Nonparametric Bayesian Matrix Factorization by Power-EP

Nan Ding[+] **Yuan (Alan) Qi**[⋆] **Rongjing Xiang**[+] **Ian Molloy**[+] **Ninghui Li**[+]
[+] Department of Computer Science, Purdue University, West Lafayette, IN 47907
[⋆]Departments of Computer Science and Statistics, Purdue University, West Lafayette, IN 47907

Abstract

Many real-world applications can be modeled by matrix factorization. By approximating an observed data matrix as the product of two latent matrices, matrix factorization can reveal hidden structures embedded in data. A common challenge to use matrix factorization is determining the dimensionality of the latent matrices from data. Indian Buffet Processes (IBPs) enable us to apply the nonparametric Bayesian machinery to address this challenge. However, it remains a difficult task to learn nonparametric Bayesian matrix factorization models. In this paper, we propose a novel variational Bayesian method based on new equivalence classes of infinite matrices for learning these models. Furthermore, inspired by the success of nonnegative matrix factorization on many learning problems, we impose nonnegativity constraints on the latent matrices and mix variational inference with expectation propagation. This mixed inference method is unified in a power expectation propagation framework. Experimental results on image decomposition demonstrate the superior computational efficiency and the higher prediction accuracy of our methods compared to alternative Monte Carlo and variational inference methods for IBP models. We also apply the new methods to collaborative filtering and role mining and show the improved predictive performance over other matrix factorization methods.

Appearing in Proceedings of the 13^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9. Copyright 2010 by the authors.

1 Introduction

Matrix factorization models have been applied in many areas of machine learning, information retrieval, and computational biology. By approximating an observed data matrix by a product of two (or three) latent matrices, we can use matrix factorization to discover the hidden structure embedded in observed data. If the data is represented by a $(N \times D)$ matrix $\mathbf{X}$ where N is the number of D-dimensional observations, the goal of matrix factorization is to find latent matrices $\mathbf{Z}$ and $\mathbf{A}$ such that $\mathbf{X} \approx \mathbf{ZA}$. Each row of $\mathbf{A}$ can be viewed as a basis vector for $\mathbf{X}$ and the loading matrix $\mathbf{Z}$ determines how to combine these basis vectors together to reconstruct observations in $\mathbf{X}$. Typically, the number of rows of $\mathbf{A}$ is smaller than the number of observations, suggesting the latent matrices $\mathbf{Z}$ and $\mathbf{A}$ offer compact summary of the data.

Varying the dimensionality of latent matrices greatly affects the performance of matrix factorization methods. Instead of fixing the dimensionality K, Griffiths and Ghahramani (2005) propose nonparametric Bayesian matrix factorization models based on Indian buffet processes (IBPs). The IBP prior allows us to model latent matrices of infinite sizes and learn the dimensionality of effective (nonzero) latent matrices automatically. Given massive data, however, it remains a difficult task to efficiently estimate this nonparametric matrix factorization model.

In this paper, we present two novel approximate Bayesian inference methods to address this issue. These approximate inference methods are based on new equivalence classes for infinite matrices that contain only non-zero columns of those infinite matrices and are not in the left-ordered form proposed by Griffiths and Ghahramani (2005). Without the left-ordered constraint, these new equivalence classes allow us to approximate each column of the latent matrix Z as independent factors, so that approximate inference can be performed elegantly. Specifically, we present in Section 2 the variational approximation for these equivalence classes and adaptively select the

dimension of the latent matrices by maximizing the marginal likelihood of these models. We also use the variational inference to estimate observation noise and hyper-parameters of this model. Later we refer to this method as *infinite matrix factorization* for simplicity. Recently, Doshi-Velez et al. (2009) proposed a variational inference method for IBP. This method is based on a truncated stick-break representation and performs unfavorably compared to finite variational approximation. Unlike this method, our new method does not require any specification of a truncation level and empirically achieves much higher prediction accuracy (with more robustness) than the finite variational approximation.

Inspired by the success of nonnegative matrix factorization on many real-world applications such as image decomposition (Lee and Seung, 2001) and computational biology (Devarajan, 2008), we extend the nonparametric Bayesian matrix factorization models in Section 3 by imposing nonnegativity constraints on elements of the latent matrices. We call this new model *infinite nonnegative matrix factorization*. For the efficient inference on this new model, we combine the variational approximation inference with expectation propagation in the power expectation propagation framework (Minka, 2004).

