a number of broader tasks, such as the aforementioned document-word co-clustering. Therefore, our idea is to jointly assign two factor matrices IBP priors (named doubly nonparametric sparse NMF), thus endowing both factor matrices with sparsity and nonparametricity.

In this paper, we propose a doubly nonparametric sparse NMF framework to enable traditional NMF with both *double*

Manuscript received March 4, 2016; revised November 15, 2016 and February 24, 2017; accepted February 25, 2017. Date of publication April 12, 2017; date of current version April 16, 2018. This work was supported in part by the Australian Research Council discovery under Grant DP140101366 and in part by the National Science Foundation of China under Grant 61471232. *(Corresponding Author: Jie Lu.)*

J. Xuan, J. Lu, G. Zhang, and R. Y. D. Xu are with the Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia and also with the School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China (e-mail: junyu.xuan@uts.edu.au; jie.lu@uts.edu.au; guangquan.zhang@uts.edu.au; yida.xu@uts.edu.cn).

X. Luo is with the School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China (e-mail: luoxf@shu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2017.2676817

sparsity and *nonparametricity*. As the core of the proposed framework, two new dependent IBPs (dIBPs) are innovatively designed as the joint prior for the two factor matrices from NMF based on bivariate Beta distribution and copula. It is dIBP that helps us endow the NMF with both *double sparsity* and *nonparametricity*. Two closely related approaches use GP-based dIBP [15] and HBP-based dIBP [16]. By comparison, the proposed dIBPs allow both factor matrices to have much greater flexibility and variation in terms of their nonzero entries. Furthermore, 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 doubly sparse nonparametric NMF framework based on new dIBPs with a simpler model structure than GP-based dIBP. Nevertheless, introducing bivariate Beta distribution and copula presents a challenge for the model inference. We have designed four inference algorithms for four implementations of the doubly nonparametric sparse NMF framework: GP-based dIBP model, HBP-based dIBP model, bivariate Beta distribution-based dIBP model, and copula-based dIBP model. The experiments on the synthetic data show the merits of this paper compared with the traditional NMF, single IBP-based NMF, GP-based NMF, and HBP-based NMF on factorization efficiency, sparsity, and correlation flexibility. The experiments on real-world data sets show that the proposed models perform well in the document-word co-clustering task without explicitly predefining the dimensionality of the factor matrices.

The contributions of this paper are summarized as follows.

1) Two new dIBPs (i.e., bivariate Beta distribution-based and copula-based) with a simpler model structure and larger correlation flexibility are proposed as alternatives to the existing GP-based and HBP-based dIBPs.
2) Four dIBP-based nonparametric doubly SNMF models are proposed to endow the traditional NMF with both *double sparsity* and *nonparametricity*.

The rest of this paper is organized as follows. Preliminary details of NMF and IBP are briefly introduced in Section II. Section III reviews related work. Our dIBP-based doubly sparse nonparametric NMF framework is proposed with four implementations in Section IV, and Gibbs samplers are designed for the four models in Section V. A set of experiments on synthetic data and real-world tasks are conducted in Section VI. Finally, Section VII concludes the study and discusses the further work.

II. Preliminary Knowledge

A. Sparse Nonnegative Matrix Factorization

Given a nonnegative matrix $Y_{M\times N}$ (extended to seminonnegative in [17]), SNMF aims to find two matrices $A_{M\times K}$ and $X_{N\times K}$ to minimize the following cost function:

$$J = \left\| Y_{M\times N} - A_{M\times K} X_{N\times K}^{T} \right\|_F^2 + \|A_{M\times K}\|_1 + \|X_{N\times K}\|_1 \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm, $\|\cdot\|_1$ is the ℓ_1 norm, and the elements of A and X have nonnegative constraint. The ℓ_1 norm in the cost function is used for the sparsity constraint.

TABLE I

Notations in This Paper

Symbol	meaning in this paper
M	the row number of Y
N	the column number of Y
K	the latent factor number
$K^{\dagger}$	the truncation level
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 with size $M \times K$
$Z^{(2)}$	mask matrix for X with size $N \times K$
$z_{m,k}^{(1)}$	the mask binary variable for the element of A at m row and k column
$z_{n,k}^{(2)}$	the mask binary variable for the element of X at n row and k column
$V^{(1)}$	loading matrix for A with size $M \times K$
$V^{(2)}$	loading matrix for X with size $N \times K$
$v_{m,k}^{(1)}$	the loading variable for the element of A at m row and k column
$v_{n,k}^{(2)}$	the loading variable for the element of X at n row and k column
$\mu_k^{(1)}$	k-th stick weight of IBP for A
$\mu_k^{(2)}$	k-th stick weight of IBP for X
θ	parameters for a bivariate Beta distribution or a copula
$corr$	correlation value

Several important notations used throughout this paper are summarized in Table I.

Take the document-word co-clustering task as an example. The input $Y_{M\times N}$ denotes the frequency of N words in M documents. $A_{M\times K}$ denotes the documents' interests in K factors (i.e., topics), and $X_{N\times K}$ denotes the words' interests in K factors (i.e., topics). All documents and words are projected into the same K-dimensional space by NMF. Based on $A_{M\times K}$ and $X_{N\times K}$, the document-word co-clustering task can be accomplished.

One problem of NMF is that the dimensionality of the factor matrices K needs to be predefined. Normally, this variable is experimentally adjusted within a range. This paper will resolve this problem using Bayesian nonparametric learning.

B. Indian Buffet Process

The IBP [13], [18] is defined as a prior for the binary matrices with an infinite number of columns. A stick-breaking construction for IBP [19] is proposed as follows:

$$\nu_j \sim \text{Beta}(\alpha, 1), \quad \mu_k = \prod_{j=1}^{k} \nu_j, \quad z_{n,k} \sim Ber(\mu_k) \tag{2}$$

where Ber() denotes Bernoulli distribution, $z_{n,k}$ forms a matrix $Z_{N\times K}$, $\{\nu_j\}$ is a set of variables with a Beta distribution, α is the parameter of the Beta distribution, and μ_k is the stick weight of column k. The bigger μ_k is, the more "ones" appear in column k of the binary matrix $Z_{N\times K}$. K is determined by the data and the parameter α of IBP.

III. Related Work

The related work mainly falls into two categories: one concerns research on nonparametric nonnegative matrix factorization and the other concerns the IBP. These state-of-the-art studies inspire our idea of using a dIBP for Bayesian nonparametric nonnegative matrix factorization.

A. Nonparametric NMF

The types of research on nonnegative matrix factorization include supervised or semisupervised extension [17], [20], the convergence rate [21], sparse [22], Bayesian [23], nonparametric [24], and more. Bayesian extension of NMF aims to model the NMF using the probabilistic distributions. In one example, latent Dirichlet allocation (LDA) [25] and correlated LDA [26] ideas are transferred to the Bayesian parametric NMF and correlated NMF [23]. These extensions are still parametric, however, which means that the dimensionality still needs to be predefined. The nonparametric extension of NMF mainly relies on the machinery of stochastic processes, and there are two categories that principally use stochastic processes to build Bayesian nonparametric NMF.

One category of models decomposes the data matrix $Y = Z \odot A$ into a binary (mask) matrix Z and a factor matrix A, and binary (mask) matrix is given an infinite prior [27], i.e., IBP [13], [14]. The binary matrix Z functions as the feature selection matrix and the other factor matrix A functions as a feature matrix. The dimensionality of A will change with the change of Z. Another similar approach is to use the Beta process [28], in which the data matrix $Y = (S \odot Z)A$ is decomposed into three matrices, where S is an ordinary nonnegative matrix and Z is modeled by a Bernoulli process, which is in turn given a Beta process prior [24]. Due to the relationship between IBP and the Beta-Bernoulli process [28], this approach is similar to the model using IBP. However, there is only one (mask) matrix controlled to have nonparametricity and sparsity properties, which is more like a factor analysis than NMF.

The other category of research assigns the data matrix a Poisson (likelihood) distribution as $Y_{m,n} \sim \text{Poi}(\sum_k r_k \phi_{m,k} \varphi_{n,k})$ (named Gamma–Poisson NMF [29]), where r is given an infinite prior, i.e., Gamma process [30]. ϕ and φ, which account for two factor matrices from NMF, respectively, are normal nonnegative matrices. The nonparametricity of the model mainly depends on the Gamma process r. The innovative idea of this model is to control the two factor matrices through the coefficient of Poisson data likelihood. This idea is extended by replacing the Gamma–Poisson process with the Beta-negative binomial process, which allows overdispersion for the count data (matrix) [31]. Since the Beta-negative binomial process is (in distribution) equal to the Beta–Gamma–Gamma–Poisson process, the idea in [31] about nonparametricity control is the same. This kind of method has been applied for assortative network modeling [32], joint document and network modeling [33], and discovering words in spoken utterances [34]. However, although the sparsity of observation is considered [29], the sparsity of the factor matrices, i.e., ϕ and φ, is overlooked, so this idea cannot be used for sparse NMF. Apart from model design, there are also researchers who are focusing on the inference methods for nonparametric NMF models, such as Power-EP [14], stochastic structured mean-field variational inference [35], and particle filtering [36]. Both of these are specially designed for the first category of nonparametric NMF.

To summarize, although there are a number of preliminary studies on the nonparametric NMF problem, the current state-of-the-art works cannot be directly used for our target: NMF with both *double sparsity* and *nonparametricity*.

B. IBP and dIBP

IBP is proposed in [13] and [18], and is a marginalization of the Beta-Bernoulli process. Its widespread popularity is due to its power to generate a binary matrix with infinite columns. Restricted IBP [37] is proposed to allow an arbitrary prior distribution rather than a fixed Poisson distribution form for the number of features in each observation; integrative IBP [38] is developed for integrating multimodal data in a latent space; Gibbs-type IBP [39] is a generalization of IBP with two-parameter IBP and three-parameter IBP [40] as special cases. All these models provide extensions for the original IBP to endow it with new meaningful features, e.g., power-law behavior. This paper focuses on the dIBP for NMF problem, so the original IBP is used here; the above-mentioned works could be considered as interesting extensions of this paper in the future.

The dIBP was first proposed in [15] based on the Gaussian process (GP), and can be used for the nonparametric NMF after being embedded in our proposed framework, discussed in Section IV. GP-based dIBP models the dependence between IBP with different covariants, such as the time tags of documents, geographic locations of people, or GDPs of countries. The idea of dependent nonparametric processes was first proposed by MacEachern [41], and seven different classes of dependence are summarized [42]. The hierarchial Beta process is proposed, so that the different Beta processes share a common base discrete measure [16]; it can also be seen as an implicit realization of dIBP, which is used for nonparametric NMF in Section IV. The phylogenetic IBP [43] considers the tree structure of the data points, which can be seen as a supervised IBP. A coupled IBP [44] is proposed for collaborative filtering, which links two IBPs through a factor matrix. However, this coupling does not guarantee that the factor matrices have the same dimensionality, which is important for the NMF. To summarize, there is no existing work on using or constructing dIBP for nonparametric NMF.