In Section 4, we describe experimental results for image decomposition, demonstrating the superior computational efficiency and the improved prediction accuracy of IMF and INMF compared to alternative Monte Carlo and variational inference methods for IBP models. In addition, we apply IMF and INMF to collaborative filtering and role mining and demonstrate the improved performance of INF and INMF over other matrix factorization methods.

2 Infinite Matrix Factorization

Infinite matrix factorization models were proposed by Griffiths and Ghahramani (2005). They derive the nonparametric Bayesian prior distribution on an infinite matrix by taking the limit for a prior distribution on a finite Bayesian matrix and constructing equivalence classes on infinite matrices. To develop the new variational method for the infinite matrix factorization model, we also start from this finite model.

2.1 Finite Bayesian Matrix Factorization

Let us denote the $(N \times D)$ data matrix by $\mathbf{X}$. Our goal is to decompose the data matrix $\mathbf{X}$ into a product of two latent matrices $\mathbf{Z}$ $(N \times K)$ and $\mathbf{A}$ $(K \times D)$. The factorization can be modeled by a likelihood function $p(\mathbf{X}|\mathbf{Z}, \mathbf{A})$, which represents a probabilistic generative process for producing the data $\mathbf{X}$. We also assign priors over $\mathbf{Z}$ and $\mathbf{A}$ to capture our uncertainty in these latent matrices. Using a hierarchical Bayesian model, the problem of matrix factorization amounts to finding the posterior distribution of $\mathbf{Z}$ and $\mathbf{A}$:

$$p(\mathbf{Z}, \mathbf{A}|\mathbf{X}) \propto p(\mathbf{X}, \mathbf{Z}, \mathbf{A}) = p(\mathbf{X}|\mathbf{Z}, \mathbf{A})p(\mathbf{A})p(\mathbf{Z}|\boldsymbol{\pi})p(\boldsymbol{\pi})$$

where $p(\mathbf{A})$, $p(\mathbf{Z}|\pi)$ are the prior distributions, $\boldsymbol{\pi}$ is the parameter vector for the prior $p(\mathbf{Z}|\boldsymbol{\pi})$, and $p(\pi)$ is the hyper-prior in this hierarchical Bayesian model.

We assign a factorized Gaussian prior on $\mathbf{A} = \{a_{kj}\}$:

$$p(\mathbf{A}) = \prod_{k,j} p(a_{kj}) = \prod_{k,j} \frac{1}{\sqrt{2\pi\sigma_A^2}} \exp\big(-\frac{a_{kj}^2}{2\sigma_A^2}\big) \qquad (1)$$

We use a binary matrix $\mathbf{Z} = \{z_{ik}\}$ and denote its k^{th} column by $\mathbf{z}_{:,k}$ and its i^{th} row by $\mathbf{z}_{i,:}$. Using a factorized discrete distribution on $\mathbf{Z}$ with the mean parameter $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$, we have

$$p(\mathbf{Z}|\boldsymbol{\pi}) = \prod_{k,i} p(z_{ik}|\pi_k) = \prod_k \pi_k^{\sum_i z_{ik}} (1-\pi_k)^{N - \sum_i z_{ik}} \qquad (2)$$

We assign a conjugate prior over $\boldsymbol{\pi}$

$$p(\boldsymbol{\pi}) = \prod_k \text{Beta}(\frac{\alpha}{K}, 1) = \prod_k \frac{\alpha}{K} \pi_k^{\frac{\alpha}{K}-1} \qquad (3)$$

Note that α/K regularizes the sparsity of $\mathbf{Z}$; if K is large, π_k is concentrated around small values and therefore many elements of $\mathbf{Z}$ will be encouraged to be zero.

We can choose a data likelihood function based on applications at hand. Here we use the Gaussian likelihood function due to its popularity in practice.

$$p(\mathbf{X}|\mathbf{Z}, \mathbf{A}) = \prod_{i,j} \frac{1}{\sqrt{2\pi\sigma_X^2}} \exp\big(-\frac{1}{2\sigma_X^2}(x_{ij} - \mathbf{z}_{i,:}\mathbf{a}_{:,j})^2\big) \qquad (4)$$

where we denote the j^{th} column of $\mathbf{A}$ by $\mathbf{a}_{:,j}$. We assume the variance parameter σ_X is known for the time being.

2.2 Equivalence Classes

Instead of choosing K, the dimensionality of $\mathbf{Z}$, to a particular value, Griffiths and Ghahramani (2005) set $K \to \infty$ to obtain nonparametric Bayesian prior on $\mathbf{Z}$. In the infinite case, however, $p(\mathbf{z}_k = 0, \pi_k = 0)$ converges to 1. Thus the probability of any nonzero matrix $\mathbf{Z}$ is 0. Apparently, such a model is not practically applicable. To solve this issue, Doshi-Velez et al. (2009); Teh et al. (2007) uses a truncated stick-breaking approximation to an Indian buffet process. Their approach requires a predefined truncation level T and, according to Doshi-Velez et al. (2009); Teh et al. (2007), empirically this approach does not outperform simple finite

matrix factorization models described in the previous section.

Another approach to address this issue is using equivalence classes of infinite matrices. By grouping the infinite non-zero matrices into the equivalence classes, we can make sure that the probability of an equivalence class does not converge to 0 even though the probability of each infinite matrix in this class converges to 0. For example, Griffiths and Ghahramani (2005) defined equivalence classes on $\mathbf{Z}$ with respect to a function on $\mathbf{Z}$ that maps $\mathbf{Z}$ to left-ordered binary matrices (so that the columns are sorted). Each of these equivalence classes has a valid nonzero probability. However, since the columns of the left-ordered binary matrices are correlated in the left-ordered form, we cannot use adopt variational inference with factorized approximations based on these equivalence classes.

To remove this coupling effect, we define *new* equivalence classes on $\mathbf{Z}$. The key observation we have is if we remove all-zero columns from many infinite binary matrices, many of these matrices can be reduced to the same non-zero sub-matrix $\mathbf{Z}_+$. All the matrices $\mathbf{Z}$ that can be reduced to the same sub-matrix $\mathbf{Z}_+$ have the same effect in the data likelihood (4). To persevere the data likelihood, we define equivalence classes on binary matrices by a many-to-one function: each binary matrix $\mathbf{Z}$ is mapped to the representative of its class $\bar{\mathbf{Z}} = [\mathbf{Z}_+, \mathbf{Z}_0]$ where $\mathbf{Z}_0$ only contains all-zero columns. This mapping moves the non-zero columns to the left side of the all-zero columns without changing the order between non-zero columns. As a result, unlike the columns in the left-ordered equivalence classes, the non-zero columns of $\mathbf{Z}_+$ can be treated as independent factors, which enables us to use factorized variational approximations to the posterior distribution of $\mathbf{Z}_+$.

For a matrix with K columns and K_+ non-zero columns, we denote its equivalence class with a representative matrix $\bar{\mathbf{Z}}$ by $[\bar{\mathbf{Z}}]^K_{K_+}$. The number of the matrices that can be mapped to the same equivalence class $[\bar{\mathbf{Z}}]^K_{K_+}$ is simply $C_K^{K_+} = \frac{K!}{K_+!(K-K_+)!}$. Considering the joint distribution over $[\bar{\mathbf{Z}}]^K_{K_+}$ and $\boldsymbol{\pi}$, we have

$$\begin{aligned} p([\bar{\mathbf{Z}}]^K_{K_+}, \boldsymbol{\pi}) =& \mathbf{C}_K^{K_+} p(\bar{\mathbf{Z}}, \boldsymbol{\pi}) = \mathbf{C}_K^{K_+} p(\mathbf{Z}_+, \boldsymbol{\pi}_+) p(\mathbf{Z}_0, \boldsymbol{\pi}_0) \\ =& \mathbf{C}_K^{K_+} \prod_{k \le K_+} p(\mathbf{z}_{:,k}, \pi_k) \prod_{k > K_+} p(\mathbf{z}_{:,k} = 0, \pi_k) \end{aligned} \quad (5)$$

Note that in the above equation we partition $\boldsymbol{\pi}$ into $\boldsymbol{\pi}_+$ and $\boldsymbol{\pi}_0$ according to the partition of $\bar{\mathbf{Z}}$). It is easy to derive that as $K \to \infty$, although $p(\mathbf{Z})$ converges to 0 for any particular $\mathbf{Z} \in [\bar{\mathbf{Z}}]^\infty_{K_+}$, $p([\bar{\mathbf{Z}}]^\infty_{K_+})$ does not.