IV. Doubly Sparse Nonparametric NMF Framework and its Three Implementations

In this section, we propose a doubly sparse nonparametric NMF framework followed by its four implementations, and we then discuss the differences between these implementations and analyze the properties of models.

A. Doubly Sparse Nonparametric NMF Framework

In our doubly sparse nonparametric NMF framework, the data matrix is modeled as

$$Y = AX^T = (V^{(1)} \odot Z^{(1)})(V^{(2)} \odot Z^{(2)})^T \tag{3}$$

where $Z^{(1)}$ and $Z^{(2)}$ are two binary matrices, $V^{(1)}$ and $V^{(2)}$ are loading matrices, and $\odot$ denotes the Hadamard product. Next, we assign these latent variable probability distributions as follows:

$$\begin{aligned} a_{m,k} &= v^{(1)}_{m,k} z^{(1)}_{m,k}, \quad v^{(1)}_{m,k} \sim \text{Gam}(1, \tau_1) \\ x_{n,k} &= v^{(2)}_{n,k} z^{(2)}_{n,k}, \quad v^{(2)}_{n,k} \sim \text{Gam}(1, \tau_2) \end{aligned} \tag{4}$$

where $\text{Gam}(\cdot, \cdot)$ denotes the Gamma distribution and τ_1 and τ_2 are two parameters. It is evident that $a_{m,k} \geq 0$ and $x_{n,k} \geq 0$ are always satisfied under the above probability distributions. With A and X determined, the likelihood of the model is defined as

$$y_{m,n} | a_{m,k}, x_{n,k} \sim \text{Exp}\left(y_{m,n}; \sum_k a_{m,k} x_{n,k} + \epsilon \right) \tag{5}$$

where $\text{Exp}(\cdot)$ denotes the exponential distribution and ϵ is a very small positive number to make the parameter of exponential distribution greater than zero. The selection of the exponential distribution is used to guarantee each element of Y with support $[0, +\infty)$. The above-mentioned parameter setting is used to retain the desired expectations of these distributions, i.e., the expectation of $y_{m,n}$ is $\sum_k a_{m,k} x_{n,k} + \epsilon$.

At the point, we have completed the construction of the doubly sparse nonparametric NMF framework except for an appropriate prior for $Z^{(1)}$ and $Z^{(2)}$. In the following subsections (B, C, D and E), we will introduce four implementations of this framework using four kinds of dIBP.

B. Implementation by Bivariate Beta Distribution-Based DIBP

There are different ways to link two IBPs. Here, we propose a method that links the initials of two IBPs, i.e., ν. In the original IBP, ν satisfies a Beta distribution with parameter $(\alpha, 1)$ as in (2). Therefore, our innovative idea is to find 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, an intuitive candidate would be Dirichlet distribution. There is a strictly negative relation, i.e., $\nu^{(1)} + \nu^{(2)} = 1$, 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 [45] is adopted here whose probability density function is

$$p(r_1, r_2) = \frac{r_1^{a^0-1} r_2^{b^0-1} (1-r_1)^{b^0+c^0-1} (1-r_2)^{a^0+c^0-1}}{B(a^0, b^0, c^0)(1 - r_1 r_2)^{a^0+b^0+c^0}}$$
$$\text{s.t.,} \quad 0 \leq r_1, r_2 \leq 1, \quad a^0, b^0, c^0 > 0 \tag{6}$$

where $a^0, b^0, and c^0$ are three parameters of this distribution and $B(a^0, b^0, c^0)$ is the normalization constant that is difficult to evaluate. One merit of this bivariate Beta distribution is that two marginal distributions are

$$r_1 \sim \text{Beta}(a^0, c^0), \quad r_2 \sim \text{Beta}(b^0, c^0) \tag{7}$$

which satisfy our requirement concerning Beta marginal distributions of $\nu^{(1)}$ and $\nu^{(2)}$. Another merit is that this distribution can model positive correlation between (r_1, r_2) with the range $[0, 1]$ adjusted by the three parameters (a^0, b^0, c^0) compared with the strictly negative correlation from the Dirichlet distribution. Here, we set c^0 of the bivariate Beta distribution to 1, because we must ensure that the marginal distribution of each ν is a Beta distribution with parameter form $(\alpha, 1)$. This condition is necessary for the distribution of the generated binary matrices that satisfy the IBPs. Considering (7), we give c^0 a fixed value. Even with a fixed value for c^0, the bivariate Beta distribution in (6) is still able to model different correlations of two variables. The correlation can be expressed as a function of a^0 and b^0 when $c^0 = 1$

$$\begin{aligned} \text{corr} = &\sqrt{a^0 b^0 (a^0+2)(b^0+2)} \\ &\times \left[\frac{\Gamma(a+2)\Gamma(b+2)}{(a+b+1)\Gamma(a+b+2)} \mathbb{F} - 1 \right]. \end{aligned} \tag{8}$$

where

$$\begin{aligned} \mathbb{F} = {}_3F_2(&a^0+1, b^0+1, a^0+b^0+1; \\ &a^0+b^0+2, a^0+b^0+2; 1) \end{aligned} \tag{9}$$

is a hypergeometric function[1] that can be evaluated given the parameters. The derivative detail of this equation is given in the appendix. For example, if $a^0 = 2.5$ and $b^0 = 4$, the correlation between r_1 and r_2 is 0.995; if $a^0 = 0.05$ and $b^0 = 0.1$, the correlation between r_1 and r_2 is 0.080.

With the desired bivariate Beta distribution in hand, we build the BiBeta-based dIBP as follows:

$$\begin{aligned} \left(\nu_k^{(1)}, \nu_k^{(2)}\right) &\sim bi\text{Beta}(\theta), \quad \theta : \{a^0, b^0, c^0 = 1\} > 0 \\ z^{(1)}_{n,k} &\overset{\text{i.i.d}}{\sim} \text{Ber}\left(\mu_k^{(1)}\right), \quad \mu_k^{(1)} = \prod_{j=1}^{k} \nu_j^{(1)} \\ z^{(2)}_{n,k} &\overset{\text{i.i.d}}{\sim} \text{Ber}\left(\mu_k^{(2)}\right), \quad \mu_k^{(2)} = \prod_{j=1}^{k} \nu_j^{(2)} \end{aligned} \tag{10}$$

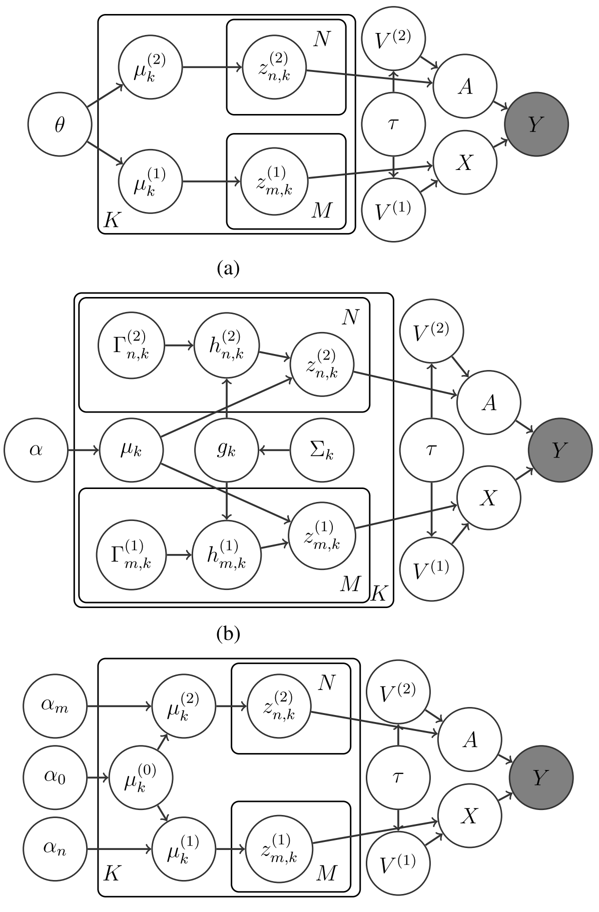
where $biBeta(\theta)$ denotes the bivariate Beta distribution in (6) and θ denotes the parameters of the distribution. The graphical model is shown in Fig. 1(a).

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. Next, we use the copula to capture more freedom of their relations.

C. Implementation by Copula-Based DIBP

Copula [46] links two variables with given marginal distributions, and is used to define a joint distribution for variables with known marginal distributions in statistics. Here, we use

[1] https://en.wikipedia.org/wiki/Generalized_hypergeometric_function



(a)

(b)

(c)

Fig. 1. Graphical models for (a) bivariate Beta distribution-based or copula-based dIBPs, (b) GP-based dIBPs, and (c) HBP-based dIBPs.

the Farlie–Gumbel–Morgenstern (FGM) copula [47] as an example. The definition of the FGM copula is

$$\begin{aligned} C_\rho(o_1, o_2) &= o_1 o_2 + \rho o_1 o_2 (1 - o_1)(1 - o_2) \\ c_\rho(o_1, o_2) &= 1 + \rho(2o_1 - 1)(2o_2 - 1) \end{aligned} \tag{11}$$

where $C_\rho(o_1, o_2)$ is the joint cumulative distribution function (cdf), $c_\rho(o_1, o_2)$ is the joint probability density function, $\rho \in [-1, 1]$ is the parameter of the FGM copula, and o_1 and o_2 are the cdf values of two marginal distributions that are known in advance.

For our problem, the two variables are $\nu^{(1)}$ and $\nu^{(2)}$ and their marginal distributions are Beta distributions with parameters $(\alpha_1, 1)$ and $(\alpha_2, 1)$. Therefore, o_1 and o_2 are defined as

$$\begin{aligned} o_1 = F^{\text{beta}}(\nu^{(1)}) \sim (\nu^{(1)})^{\alpha_1}, \quad f^{\text{beta}}(\nu^{(1)}) \sim \alpha_1 (\nu^{(1)})^{\alpha_1 - 1} \\ o_2 = F^{\text{beta}}(\nu^{(2)}) \sim (\nu^{(2)})^{\alpha_2}, \quad f^{\text{beta}}(\nu^{(2)}) \sim \alpha_2 (\nu^{(2)})^{\alpha_2 - 1} \end{aligned} \tag{12}$$

where $F^{\text{beta}}(\nu^{(1)})$ and $f^{\text{beta}}(\nu^{(1)})$ represent the cdf and probabilistic density function of $\nu^{(1)}$.

In copula, the correlation between $\nu^{(1)}$ and $\nu^{(2)}$ is modeled or reflected by the value of ρ. Various correlations (i.e., positive, null, or negative) can be captured by the different values of ρ. The Spearman correlation can be evaluated by corr $= (\rho/3)$. Since the support of ρ is $[-1, 1]$, the correlation range that can be modeled by the FGM copula is $[-(1/3), (1/3)]$. In particular, there is no correlation if $\rho = 0$.

Copula-based dIBP is defined by replacing the bivariate Beta distribution in (10) with joint distribution defined by the FGM copula

$$\left(\nu_k^{(1)}, \nu_k^{(2)}\right) = c_\rho(\theta), \quad \theta : \{\rho \in [-1, 1], \alpha_1 > 0, \alpha_2 > 0\}. \tag{13}$$

The graphical model is the same as the BiBeta-based dIBP in Fig. 1(a) but with different parameters θ.