We also divide the rows of $\mathbf{A}$ into $\mathbf{A}_+$ and $\mathbf{A}_0$, such that $\mathbf{ZA} = \mathbf{Z}_+\mathbf{A}_+$. Now the task of learning IMFs is to find the posterior distribution of $[\bar{\mathbf{Z}}]^\infty_{K_+}$, $\mathbf{A}$ and $\boldsymbol{\pi}$. Since the exact posterior distributions of $[\bar{\mathbf{Z}}]^\infty_{K_+}$, $\boldsymbol{\pi}$ and $\mathbf{A}$ are computationally intractable, we describe a variational method to approximate them in the next section.

2.3 Variational inference for IMF

First, let us define the notation. We use $\tilde{\mathbf{z}}_{i,:}$ to denote the i^{th} row of $\mathbf{Z}_+$ and use $\tilde{\mathbf{a}}_{:,j}$ to denote the j^{th} column of $\mathbf{A}_+$.

We choose a factorized distribution to approximate the posterior distributions of $\bar{\mathbf{Z}}$, $\mathbf{A}$ and $\boldsymbol{\pi}$:

$$\begin{aligned} q(\bar{\mathbf{Z}}, \mathbf{A}, \boldsymbol{\pi}) =& \Big(\prod_{k=1}^{\infty} q(\pi_k) \prod_i q(z_{ik})\Big) \prod_{j=1}^{D} q(\mathbf{a}_{:,j}) \\ =& q(\boldsymbol{\pi}_+) q(\mathbf{Z}_+) q(\mathbf{A}_+) q(\mathbf{Z}_0, \boldsymbol{\pi}_0) q(\mathbf{A}_0) \\ \propto& \prod_{k \le K_+} \big(q(\pi_k) \prod_i q(z_{ik})\big) \Big(\prod_{j=1}^{D} q(\tilde{\mathbf{a}}_{:,j})\Big) q(\mathbf{Z}_0, \boldsymbol{\pi}_0) q(\mathbf{A}_0) \end{aligned}$$

Using Jensen's inequality, we immediately obtain

$$\begin{aligned} & \ln p(\mathbf{X}, K_+) \\ \ge& \sum_{\bar{\mathbf{Z}}} \int q(\bar{\mathbf{Z}}, \boldsymbol{\pi}, \mathbf{A}_+, \mathbf{A}_0) \ln \frac{p(\mathbf{X}, [\bar{\mathbf{Z}}]^\infty_{K_+}, \mathbf{A}, \boldsymbol{\pi})}{q(\bar{\mathbf{Z}}, \boldsymbol{\pi}, \mathbf{A})} \mathrm{d}\mathbf{A} \mathrm{d}\boldsymbol{\pi} \\ =& \lim_{K \to \infty} \sum_{\mathbf{Z}_+} \int q(\mathbf{Z}_+, \mathbf{A}_+, \boldsymbol{\pi}_+) \ln \frac{\mathbf{C}_K^{K_+} p(\mathbf{X}, \mathbf{Z}_+, \mathbf{A}_+, \boldsymbol{\pi}_+)}{q(\mathbf{Z}_+, \mathbf{A}_+, \boldsymbol{\pi}_+)} \mathrm{d}\mathbf{A}_+ \mathrm{d}\boldsymbol{\pi}_+ \\ \triangleq& L(q; K_+) \end{aligned} \quad (6)$$

The first equation holds since $p(\mathbf{X}, [\bar{\mathbf{Z}}]^\infty_{K_+}, \mathbf{A}, \pi) = \lim_{K \to \infty} \mathbf{C}_K^{K_+} p(\mathbf{X}, \mathbf{Z}_+, \mathbf{A}_+, \boldsymbol{\pi}_+) p(\mathbf{Z}_0, \boldsymbol{\pi}_0) p(\mathbf{A}_0)$ and we set $q(\mathbf{Z}_0, \boldsymbol{\pi}_0) = p(\mathbf{Z}_0, \boldsymbol{\pi}_0)$ and $q(\mathbf{A}_0) = p(\mathbf{A}_0)$. We denote the above lower bound by $L(q; K_+)$ to emphasize that it depends on the value of K_+.

The details for calculating the lower bound $L(q; K_+)$ are shown in the appendix. Note that if we apply a variational lower bound without using the equivalence classes on $\mathbf{Z}$, the lower bound becomes negatively infinite when $K \to \infty$ since $p(\mathbf{Z}_+, \boldsymbol{\pi}_+) \to 0$. Since $\mathbf{Z}_+$ is defined to contain only non-zero columns but our factorized $q(z_{ik})$ does not impose this constraint, the above inequality holds only approximately. Since the dataset that we apply IMF to has at least dozens of, if not hundreds or thousands (or more) of, data points, $q(\mathbf{z}_{:,k} = \mathbf{0}) = \prod_{i=1}^N q(z_{ik} = 0)$ is a very small value. Therefore, the approximate lower bound is a very accurate approximation to the exact lower bound. This is confirmed by our empirical results (e.g., see in Figure 3 where N=50).

Maximizing the lower bound $L(q; K_+)$, we obtain the following iterative updates for $k \le K_+$:

$$q(\pi_k) = \mathrm{Beta}(\hat{\alpha}_k, \hat{\beta}_k) \quad (7)$$

$$q(z_{ik} = 1) = \frac{\lambda_{ik}}{\lambda_{ik} + 1} \quad (8)$$

$$q(\tilde{\mathbf{a}}_{:,j}) = \mathcal{N}(\tilde{\mathbf{a}}_{:,j} | \mathbf{m}_j, \mathbf{V}_j) \quad (9)$$

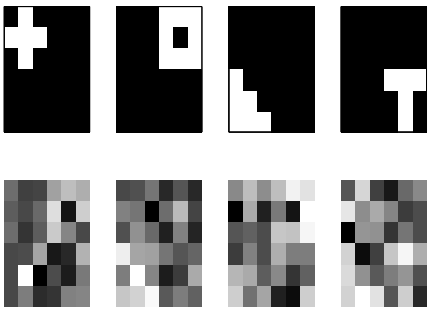

Figure 1: Illustration of the synthetic image data. The first row shows four latent images. The second row are examples of the observed images that are random combinations of the latent images with Gaussian noise (variance =0.5).

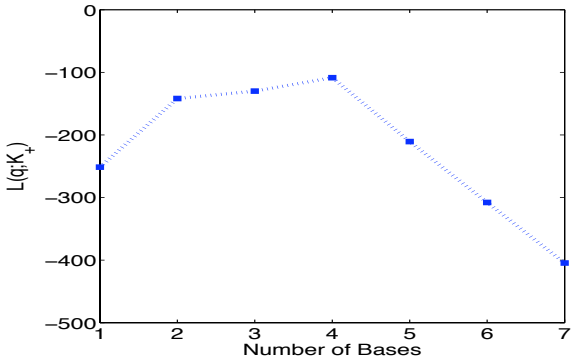

Figure 2: The approximate marginal model likelihood $L(q; K_+)$ on image data. By maximizing $L(q; K_+)$, the variational infinite matrix factorization discovers the true number of latent images, which is 4.

where

$$\hat{\alpha}_k = \sum_i q(z_{ik} = 1)$$

$$\hat{\beta}_k = N - \hat{\alpha}_k + 1$$

$$\mathbf{V}_j = \left((\mathbf{V}_j^0)^{-1} + \frac{1}{\sigma_x^2} \sum_i \mathbf{E}_{q(\mathbf{Z}_+)}[\tilde{\mathbf{z}}_{i,:}^T \tilde{\mathbf{z}}_{i,:}]\right)^{-1} \tag{10}$$

$$\mathbf{m}_j = \mathbf{V}_j((\mathbf{V}_j^0)^{-1}\mathbf{m}_j^0 + \frac{1}{\sigma_x^2} \sum_i x_{ij} \mathbf{E}_{q(\mathbf{Z}_+)}[\tilde{\mathbf{z}}_{i,:}]) \tag{11}$$

$$\lambda_{ik} = \exp\left[\Psi(\hat{\alpha}_k) - \Psi(\hat{\beta}_k) - \frac{1}{2\lambda_x^2} \sum_j \left(2 \sum_{r \neq k} (q(z_{ir} = 1)\Lambda_{rk}^j) + \Lambda_{kk}^j - 2x_{ij} m_j(k)\right)\right]$$

where $\Lambda^j = \mathbf{V}_j + \mathbf{m}_j \mathbf{m}_j^T$, $m_j(k)$ is the k^{th} element of $\mathbf{m}_j$ and $\Psi(\cdot)$ denotes the digamma function. For all $k > K_+$, $q(\mathbf{z}_{:,k} = 0, \pi_k = 0) = 1$.