There are a number of works concerning the constructing of dependent random measures through the Levy copula [48], [49], such as the Levy copula-based dependent Dirichlet process [50] and the Levy copula-based dependent Poisson-Dirichlet process [51]. Levy copula-based-dependent random measures have an advantage that the marginal random measures do not have to be of the same type of process [48]. For example, a Beta process could be linked with a Gamma process. The Levy copula is mainly used to link two Levy processes, but IBP is not a Levy process although its de Finetti mixing distribution (i.e., Beta process) is. The Copula used in this paper is an ordinary one that is used for linking two random variables with fixed marginal distributions (Beta distributions in this paper).

D. Implementation by GP-Based DIBP

The first dIBP is proposed based on Gaussian process [15]. In this GP-based dIBP, each stick weight μ k is used to generated a different number of columns of binary matrices. In other words, the GP-based dIBP uses the same set of sticks for different binary matrices. We can use this GP-based dIBP for the NMF as the prior for the matrices $Z^{(1)}$ and $Z^{(2)}$. The graphical model is shown in Fig. 1(b), and the generative process is as follows:

$$\nu_j \sim Beta(\alpha, 1), \quad \mu_k = \prod_{j=1}^{k} \nu_j \tag{14}$$

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

$$\begin{aligned} g_k &\sim \text{GP}(0, \Sigma_k) \\ h_{m,k}^{(1)} &\sim \text{GP}\big(g_k, \Gamma_{m,k}^{(1)}\big) \\ h_{n,k}^{(2)} &\sim \text{GP}\big(g_k, \Gamma_{n,k}^{(2)}\big) \\ \Sigma_k(t, t') &= \sigma^2 \exp\left(-\frac{(t - t')^2}{s^2}\right) \\ \Gamma^{(1)} &= \Gamma^{(2)} = \eta^2 I \\ z_{m,k}^{(1)} &= \delta\big\{h_{m,k}^{(1)} < F^{-1}\big(\mu_k | 0, (\Sigma_k)^{(1,1)} + \big(\Gamma_{m,k}^{(1)}\big)^{(1,1)}\big)\big\} \\ z_{n,k}^{(2)} &= \delta\big\{h_{n,k}^{(2)} < F^{-1}\big(\mu_k | 0, (\Sigma_k)^{(2,2)} + \big(\Gamma_{n,k}^{(2)}\big)^{(2,2)}\big)\big\} \end{aligned}$$

where $\text{GP}(0, \Sigma_k)$ denotes a GP parameterized by a mean function 0 and a kernel function Σ_k. g_k is a random draw from $\text{GP}(0, \Sigma_k)$, which is in turn set as the mean function of $\text{GP}(g_k, \Gamma_{m,k}^{(1)})$ and $\text{GP}(g_k, \Gamma_{n,k}^{(2)})$, so there is a hierarchy between different GPs where the correlation is captured. $\Gamma_{m,k}^{(1)}$

and $\Gamma_{n,k}^{(2)}$ are identity matrices with a parameter η. $h_{m,k}^{(1)}$ and $h_{n,k}^{(2)}$ are two random draws from the corresponding GPs. $F^{-1}(\cdot)$ is the inverse normal cdf , and $\delta\{\cdot\}$ is the indicator function. More details can be found in [15]. Since there are only two binary matrices, the GP is degenerated to a 2-D Gaussian distribution, and Σ_k is equal to a 2×2 matrix.

E. Implementation by HBP-Based DIBP

Another possible candidate for the dIBP construction is the hierarchial Beta process (HBP) [16]. The graphical model of HBP-dIBP is shown in Fig. 1(c) and the generative process is as follows:

$$B = \sum_{k=1}^{\infty} \mu_k^{(0)} \delta_{\theta_k} \sim \mathrm{BP}(\alpha_0, B_0)$$

$$B^{(1)} = \sum_{k=1}^{\infty} \mu_k^{(1)} \delta_{\theta_k} \sim \mathrm{BP}(\alpha_m, B), \quad Z^{(1)} \sim \mathrm{BeP}(B^{(1)})$$

$$B^{(2)} = \sum_{k=1}^{\infty} \mu_k^{(2)} \delta_{\theta_k} \sim \mathrm{BP}(\alpha_n, B), \quad Z^{(2)} \sim \mathrm{BeP}(B^{(2)})$$

where $\mathrm{BP}(\alpha, B_0)$ denotes a Beta process parameterized by a base measure B_0 and a concentration parameter α_0; $\mathrm{BeP}(B^{(1)})$ denotes a Bernoulli process parameterized by a base measure $B^{(1)}$. In HBP, B, which is a random draw from $\mathrm{BP}(\alpha_0, B_0)$, is set as the base measure of $\mathrm{BP}(\alpha_m, B)$ and $\mathrm{BP}(\alpha_n, B)$. Strictly speaking, the above-mentioned model is not dependent on a dIBP, because no explicit dIBP is built in this model. As proven in [16] and [18], when $\widetilde{Z} \sim \mathrm{BeP}(\widetilde{B})$ and $\widetilde{B} \sim \mathrm{BP}(1, \widetilde{B_0})$ where the total mass of $\widetilde{B_0}$ is $\widetilde{\alpha}$, the marginal distribution of $\widetilde{Z}$ with $\widetilde{B}$ integrated out is $\mathrm{IBP}(\widetilde{\alpha})$. However, the marginal distribution of $B^{(1)}$ or $B^{(2)}$ with B integrated out is not necessarily a BP (not to mention with the specific parameters, i.e., 1 and $\widetilde{B_0}$), so we cannot draw the conclusion that $Z^{(1)}$ and $Z^{(2)}$ are with IBP marginal distributions. Although there is no dIBP explicitly built in this model, it can still be seen as a prior for the $Z^{(1)}$ and $Z^{(2)}$. Note that the stick weights $\mu_k^{(1)}$ and $\mu_k^{(2)}$ for two IBPs both center on $\mu_k^{(0)}$. This means that there are the same expected nonzero items in $Z^{(1)}$ and $Z^{(2)}$ across all columns, which is too restrictive. Take document-word co-clustering as an example. This restriction means that the nonzero items of documents and words tend to have a similar number on all factors/topics. It is more reasonable to let documents and words have flexible behavior/'interests' on the different factors/topics.

F. Discussion on Models

One apparent advantage of the bivariate Beta distribution-based and copula-based dIBP compared with the GP-based dIBP is that fewer latent variables are involved. This can easily be observed in the graphical models in Fig. 1. More latent variables tend to slow down the convergence of the model inference. The advantage of Copula compared with the bivariate Beta distribution is that we can easily obtain both the cdf and probability density function of $(\nu_k^{(1)}, \nu_k^{(2)})$. This will impact on the model inference, which will be discussed later.

Another advantage of the proposed dIBPs is their greater flexibility. GP-based dIBP works in the following fashion: let $Z_j^{(1)}$ and $Z_j^{(2)}$ be the jth corresponding columns of two identical sized binary matrices $Z^{(1)}$ and $Z^{(2)}$. The model assumes that $Z_j^{(1)}$ and $Z_j^{(2)}$ share the same stick weight μ_j, making the expectation of the number of nonzero 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 either one or zero. HBP-based dIBP is similar to GP-based dIBP, which also makes the expectation of the number of nonzero entries identical for both $Z_j^{(1)}$ and $Z_j^{(2)}$ through a hierarchal structure. The two parameters α_m and α_n are instead used to control the respective variance from the expectation for two matrices. Although both models could be used to implement a doubly sparse nonparametric NMF, it nonetheless is inadequate in many matrix factorization scenarios as the assumption that the total number of nonzero entries is (in expectation) distributed identically does not hold universally. Take document-word co-clustering as an example. The number of nonzero factors of a document column may be drastically different from the number of nonzero factors of a corresponding Word column in a document-word matrix. Contrarily, there is no restriction on the expectations of μ_1 and μ_2 for two binary matrices $Z^{(1)}$ and $Z^{(2)}$ in the proposed bivariate Beta distribution-based dIBP and Copula-based dIBP, which allows both $Z_j^{(1)}$ and $Z_j^{(2)}$ to have much greater flexibility and variation in terms of their nonzero entries.

What does the correlation between $\mu^{(1)}$ and $\mu^{(2)}$ reveal from the data? We explain its meaning using the document-word matrix as an example again. We obtain two factor matrices $A_{M \times K}$ and $X_{N \times K}$ after factorization. The K hidden factors are also named *topics* in some studies. We know that $\mu_k^{(1)}$ and $\mu_k^{(2)}$ account for the generation of $Z_k^{(1)}$ and $Z_k^{(2)}$, which are the kth column of $Z_{M \times K}^{(1)}$ and $Z_{N \times K}^{(2)}$. If $\mu_k^{(1)}$ is larger, the number of 'ones' in $Z_k^{(1)}$ is greater; the same holds for $\mu_k^{(2)}$ and $Z_k^{(2)}$. When there is a large negative correlation between $\mu_k^{(1)}$ and $\mu_k^{(2)}$, the number of 'ones' in $Z_k^{(2)}$ will be smaller if the number of 'ones' in $Z_k^{(1)}$ is larger. This means that the commonly used topics in documents (the number of 'ones' in $Z_k^{(1)}$ is large) tend to be *focused topics* which only focus on infrequent words (the number of 'ones' in $Z_k^{(2)}$ is small), as illustrated by the red/solid curve/distribution on words in Fig. 2. In contrast, when there is a large positive correlation between $\mu_k^{(1)}$ and $\mu_k^{(2)}$, the number of "ones" in $Z_k^{(2)}$ will be larger if the number of "ones" in $Z_k^{(1)}$ is also larger. This means that the commonly used topics in documents (the number of "ones" in $Z_k^{(1)}$ is large) tend to be *non-focused topics* which have many words (the number of "ones" in $Z_k^{(2)}$ is also large), as illustrated by the blue/dashed curve/distribution on words in Fig. 2. Therefore, we can conclude that the correlation between $\mu^{(1)}$ and $\mu^{(2)}$ can be seen as the measurement of the *focus degree* of commonly used hidden factors/topics in

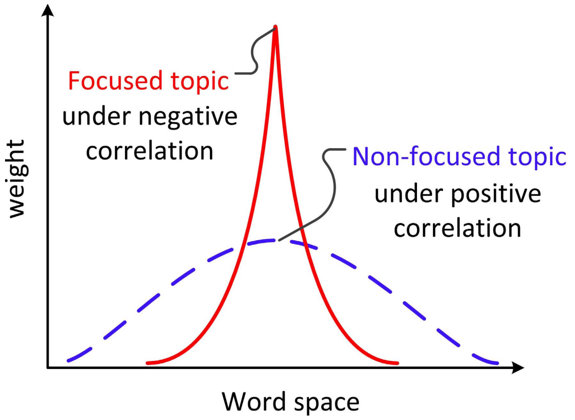


Fig. 2. Ilustration of the meaning of learned correlation. Red solid curve: distribution on words from a large negative correlation. Blue dashed curve: nonfocused distribution on words from a large positive correlation.

the data set. Note that *focused* here is different from the one used in [52], where *focused* means each document could flexibly select their own topics rather than rigidly following the correlation defined in HDP [53], but it is here to highlight the topic with limited number of words with dominant weights.

V. Model Inference

The objective of this section with data matrix Y is to estimate the hidden variables by a properly designed MCMC inference algorithm 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 Bayesian nonparametric learning is to use a truncation method. The truncation method, which uses a relatively big $K^\dagger$ as the (potential) maximum number of factors, is widely accepted. We want to highlight that GP-based and HBP-based dIBPs are not our contribution. Our contribution is to link the GP-based and HBP-based dIBPs with the nonnegative matrix factorization likelihood. The inference of these models is given in the appendix for the self-contained purpose.

A. Update Stick Weights μ

When updating μ_k, we need to find its conditional distribution given μ_{k-1} and μ_{k+1}, because the order of μ must be maintained to make the marginal distributions of Z satisfy two IBPs. Based on the M–H sampler, the acceptance ratio of a new sample for $\mu_k = [\mu_k^{(1)}, \mu_k^{(2)}]$ is

$$\min\left(1, \frac{p(Z|\mu_k^*)\,p(\mu_k^*|\mu_{k+1}, \mu_{k-1})}{p(Z|\mu_k)\,p(\mu_k|\mu_{k+1}, \mu_{k-1})} \times \frac{q(\mu_k)}{q(\mu_k^*)}\right) \quad (15)$$

where $q(\cdot)$ is a proposal distribution which will be explained later, μ_k^* is a new sample drawn from the proposal distribution $q(\cdot)$, $p(Z|\mu_k)$ is the likelihood of μ_k to generate the kth column of binary matrix Z as in (10), and the $p(\mu_k^*|\mu_{k+1}, \mu_{k-1})$ is the conditional probability density function of μ_k^* within the range of $[\mu_{k+1}, \mu_{k-1}]$, which will be different when different strategies in Section IV are used to link $\mu_k^{(1)}$ and $\mu_k^{(2)}$.

Next, we derive the conditional probability density function of $p(\mu_k|\mu_{k+1}, \mu_{k-1})$ as follows: for the first column

$$\mu_1^{(1)} = \nu_1^{(1)}, \quad \mu_1^{(2)} = \nu_1^{(2)} \quad (16)$$

and

$$p\left(\mu_1^{(1)}, \mu_1^{(2)}\right) = p\left(\nu_1^{(1)}, \nu_1^{(2)}\right) \quad (17)$$

where $p(\cdot, \cdot)$ is the joint probability density function given by (6) or (11) from two proposed dIBPs. For the second column

$$\mu_2^{(1)} = \nu_2^{(1)}\mu_1^{(1)}, \quad \mu_2^{(2)} = \nu_2^{(2)}\mu_1^{(2)} \quad (18)$$

and

$$p\left(\mu_2^{(1)}, \mu_2^{(2)}\right) = \frac{p\left(\mu_2^{(1)}/\mu_1^{(1)}, \mu_2^{(2)}/\mu_1^{(2)}\right)}{\mu_1^{(1)}\mu_1^{(2)}} \quad (19)$$

where J_1 is the Jacobian matrix.

For the kth column

$$\mu_k^{(1)} = \nu_k^{(1)}\mu_{k-1}^{(1)}, \quad \mu_k^{(2)} = \nu_k^{(2)}\mu_{k-1}^{(2)} \quad (20)$$

and

$$p\left(\mu_k^{(1)}, \mu_k^{(2)}\right) = \frac{p\left(\mu_k^{(1)}/\mu_{k-1}^{(1)}, \mu_k^{(2)}/\mu_{k-1}^{(2)}\right)}{\mu_{k-1}^{(1)}\mu_{k-1}^{(2)}}. \quad (21)$$

To summarize, the conditional density of μ_k is

$$(\mu_k|\mu_{k-1}, \mu_{k+1}) \propto \frac{p\left(\mu_k^{(1)}/\mu_{k-1}^{(1)}, \mu_k^{(2)}/\mu_{k-1}^{(2)}\right)}{\mu_{k-1}^{(1)}\mu_{k-1}^{(2)}} \frac{p\left(\mu_{k+1}^{(1)}/\mu_k^{(1)}, \mu_{k+1}^{(2)}/\mu_k^{(2)}\right)}{\mu_k^{(1)}\mu_k^{(2)}}. \quad (22)$$

Considering the support of μ_k, the proposal distribution, $q(\cdot)$, is set as the product of two independent truncated Beta distributions: $\mu_k^{(1)} \sim \text{Beta}((\alpha_1/K^\dagger), 1)$, $\mu_k^{(1)} \in [\mu_{k+1}^{(1)}, \mu_{k-1}^{(1)}]$ and $\mu_k^{(2)} \sim \text{Beta}((\alpha_2/K^\dagger), 1)$, $\mu_k^{(2)} \in [\mu_{k+1}^{(2)}, \mu_{k-1}^{(2)}]$.

It is quite easy to sample the truncated Beta distribution, because Beta distribution is a standard distribution.

B. Update Binary Matrices Z

Two binary matrices, $Z : \{Z^{(1)}, Z^{(2)}\}$, can be updated separately. Each element in the two matrices satisfies a Bernoulli distribution with the following conditional posterior probabilities:

$$p\left(z_{m,k}^{(1)} = 1\right) \propto \mu_k^{(1)} \prod_n e^{-y_{m,n}\left(\sum_l v_{m,l}^{(1)} z_{m,l}^{(1)} v_{n,l}^{(2)} z_{n,l}^{(2)} + \epsilon\right)}$$

$$p\left(z_{m,k}^{(1)} = 0\right) \propto (1 - \mu_k^{(1)}) \prod_n e^{-y_{m,n}\left(\sum_l v_{m,l}^{(1)} z_{m,l}^{(1)} v_{n,l}^{(2)} z_{n,l}^{(2)} + \epsilon\right)} \quad (23)$$

and

$$p\left(z_{n,k}^{(2)} = 1\right) \propto \mu_k^{(2)} \prod_m e^{-y_{m,n}\left(\sum_l v_{m,l}^{(1)} z_{m,l}^{(1)} v_{n,l}^{(2)} z_{n,l}^{(2)} + \epsilon\right)}$$

$$p\left(z_{n,k}^{(2)} = 0\right) \propto (1 - \mu_k^{(2)}) \prod_m e^{-y_{m,n}\left(\sum_l v_{m,l}^{(1)} z_{m,l}^{(1)} v_{n,l}^{(2)} z_{n,l}^{(2)} + \epsilon\right)} \quad (24)$$

where ϵ is a small positive number. Since elements of Z having discrete distribution, it is easy to obtain samples.

C. Update Loading Matrices V

Since the prior for V is Gamma distribution and the likelihood is exponential distribution, the conditional distribution for V is

$$p\left(v^{(1)}_{m,k}|\cdots\right) \propto e^{v^{(1)}_{m,k}\tau_1} \prod_n e^{-y_{m,n}\left(\sum_k v^{(1)}_{m,k} z^{(1)}_{m,k} v^{(2)}_{n,k} z^{(2)}_{n,k} + \epsilon\right)} \tag{25}$$

and

$$p\left(v^{(2)}_{n,k}|\cdots\right) \propto e^{v^{(2)}_{n,k}\tau_2} \prod_m e^{-y_{m,n}\left(\sum_l v^{(1)}_{m,l} z^{(1)}_{m,l} v^{(2)}_{n,l} z^{(2)}_{n,l} + \epsilon\right)} \tag{26}$$

Due to the existence of the ϵ, the posterior of V is not a Gamma distribution, so we have to use the M–H sampler to obtain its samples.

D. Update Model Parameter θ

The graphical model in Fig. 1(c) has the parameter θ. 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.

1) Bivariate Beta Distribution: The parameters of bivariate Beta distribution, $\theta : \{a^0, b^0\}$, are given two Gamma priors. The conditional distributions are

$$p([a^0\ b^0]) \propto \mathrm{Gam}([a^0\ b^0]; hp) \prod_{k=1}^{K} p\left(\mu^{(1)}_k, \mu^{(2)}_k | a^0, b^0\right) \tag{27}$$

where hp is the hyperparameter of the prior for a^0 and b^0 and K is the number of active columns of Z. The "active" column means that there is at least one element with 1 in that column.

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

$$p([\alpha_1\ \alpha_2]) \propto \mathrm{Gam}([\alpha_1\ \alpha_2]; hp) \prod_{k=1}^{K} c\left(\mu^{(1)}_k, \mu^{(2)}_k\right) \tag{28}$$

where $c(\cdot,\cdot)$ is the copula density in (11). We give the ρ of the FGM copula a uniform distribution on its support $[-1, 1]$, and its posterior is

$$p(\rho|\cdots) \propto \prod_{k=1}^{K} c\left(\mu^{(1)}_k, \mu^{(2)}_k | \rho\right). \tag{29}$$

After introducing update methods for all the latent variables, we summarize the inference (i.e., Gibbs sampler) for the four models in Algorithm 1 for the bivariate Beta distribution-based dIBP-NMF (BB-dIBP-NMF) model, Algorithm 2 for the Copula-based dIBP-NMF (C-dIBP-NMF) model, Algorithm 3 for the GP-based dIBP-NMF (GP-dIBP-NMF) model, and Algorithm 4 for the HBP-based dIBP-NMF (HBP-dIBP-NMF) model.

Algorithm 1 Gibbs Sampler for BB-dIBP-NMF

```
Input: $Y$
Output: $A$, $X$
initialization;
while $i \le max_{iter}$ do
    // latent variables of dIBP
    Update $\mu$ by Eq. (15);
    Update $Z$ by Eq. (23) and (24);
    Update $a^0$ and $b^0$ by Eq. (27);
    // latent variables of NMF
    Update $V$ by Eq. (25) and (26);
    $i++$;
return $A$ and $X$;
```

Algorithm 2 Gibbs Sampler for C-dIBP-NMF

```
Input: $Y$
Output: $A$, $X$
initialization;
while $i \le max_{iter}$ do
    // latent variables of dIBP
    Update $\mu$ by Eq. (15);
    Update $Z$ by Eq. (23) and (24);
    Update $\alpha_1$ and $\alpha_2$ by Eq. (28);
    Update $\rho$ by Eq. (29);
    // latent variables of NMF
    Update $V$ by Eq. (25) and (26);
    $i++$;
return $A$ and $X$;
```

Algorithm 3 Gibbs Sampler for GP-dIBP-NMF

```
Input: $Y$
Output: $A$, $X$
initialization;
while $i \le max_{iter}$ do
    // latent variables of dIBP
    Update $\mu$ by Eq. (33);
    Update $Z$ by Eq. (39) and (40);
    Update $g$ by Eq. (35);
    Update $h$ by Eq. (36);
    Update $s$ by Eq. (38);
    // latent variables of NMF
    Update $V$ by Eq. (25) and (26);
    $i++$;
return $A$ and $X$;
```

E. Computational Complexity Analysis

The Gibbs sampling updates of BB-dIBP-NMF can be calculated in $O((M+N)(K^\dagger)^2 + (M+N+3)K^\dagger)$ time, where $K^\dagger$ is the truncation level. Note that, $O((M+N)K^\dagger)$ accounts for the latent variable of NMF that is shared across four implementations, and $O(3K^\dagger + (M+N)(K^\dagger)^2)$ accounts for BB-dIBP. When $K^\dagger$ is set relatively large in comparison to the scale of data matrix (i.e., M and N), the cost of BB-dIBP is primarily of the order of $O((M+N)(K^\dagger)^2)$. We do not consider the cost from the latent

Algorithm 4 Gibbs Sampler for HBP-dIBP-NMF

```
Input: $Y$
Output: $A$, $X$
initialization;
while $i \leq max_{iter}$ do
    // latent variables of dIBP
    Update $\mu^{(0)}$ by Eq. (43);
    Update $\mu^{(1)}$ by Eq. (41);
    Update $\mu^{(2)}$ by Eq. (42);
    Update $Z$ by Eq. (23) and (24);
    Update $\alpha_m$ and $\alpha_n$ by Eq. (44);
    // latent variables of NMF
    Update $V$ by Eq. (25) and (26);
    $i++$;
return $A$ and $X$;
```

variable of NMF for other implementations. The cost of the posterior inference for the C-dIBP update is $O(2K^\dagger + (M+N)(K^\dagger)^2)$, which is smaller than BB-dIBP with $K^\dagger$. Under relatively large $K^\dagger$, they are almost equal. The update cost for GP-dIBP is $O(MNK^\dagger + (M+N)(K^\dagger)^2)$ which is higher than BB-dIBP and C-dIBP. Finally, the inference of BB-dIBP can be calculated in $O((M+N+3)K^\dagger + (M+N)(K^\dagger)^2)$ time, which is lower than GP-dIBP, but a little higher than BB-dIBP and C-dIBP. Note that the scalability of the traditional Gibbs sampling is naturally not good because the samples are dependent upon each other's. Fortunately, with the derived conditional posterior distributions in hand, it is easy to design variational inference algorithms for them to handle the so-called big data [54], [55], which will be considered in our future work.

VI. Experiments

In this section, we first take a series of experiments on the synthetic data to show the merits of this paper by comparing the traditional NMF, single IBP-based NMF, and GP-based NMF on sparsity and nonparametricity (Section VI-A) and flexibility (Section VI-B). A real-world task, i.e., document-word co-clustering, is conducted to show the usefulness of this paper compared with other models (Section VI-C).

A. Evaluation of Convergence

This section checks and compares the convergence of the proposed models. First, we randomly generate a data matrix with size $M = 10 \; and N = 20$. The Geweke test [56] is then used for the sampling convergence check. This test splits the samples into two parts (after removing a burn-in period, i.e., the first 1000 iterations): the first 1000 and the last 2000. If the chain is at stationarity, the means of the two parts should be equal. A smaller difference between them therefore indicates better convergence. Here, we use two latent variables: one is the latent factor number and the other is the data likelihood. Fig. 3 shows the statistical results of the models, i.e., GP-dIBP-NMF, BB-dIBP-NMF, C-dIBP-NMF, and HBP-dIBP-NMF, after 10 independent runs on the same synthetic data set. From this figure, we can see that the proposed models (BB-dIBP and C-dIBP) have better convergence than GP-dIBP. Of the proposed models, the Copula-based model achieves the best performance on convergence.

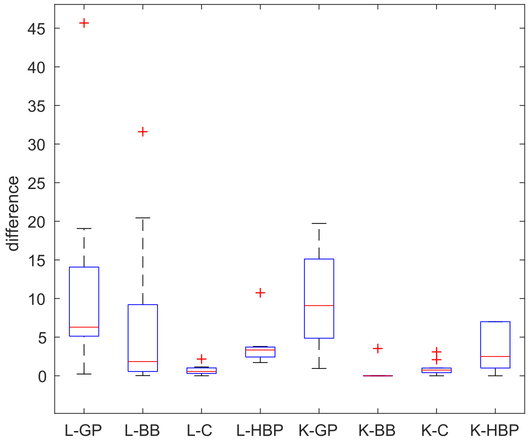


Fig. 3. Comparison of convergence of different models. For instance, $L-GP$ denotes the difference between the data likelihoods from GP-dIBP-NMF; $K-GP$ denotes the difference between effective factor number from GP-dIBP-NMF.

B. Evaluation of Sparsity and Nonparametricity

We randomly generate a matrix $Y_{20\times30}$ using the following procedure: 1) give a vector [0.5, 0.4, 0.3, 0.2, 0.1] as the parameters of five Bernoulli distributions; 2) randomly generate the elements of ith column (i.e., $A(:,i)$ and $X(:,i)$) of both matrices $A_{20\times5}$ and $X_{30\times5}$ using the Bernoulli distribution with ith value of the above vector; and 3) generate Y as the product of A and X as $Y = AX^T$. Since the values of the parameters of Bernoulli distributions are small, the generated factor matrices A and X tend to be sparse. Here, the matrix Y is used as the input data for different algorithms (i.e., traditional sparse NMF (sNMF) in (1), sIBP-NMF, and the proposed dIBP-NMF), and the sparsity of the learned factor matrices from the different algorithms are evaluated and compared. We design the following metric to quantitatively compare the sparsity from different algorithms: $S_A = \sum_m \sum_k \mathbb{1}(A(m,k) = 0)$ and $S_X = \sum_n \sum_k \mathbb{1}(X(n,k) = 0)$, where $\mathbb{1}(\cdot)$ is an indicator function parameterized by a condition which equals 1 if condition is satisfied; 0, otherwise. Here, we separately evaluate the sparsity of A and X considering the sIBP-NMF. Note that the 0.00000001 acts as a relaxation of the sparsity in the implementation. In the experiments, we randomly generate a number 100 of $Y_{20\times30}$ using the above procedure. The results are shown in Fig. 4, which compares the sparsity of the learned factor matrices A and X from different algorithms. Note that the results are an average of 100 trials. The x-axis denotes the number of factors for NMF and sNMF since they need this as input, but sIBP-NMF and dIBP-NMF do not. We can see that sNMF has greater sparsity than NMF due to the sparse constraint. It appears that the sparsity increases as the number of factors also increases. Since sIBP-NMF and dIBP-NMF do not need the factor number as input, the values of S_A and S_X from two algorithms are equal. For each trial, we can obtain a distribution on factor number, as shown by two examples in Fig. 5 for

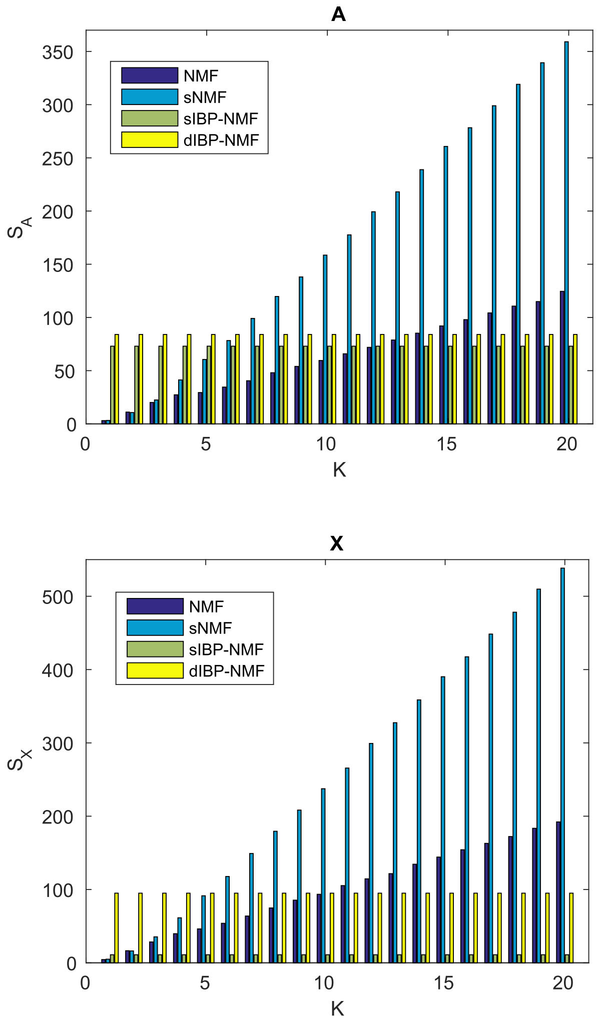


Fig. 4. Comparison of the sparsity on synthetic data set between traditional NMF (NMF), traditional sparse NMF (sNMF), single IBP-based sparse NMF (sIBP-NMF), and doubly IBP-based sparse NMF (dIBP-NMF). Note that dIBP-NMF here is based on bivariate Beta distribution.

sIBP-NMF and dIBP-NMF with 5000 Gibbs samples. The peak of distribution is seen as the final learned factor number from the algorithm. The statistics on the learned factor numbers from 100 trials are shown in the right subfigure in Fig. 5. The averages are around the benchmark (i.e., 5 in the generative procedure), which denotes the relative accuracy of the factor learning (i.e., nonparametric property). Not surprisingly, sIBP-NMF has sparse A but not sparse X (the value of S_A is large even compared with the value in sNMF but S_X is small). Its sparsity of A is due to its IBP prior. Since there are two IBP priors for A and X in dIBP-NMF, the resultant A and X are both sparse. Therefore, we can draw the conclusion that the proposed dIBP-NMF could obtain two sparse factor matrices but sIBP-NMF could only obtain one, which will impact on the ability to conduct the co-clustering task that will be demonstrated in Section VI-D.

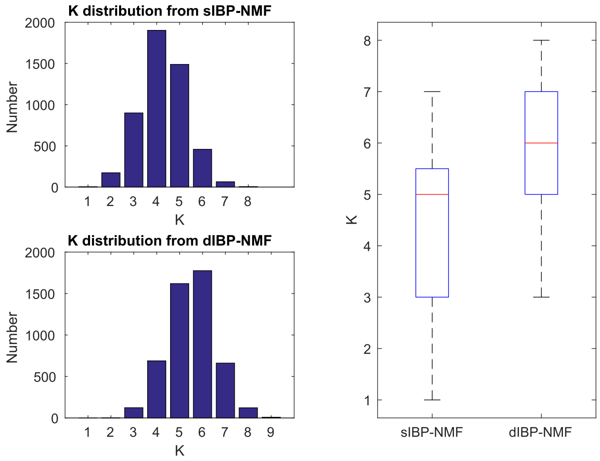


Fig. 5. Comparison of the learned topic number distribution on synthetic data set between single IBP-based sparse NMF (sIBP-NMF) and doubly IBP-based sparse NMF (dIBP-NMF). Note that dIBP-NMF here is based on bivariate Beta distribution.

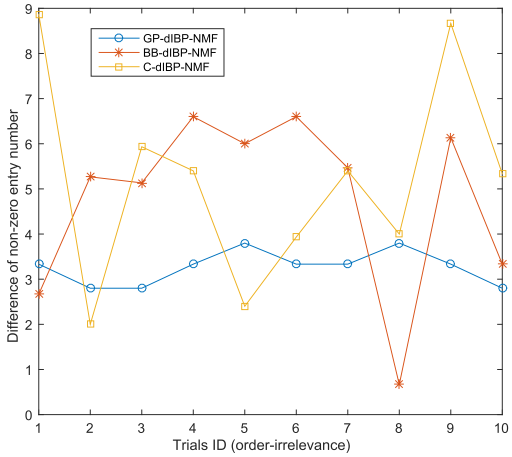


Fig. 6. Results on synthetic data to show the flexibility of the different models. The x-axis denotes the trial IDs (order is irrelevant).

C. Evaluation of Correlation Flexibility

As claimed, our proposed models have greater correlation flexibility for allowing the numbers of nonzero entries in factor matrices more different from each other compared with the GP-based dIBP. To show this flexibility, we first design a metric to measure the flexibility by comparing the number of nonzero entries of two factor matrices (A and X) from the models. The mean of the differences between the corresponding columns of A and X is $(1/K)\sum_{k=1}^{K}|N_k^{(1)}-N_k^{(2)}|$, where K is the number of columns of both matrices, $N_k^{(1)}$ is the number of nonzero entries of kth column of A, and $N_k^{(2)}$ is the number of nonzero entries of kth column of X. It appears that the larger this metric is, the more flexibility a model has. We have ten randomly generated matrices of the same size: 20 × 30, and we run three models on ten matrices. The designed metric has been evaluated on the learned factor matrices of different models. As shown in Fig. 6, we can see that the metrics on BB-dIBP-NMF and C-dIBP-NMF are larger than those on GP-dIBP-NMF in most trials and are also with more fluctuations and larger nonzero entry number differences compared with the GP-dIBP-NMF. We conclude that the proposed dIBPs are more flexible than the GP-based dIBP.

TABLE II
NUMERICAL RESULTS OF DOCUMENT-WORD CO-CLUSTERING ON *Citeseer* DATASET

Evaluation Metric		Models or Algorithms							
		Bayesian Nonparametric NMF Models					Parametric Models or Algorithms		
		sIBP	BB-dIBP	C-dIBP	GP-dIBP	HBP-dIBP	NMF	SNMF	Spectral Clustering
doc	JC	0.2390	0.2483	0.2411	0.2323	0.2406	0.2050±0.0321	0.2318±0.0312	0.1784
	FM	0.3860	0.4132	0.3876	0.3902	0.3906	0.3420±0.0440	0.3764±0.0404	0.4219
	F1	0.3510	0.3417	0.3729	0.3851	0.3697	0.3398±0.0447	0.3753±0.0415	0.3029
word	JC	0.6521	0.6923	0.6793	0.6762	0.6627	0.6412±0.0425	0.6940±0.0467	N/A
	FM	0.7798	0.8417	0.8301	0.8361	0.8154	0.7809±0.0242	0.8325±0.0263	N/A
	F1	0.7973	0.8223	0.8119	0.8182	0.8063	0.7667±0.0273	0.8186±0.0294	N/A

TABLE III
NUMERICAL RESULTS OF DOCUMENT-WORD CO-CLUSTERING ON *Cora* DATASET

Evaluation Metric		Models or Algorithms							
		Bayesian Nonparametric Models					Parametric Models or Algorithms		
		sIBP	BB-dIBP	C-dIBP	GP-dIBP	HBP-dIBP	NMF	SNMF	Spectral Clustering
doc	JC	0.1680	0.1720	0.1700	0.1650	0.1683	0.1524±0.0239	0.1678±0.0170	0.1691
	FM	0.2770	0.2790	0.2610	0.2630	0.2700	0.2729±0.0366	0.2882±0.0263	0.3123
	F1	0.2890	0.2930	0.3090	0.2610	0.2633	0.2688±0.0354	0.2870±0.0254	0.2737
word	JC	0.4380	0.5320	0.5190	0.5170	0.5210	0.4328±0.0545	0.5098±0.0464	N/A
	FM	0.6780	0.7320	0.7190	0.7300	0.6809	0.5946±0.0314	0.7129±0.0292	N/A
	F1	0.6420	0.6770	0.6650	0.6650	0.6532	0.5739±0.0361	0.6742±0.0366	N/A

D. Real-World Task: Document-Word Co-Clustering

In this section, we apply the proposed algorithms to a real-world task: document-word co-clustering. The real-world data sets[2] used for this task are:

1) *Cora Data Set:* The Cora data set consists of 2708 scientific publications classified into seven classes. The dictionary consists of 1433 unique words.
2) *Citeseer Data Set:* The CiteSeer data set consists of 3312 scientific publications. The dictionary consists of 3703 unique words. The labels of these papers are set as their research areas.

The above-mentioned data sets already have benchmarks for the document clusters, but do not have benchmarks for the word clusters. We use the co-occurrence relations between words to generate a distance matrix through which we can obtain a number (10 being the best) of word clusters by spectral clustering algorithm. These word clusters are seen as benchmarks. The evaluation metrics (bigger means better) for clustering are Jaccard Coefficient: $JC = (a/a + b + c)$, Folkes&Mallows: $FM = ((a/a + b)(a/a + c))^{1/2}$ and F1 measure: $F1 = (2a^2/2a^2 + ac + ab)$, where a is the number of two points that are in the same cluster of both benchmark result and clustering result, b is the number of two points that are in the same cluster of benchmark result but in different clusters of clustering result, and c is the number of two points that are not in the same cluster of the two benchmark result but are in the same cluster of clustering result.

We compare the following models on the document-word co-clustering task: three parametric models or algorithms [classical NMF, sparse NMF (SNMF) in (1), and spectral clustering] and five Bayesian nonparametric models [single IBP-based sparse NMF (sIBP-NMF) [13], [14], GP-dIBP-NMF, BB-dIBP-NMF, C-dIBP-NMF, and HBP-dIBP-NMF]. When applied to document-word co-clustering, the output A and X from each of the above-mentioned models can be considered as new representations of documents and words on latent factors. Based on these new representations, we can use a clustering algorithm (*K-means*, in this section, and K is set as the number of benchmark clusters) to conduct document and word clustering. Since a common algorithm (i.e., *K-means*) is adopted for all models, the performance of models will be only determined by the learned new data representation A and X. We also compare the performance of above-mentioned models with spectral clustering on document clustering. Note that since the benchmark of word clustering is built based on spectral clustering, we do not compare spectral clustering with other models or algorithms on word clustering.

The results on *Citeseer* are listed in Table II. Since NMF and SNMF require the factor number as input, we adjust this parameter for them from 1 to 100; thus, there are fluctuations in the results from NMF and SNMF in Table II where the standard deviations are given after the mean value. The other models are all nonparametric models which do not need the factor number as input, so the results of these models are listed in the table with only one value. The learned factor numbers are 9 (from sIBP-NMF), 10 (from BB-dIBP-NMF), 12 (from C-dIBP-NMF), 12 (from GP-dIBP-NMF), and 10 (from HBP-dIBP-NMF). The results on *Cora* are listed in Table III. The learned factor numbers are 10 (from sIBP-NMF), 17 (from BB-dIBP-NMF), 15 (from C-dIBP-NMF), 22 (from GP-dIBP-NMF), and 12 (from HBP-dIBP-NMF). From these results,

[2] http://linqs.cs.umd.edu/projects/projects/lbc/

TABLE IV

NUMERICAL RESULTS OF HELD-OUT DATA PREDICTION ON *Citeseer* DATA SET

Prediction precision		Models or Algorithms						
		Bayesian Nonparametric NMF Models					Parametric Models or Algorithms	
		sIBP	BB-dIBP	C-dIBP	GP-dIBP	HBP-dIBP	NMF	SNMF
doc	training	0.6981	0.7165	0.7293	0.7097	0.7121	0.6613±0.0659	0.6792±0.0953
	test	0.6977	0.7063	0.7087	0.6944	0.7002	0.6386±0.0753	0.6973±0.0891
word	training	0.8638	0.8810	0.8677	0.8598	0.8701	0.8655±0.0159	0.8712±0.0155
	test	0.8203	0.8332	0.8307	0.8267	0.8312	0.8193±0.0269	0.8170±0.0288

TABLE V

NUMERICAL RESULTS OF HELD-OUT DATA PREDICTION ON *Cora* DATA SET

Prediction precision		Models or Algorithms						
		Bayesian Nonparametric NMF Models					Parametric Models or Algorithms	
		sIBP	BB-dIBP	C-dIBP	GP-dIBP	HBP-dIBP	NMF	SNMF
doc	training	0.6790	0.7212	0.7197	0.7112	0.7081	0.6362±0.0832	0.6487±0.0877
	test	0.6298	0.6765	0.6904	0.6600	0.6556	0.6117±0.0762	0.6231±0.0965
word	training	0.8952	0.9091	0.9126	0.8889	0.9001	0.8960±0.0157	0.8940±0.0262
	test	0.8643	0.8673	0.8799	0.8620	0.8692	0.8759±0.0280	0.8431±0.0301

we can see that: 1) there are fluctuations in the performance of an SNMF and NMF on document and word clustering due to the factor number parameter. Note that spectral clustering needs the cluster number to be fixed in advance, like traditional clustering methods. For the sake of comparison, we simply feed the benchmark cluster number of the data sets to algorithms, so there is no fluctuation in its results in the table; 2) although sIBP-NMF has relatively good performance on document clustering, the performance on word clustering is poor, which may be due to its single side sparse and nonparametric control; and 3) five Bayesian nonparametric models achieve comparable or better (in most cases) performance than three parametric models or algorithms. Considering the release of the factor number assumption, these Bayesian nonparametric models are an improvement on traditional NMF. 4) comparing sIBP-NMF, the other Bayesian nonparametric models all perform better on both document and word clustering with a weak exception by HBP-dIBP on document clustering on *Cora*. To summarize, without 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 overall.

We also check how the learned models perform on the prediction of held out data by five-fold cross validation. The setting is as follows: 1) train the models on the training data ($Y^{\text{training}}_{\text{doc}\times\text{words}}$) to obtain new representations for the training data, i.e., new representations of documents $A^{\text{training}}_{\text{doc}\times\text{factor}}$; 2) use the new representation of data to train a classifier (i.e., KNN here); and 3) predict the labels of the test data using the new representations of test data $A^{\text{test}}_{\text{doc}\times\text{factor}} = Y^{\text{test}}(X^{\text{test}}_{\text{factor}\times\text{words}})^{-1}$ and the trained classifier. The procedure for words is similar to the above. The prediction metric is the ratio of the correctly predicted data number to the number of all data. The results are listed in Tables IV and V. Note that the standard deviations in these tables are from the different factor numbers, not the cross validation. We can see from these results that: 1) The models are not overfitting because there is not a big difference between the prediction on the training and test data; 2) The Bayesian nonparametric models have relatively better performance compared with the NMF and SNMF ;and 3) BB-dIBP-NMF and C-dIBP-NMF achieve the best prediction on *Citeseer* and *Cora*, respectively.

VII. CONCLUSION AND FURTHER STUDY

Nonnegative matrix factorization is advantageous for many machine learning tasks (e.g., co-clustering), 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 doubly sparse nonparametric NMF framework based on dIBP to remove the assumption. First, two models were built by implementing this framework using GP-based dIBP and HBP-based dIBP, which successfully remove the assumption but suffers from larger model complexity or less flexibility. Then, we proposed two new dIBPs through bivariate Beta distribution and a copula. The advantages of the models based on the new dIBPs is that: 1) they have simpler model structures than models with GP-based dIBP and 2) the correlation in data can be directly learned out, which can be seen as a measurement of the focus degree of hidden factors/topics. Finally, four inference algorithms have been designed for the proposed models, respectively. The experiments on synthetic and real-world data sets demonstrates the capability of the proposed models to perform NMF without predefining the dimensionality and more correlation flexibility compared with the GP-based dIBP and HBP-based dIBP.

One possible future area of study for this paper is the aspect of efficiency. Current Gibbs sampling inference is not efficient enough for big data. Our future study will focus on the efficiency of the inference of the proposed models using the variational inference strategy.

APPENDIX A
CORRELATION FROM BIVARIATE BETA DISTRIBUTION-BASED DIBP

Initially, we know the expectations and variances for r_1 and r_2 of bivariate Beta distribution in (6) with $c^0 = 1$

$$\begin{aligned}\mathbb{E}[r_1] &= \frac{a^0}{a^0+1}, \quad \mathbb{V}[r_1] = \frac{a^0}{(a^0+1)^2(a^0+2)}\\ \mathbb{E}[r_2] &= \frac{b^0}{b^0+1}, \quad \mathbb{V}[r_2] = \frac{b^0}{(b^0+1)^2(b^0+2)}.\end{aligned} \tag{30}$$

and

$$\begin{aligned}\mathbb{E}[r_1 r_2] = \frac{a^0 b^0 \Gamma(a^0+1)\Gamma(b^0+1)}{(a^0+b^0+1)\Gamma(a^0+b^0+2)}\\ {}_3F_2(a^0+1, b^0+1, a^0+b^0+1\\ a^0+b^0+2, a^0+b^0+2; 1).\end{aligned} \tag{31}$$

Then, we know

$$\mathrm{corr}(r_1, r_2) = \frac{cov(r_1, r_2)}{\sqrt{\mathbb{V}[r_1]\mathbb{V}[r_2]}} = \frac{\mathbb{E}[r_1 r_2] - \mathbb{E}[r_1]\mathbb{E}[r_2]}{\sqrt{\mathbb{V}[r_1]\mathbb{V}[r_2]}} \tag{32}$$

where $\mathrm{corr}(r_1, r_2)$ is the correlation between r_1 and r_2, and cov is the covariance. After substitution with the above equations, we obtain the result of (9).

APPENDIX B
CONDITIONAL DISTRIBUTIONS FOR THE GP-BASED DIBP-NMF

The conditional distributions are

A. Sampling μ

$$p(\mu_k|\cdots) \propto \frac{\mu_K^\alpha}{\mu_k} \prod_{t=1}^{2} \prod_{n}^{N_t} \left(\gamma_k^t\right)^{z_{n,k}^t} \left(1 - \gamma_k^t\right)^{1-z_{n,k}^t} \tag{33}$$

where

$$\gamma_k^t = F\left(F^{-1}\left(\mu_k|0, \Sigma_k^{(t,t)} + \eta^2\right) - g_k^t|0, \eta^2\right) \tag{34}$$

where $F()$ is a normal cdf.

B. Sampling g

$$p(g_k|\cdots) \propto \mathcal{N}(g_k|0, \Sigma_k) \cdot \prod_t \prod_n^{N_t} \mathcal{N}\left(h_{n,k}^t | g_k^t, \eta^2\right) \tag{35}$$

where $\mathcal{N}()$ denotes normal distribution.

C. Sampling h

$$p\left(h_{n,k}^t|\cdots\right) \propto \mathcal{N}\left(g_k^t, \eta^2\right), \quad \begin{cases} h_{n,k}^t \in \left(-\infty, \widetilde{\mu}_k^t\right] & \text{if } z_{n,k}^t = 1\\ h_{n,k}^t \in \left[\widetilde{\mu}_k^t, +\infty\right) & \text{if } z_{n,k}^t = 0 \end{cases} \tag{36}$$

where

$$\widetilde{\mu}_k^t = F^{-1}\left(\mu_k|0, \Sigma_k^{(t,t)} + \eta^2\right). \tag{37}$$

D. Sampling s

$$p(s|\cdots) \propto \mathrm{Gam}(s; hs, 1) \prod_k^K \mathcal{N}(g_k|0, \Sigma_k). \tag{38}$$

E. Sampling Z

$$\begin{aligned} p\left(z_{m,k}^{(1)} = 1\right) &\propto \gamma_k^1 \prod_n e^{-y_{m,n}\left(\sum_l v_{m,l}^{(1)} z_{m,l}^{(1)} v_{n,l}^{(2)} z_{n,l}^{(2)} + \epsilon\right)}\\ p\left(z_{m,k}^{(1)} = 0\right) &\propto (1 - \gamma_k^1) \prod_n e^{-y_{m,n}\left(\sum_l v_{m,l}^{(1)} z_{m,l}^{(1)} v_{n,l}^{(2)} z_{n,l}^{(2)} + \epsilon\right)} \end{aligned} \tag{39}$$

and

$$\begin{aligned} p\left(z_{n,k}^{(2)} = 1\right) &\propto \gamma_k^2 \prod_m e^{-y_{m,n}\left(\sum_l v_{m,l}^{(1)} z_{m,l}^{(1)} v_{n,l}^{(2)} z_{n,l}^{(2)} + \epsilon\right)}\\ p\left(z_{n,k}^{(2)} = 0\right) &\propto (1 - \gamma_k^2) \prod_m e^{-y_{m,n}\left(\sum_l v_{m,l}^{(1)} z_{m,l}^{(1)} v_{n,l}^{(2)} z_{n,l}^{(2)} + \epsilon\right)} \end{aligned} \tag{40}$$

where γ_k^t is the same as in (34).

APPENDIX C
CONDITIONAL DISTRIBUTIONS FOR THE HBP-BASED DIBP-NMF

When using truncation level $K^\dagger$, the approximation of the original HBP-based dIBP [57] is as $\mu_k^{(0)} \sim Beta((\alpha_0/K^\dagger), \alpha_0(1 - (1/K^\dagger)))$ where $\mu_k^{(1)} \sim Beta(\alpha_m \mu_k^{(0)}, \alpha_m(1 - \mu_k^{(0)}))$ and $\mu_k^{(2)} sim Beta(\alpha_n \mu_k^{(0)}, \alpha_n(1 - \mu_k^{(0)}))$.

The conditional distributions of the hidden variables are

A. Sampling μ_k^1

$$\mu_k^{(1)} \sim \mathrm{Beta}\left(\alpha_m \mu_k^{(0)} + \sum_m z_{m,k}^{(1)}, \alpha_m\left(1 - \mu_k^{(0)}\right) + M - \sum_m z_{m,k}^{(1)}\right). \tag{41}$$

B. Sampling μ_k^2

$$\mu_k^{(2)}|\cdots \sim \mathrm{Beta}\left(\alpha_n \mu_k^{(0)} + \sum_n z_{n,k}^{(2)}, \alpha_n\left(1 - \mu_k^{(0)}\right) + N - \sum_n z_{n,k}^{(2)}\right). \tag{42}$$

C. Sampling μ_k^0

The log probabilistic density of μ_k^0 is proportional to

$$\begin{aligned} &\alpha_0\left(\frac{1}{K^\dagger} - 1\right)\log \mu_k^{(0)} + \frac{\alpha_0}{K^\dagger}\log\left(\mu_k^{(0)}\right)\\ &\quad + \alpha_m \mu_k^{(0)} \log \mu_k^{(1)} + \alpha_m\left(1 - \mu_k^{(0)}\right)\log\left(1 - \mu_k^{(1)}\right)\\ &\quad + \left(\alpha_n \mu_k^{(0)}\right)\log \mu_k^{(1)} + \alpha_n\left(1 - \mu_k^{(0)}\right)\log\left(1 - \mu_k^{(2)}\right)\\ &\quad - \log\Gamma\left(\alpha_m \mu_k^{(0)}\right) - \log\Gamma\left(\alpha_m\left(1 - \mu_k^{(0)}\right)\right)\\ &\quad - \log\Gamma\left(\alpha_n \mu_k^{(0)}\right) - \log\Gamma\left(\alpha_n\left(1 - \mu_k^{(0)}\right)\right). \end{aligned} \tag{43}$$

D. Sampling α_m

$$p(\alpha_m) \propto \mathrm{Gam}(\alpha_m; hp) \prod_{k=1}^{K} \mathrm{Beta}\big(\alpha_m \mu_k^{(0)}, \alpha_m \big(1 - \mu_k^{(0)}\big)\big). \tag{44}$$

References

[1] M. Ye, Y. Qian, and J. Zhou, "Multitask sparse nonnegative matrix factorization for joint spectral–spatial hyperspectral imagery denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2621–2639, May 2015.

[2] R. Sandler and M. Lindenbaum, "Nonnegative matrix factorization with earth mover's distance metric for image analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1590–1602, Aug. 2011.

[3] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1273–1284, May 2014.

[4] M. Heiler and C. Schnörr, "Learning sparse representations by non-negative matrix factorization and sequential cone programming," *J. Mach. Learn. Res.*, vol. 7, pp. 1385–1407, Jul. 2006.

[5] Z. Yang, G. Zhou, S. Xie, S. Ding, J.-M. Yang, and J. Zhang, "Blind spectral unmixing based on sparse nonnegative matrix factorization," *IEEE Trans. Image Process.*, vol. 20, no. 4, pp. 1112–1125, Apr. 2011.

[6] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proc. 7th ACM Int. Conf. Knowl. Discovery Data Mining (KDD)*, San Francisco, CA, USA, Aug. 2001, pp. 269–274.

[7] C.-E. Bichot, "Co-clustering documents and words by minimizing the normalized cut objective function," *J. Math. Model. Algorithms*, vol. 9, no. 2, pp. 131–147, 2010.

[8] C. J. Crouch, "A cluster-based approach to thesaurus construction," in *Proc. 11th Annu. Int. ACM Conf. Res. Develop. Inf. Retr. (SIGIR)*, Grenoble, France, Jun. 1988, pp. 309–320.

[9] H. Shan and A. Banerjee, "Bayesian co-clustering," in *Proc. 8th IEEE Int. Conf. Data Mining (ICDM)*, Pisa, Italy, Dec. 2008, pp. 530–539.

[10] S. J. Gershman and D. M. Blei, "A tutorial on Bayesian nonparametric models," *J. Math. Psychol.*, vol. 56, no. 1, pp. 1–12, 2012.

[11] R. M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *J. Comput. Graph. Statist.*, vol. 9, no. 2, pp. 249–265, Jun. 2000.

[12] J. Xuan, J. Lu, G. Zhang, R. Y. Da Xu, and X. Luo, "Infinite author topic model based on mixed gamma-negative binomial process," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Atlantic City, NJ, USA, Nov. 2015, pp. 489–498.

[13] T. L. Griffiths and Z. Ghahramani, "Infinite latent feature models and the Indian buffet process," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, Dec. 2005, pp. 475–482.

[14] N. Ding, Y. Qi, R. Xiang, I. Molloy, and N. Li, "Nonparametric Bayesian matrix factorization by power-EP," in *Proc. 13th Int. Conf. Artif. Intell. Statist. (AISTATS)*, Sardinia, Italy, May 2010, pp. 169–176.

[15] S. Williamson, P. Orbanz, and Z. Ghahramani, "Dependent Indian buffet processes," in *Proc. 13th Int. Conf. Artif. Intell. Statist. (AISTATS)*, Sardinia, Italy, May 2010, pp. 924–931.

[16] R. Thibaux and M. I. Jordan, "Hierarchical beta processes and the Indian buffet process," in *Proc. 11th Int. Conf. Artif. Intell. Statist. (AISTATS)*, San Juan, Puerto Rico, Mar. 2007, pp. 564–571.

[17] C. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 45–55, Jan. 2010.

[18] T. L. Griffiths and Z. Ghahramani, "The Indian buffet process: An introduction and review," *J. Mach. Learn. Res.*, vol. 12, pp. 1185–1224, Apr. 2011.

[19] Y. W. Teh, D. Görür, and Z. Ghahramani, "Stick-breaking construction for the Indian buffet process," in *Proc. 11th Int. Conf. Artif. Intell. Statist. (AISTATS)*, San Juan, Puerto Rico, Mar. 2007, pp. 556–563.

[20] S.-I. Huh, M. D. Gupta, and J. Xiao, "Supervised nonnegative matrix factorization," U.S. Patent 8 498 949, Jul. 30, 2013.

[21] C. J. Lin, "On the convergence of multiplicative update algorithms for nonnegative matrix factorization," *IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1589–1596, Nov. 2007.

[22] Y. Yuan, X. Li, Y. Pang, X. Lu, and D. Tao, "Binary sparse nonnegative matrix factorization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 5, pp. 772–777, May 2009.

[23] E. M. Airoldi, D. M. Blei, E. A. Erosheva, and S. E. Fienberg, Eds., *Handbook of Mixed Membership Models and Their Applications*. London, U.K.: Chapman & Hall, 2014.

[24] D. Liang, M. D. Hoffman, and D. P. W. Ellis, "Beta process sparse nonnegative matrix factorization for music," in *Proc. 14th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, Curitiba, Brazil, Nov. 2013, pp. 375–380.

[25] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[26] D. M. Blei and J. D. Lafferty, "A correlated topic model of science," *Ann. Appl. Statist.*, vol. 1, no. 1, pp. 17–35, 2007.

[27] P. Vellanki, T. Duong, S. Venkatesh, and D. Phung, "Nonparametric discovery of learning patterns and autism subgroups from therapeutic data," in *Proc. 22nd Int. Conf. Pattern Recognit. (ICPR)*, Stockholm, Sweden, Aug. 2014, pp. 1828–1833.

[28] T. Broderick, M. I. Jordan, and J. Pitman, "Beta processes, stick-breaking and power laws," *Bayesian Anal.*, vol. 7, no. 2, pp. 439–476, 2012.

[29] M. D. Hoffman, D. M. Blei, and P. R. Cook, "Bayesian nonparametric matrix factorization for recorded music," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, Haifa, Israel, Jun. 2010, pp. 439–446.

[30] A. Roychowdhury and B. Kulis, "Gamma processes, stick-breaking, and variational inference," in *Proc. 18th Int. Conf. Artif. Intell. Statist. (AISTATS)*, Stockholm, Sweden, May 2015, pp. 800–808.

[31] M. Zhou, L. A. Hannah, D. B. Dunson, and L. Carin, "Beta-negative binomial process and Poisson factor analysis," in *Proc. 15th Int. Conf. Artif. Intell. Statist. (AISTATS)*, La Palma, Spain, Dec. 2012, pp. 1462–1471.

[32] M. Zhou, "Nonparametric Bayesian matrix factorization for assortative networks," in *Proc. IEEE 23rd Eur. Signal Process. Conf. (EUSIPCO)*, Nice, France, Aug./Sep. 2015, pp. 2776–2780.

[33] A. Acharya, D. Teffer, J. Henderson, M. Tyler, M. Zhou, and J. Ghosh, "Gamma process Poisson factorization for joint modeling of network and documents," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases (ECML)*, Porto, Portugal, Sep. 2015, pp. 283–299.

[34] S. Mirzaei, H. Van Hamme, and Y. Norouzi, "Bayesian non-parametric matrix factorization for discovering words in spoken utterances," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, Oct. 2013, pp. 1–4.

[35] D. Liang and M. D. Hoffman. (2014). "Beta process non-negative matrix factorization with stochastic structured mean-field variational inference." [Online]. Available: https://arxiv.org/abs/1411.1804

[36] F. Wood and T. L. Griffiths, "Particle filtering for nonparametric Bayesian matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, Dec. 2006, pp. 1513–1520.

[37] F. Doshi-Velez and S. A. Williamson, "Restricted Indian buffet processes," *Statist. Comput.*, pp. 1–19, Jul. 2016. [Online]. Available: https://link.springer.com/article/10.1007/s11222-016-9681-y

[38] B. Ozdemir and L. S. Davis, "A probabilistic framework for multimodal retrieval using integrative Indian buffet process," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Denver, CO, USA, Dec. 2014, pp. 2384–2392.

[39] C. Heaukulani and D. M. Roy. (2015). "Gibbs-type Indian buffet processes." [Online]. Available: https://arxiv.org/abs/1512.02543

[40] Y. W. Teh and D. Gorur, "Indian buffet processes with power-law behavior," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, Dec. 2009, pp. 1838–1846.

[41] S. N. MacEachern, "Dependent Dirichlet processes," Dept. Statist., Ohio State Univ., Columbus, OH, USA, Tech. Rep., 2000.

[42] N. J. Foti and S. A. Williamson, "A survey of non-exchangeable priors for Bayesian nonparametric models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 2, pp. 359–371, Feb. 2015.

[43] K. T. Miller, T. Griffiths, and M. I. Jordan. (2012). "The phylogenetic Indian buffet process: A non-exchangeable nonparametric prior for latent features." [Online]. Available: https://arxiv.org/abs/1206.3279

[44] S. P. Chatzis, "A coupled Indian buffet process model for collaborative filtering," in *Proc. 4th Asian Conf. Mach. Learn. (ACML)*, Singapore, Nov. 2012, pp. 65–79.

[45] I. Olkin and R. Liu, "A bivariate beta distribution," *Statist. Probab. Lett.*, vol. 62, no. 4, pp. 407–412, 2003.

[46] P. K. Trivedi and D. M. Zimmer, *Copula Modeling: An Introduction for Practitioners*. Norwell, MA, USA: Now Publishers Inc, 2007.

[47] B. K. Beare, "Copulas and temporal dependence," *Econometrica*, vol. 78, no. 1, pp. 395–410, 2010.

[48] N. Foti, "Bayesian nonparametric methods for non-exchangeable data," Ph.D. dissertation, Dartmouth College, Hanover, NH, USA, 2013.

[49] J. E. Griffin and F. Leisen, “Compound random measures and their use in Bayesian non-parametrics,” *J. Roy. Statist. Soc. B (Statist. Methodol.)*, vol. 79, no. 2, pp. 525–545, 2017. [Online]. Available: http://dx.doi.org/10.1111/rssb.12176

[50] F. Leisen, A. Lijoi, and D. Spanó, “A vector of Dirichlet processes,” *Electron. J. Statist.*, vol. 7, pp. 62–90, 2013. [Online]. Available: http://dx.doi.org/10.1214/12-EJS764

[51] W. Zhu and F. Leisen, “A multivariate extension of a vector of two-parameter Poisson–Dirichlet processes,” *J. Nonparametric Statist.*, vol. 27, no. 1, pp. 89–105, 2015.

[52] S. Williamson, C. Wang, K. A. Heller, and D. M. Blei, “The IBP compound Dirichlet process and its application to focused topic modeling,” in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, Haifa, Israel, Jun. 2010, pp. 1151–1158.

[53] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, “Hierarchical Dirichlet processes,” *J. Amer. Statist. Assoc.*, vol. 101, no. 476, pp. 1566–1581, Dec. 2006.

[54] T. Salimans and D. A. Knowles, “Fixed-form variational posterior approximation through stochastic linear regression,” *Bayesian Anal.*, vol. 8, no. 4, pp. 837–882, Dec. 2013. [Online]. Available: http://dx.doi.org/10.1214/13-BA858

[55] A. Ansari, Y. Li, and J. Z. Zhang, “Variational Bayesian inference for big data marketing models1,” Ph.D. dissertation, Univ. Washington, Seattle, WA, USA, 2014.

[56] J. Geweke *et al.*, *Evaluating the Accuracy of Sampling-based Approaches to the Calculation of Posterior Moments*, vol. 196. Minneapolis, MN, USA: Federal Reserve Bank of Minneapolis, Research Department, 1991.

[57] T. Broderick, L. Mackey, J. Paisley, and M. I. Jordan, “Combinatorial clustering and the beta negative binomial process,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 2, pp. 290–306, Feb. 2015.

Junyu Xuan is currently a Post-Doctoral Research Fellow with the Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia. He has authored about 20 papers, including ACM Transactions on Information Systems, IEEE Transactions on Systems, Man and Cybernetics: Systems (TSMC), IEEE Transactions on Cybernetics, IEEE International Conference on Data Mining, and International Joint Conference on Neural Networks. His current research interests include machine learning, text mining, Web mining, and complex network.

Jie Lu (SM’13) is currently a Distinguished Professor and an Associate Dean Research with the Faculty of Engineering and Information Technology, University of Technology Sydney Ultimo, NSW, Australia. She has authored ten research books and 400 papers. Her current research interests include learning-based decision support systems.

Dr. Lu was a recipient of eight Australian Research Council discovery grants and 20 other grants. She serves as the Editor-in-Chief of Knowledge-based Systems and International Journal of Computational Intelligence Systems, and delivered 14 keynotes in international conferences.

Guangquan Zhang is currently an Associate Professor with the Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia. He has authored four monographs and over 300 papers in refereed journals, conference proceedings and book chapters. His current research interests include uncertain information processing.

Dr. Zhang received seven Australian Research Council discovery grants and guest edited many special issues for international journals.

Richard Yi Da Xu is currently a Senior Lecturer with the Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia. He has authored about 50 papers, including IEEE Transactions on Image Processing, IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Neural Networks and Learning Systems, Pattern Recognition, ACM Transactions on Knowledge Discovery from Data, Association for the Advancement of Artificial Intelligence, and The International Conference on Image Processing. His current research interests include machine learning, computer vision, and statistical data mining.

Xiangfeng Luo (M’10) is currently a Professor with the School of Computers, Shanghai University, Shanghai, China. He has authored over 140 papers in refereed journals, conference proceedings and book chapters, including THMS, TSMC, TBD, and TLT. His current research interests include Web wisdom, cognitive informatics, and text understanding.

Dr. Luo received four grants from the National Science Foundation of China and five other grants.