Now we address the issue of how to K_+. To this end, we maximize the lower bound $L(q; K_+)$ over K_+. Since $L(q; K_+)$ approximates the marginal likelihood $p(\mathbf{X}|K_+) \propto p(K_+|\mathbf{X})p(\mathbf{X})$, the maximization over K_+ is justified by Bayesian evidence maximization and does not lead to overfitting. In practice, the algorithm is initialized with $K_+ = 1$. When the updates converge for the current K_+, we increase K_+ by one and use the current approximate posteriors to initialize the next iterations. We stop increasing K_+ when the approximate marginal model likelihood (i.e., evidence) $L(q; K_+)$ stops to increase with a bigger K_+.

To illustrate how evidence maximization is used to choose K_+, we apply infinite matrix factorization on the synthetic data that was used by Griffiths and Ghahramani (2005). First, we define four latent images (each image corresponds a row of $\mathbf{A}$ and each element of the latent image is either 1 or 0, as shown in the first row of figure 1) and generate a 50 by 4 loading matrix $\mathbf{Z}$ by randomly sampling its elements with $p(z_{ik} = 0.5)$. Then we combine the latent images $\mathbf{A}$ based on $\mathbf{Z}$ and add the Gaussian noise with mean 0 and variance 0.5 to generate 50 observed image data $\mathbf{X}$ (see the second row of figure 1). We plot the variational lower bound $L(q; K_+)$ for different values of K_+ in figure 2. The maximal value of $L(q; K_+)$ indicates $K_+ = 4$, which matches the true number of the latent images. In other words, infinite matrix factorization finds the true number of the latent images.

2.4 Learning variance parameters σ_X and σ_A

In the previous section, we assume σ_X (the variance of the observation noise in (4)) is known and we predefine the hyperparameter, σ_A, in the prior distribution (1). Now we extend our model to estimate σ_X and σ_A from data. Specifically, we assign inverse-Gamma prior distributions over σ_X and σ_A to represent the uncertainty about them and approximate their posterior distributions via variational inference. Using the inverse-Gamma priors is equivalent to using Gamma priors on the reciprocals of σ_X and σ_A, $\eta_X \triangleq 1/\sigma_X^2$ and $\eta_A \triangleq 1/\sigma_A^2$:

$$p(\eta_X) = \text{Gamma}(\eta_X | a_X^0, b_X^0) \quad p(\eta_A) = \text{Gamma}(\eta_A | a_A^0, b_A^0)$$

where Gamma$(\cdot)$ is a Gamma distribution and a_X^0, a_X^0, a_X^0, and a_X^0 are set to make the hyper-priors noninformative.

Given these priors, we obtain the following variational updates for $q(\eta_X)$ and $q(\eta_A)$:

$$q(\eta_X) = \text{Gamma}(\eta_X | \hat{a}_X, \hat{b}_X) \quad q(\eta_A) = \text{Gamma}(\eta_A | \hat{a}_A, \hat{b}_A)$$

where

$$\hat{a}_X = a_X^0 + \frac{ND}{2}$$
$$\hat{b}_X = b_X^0 + \frac{1}{2}\sum_{ij}\big(x_{ij}^2 - 2x_{ij}\mathbf{E}_q[\tilde{\mathbf{z}}_{i,:}]\mathbf{E}_q[\tilde{\mathbf{a}}_{:,j}] + \mathbf{E}_q[\tilde{\mathbf{z}}_{i,:}]\mathbf{E}_q[\tilde{\mathbf{a}}_{:,j}\tilde{\mathbf{a}}_{:,j}^T]\mathbf{E}_q[\tilde{\mathbf{z}}_{i,:}^T]\big)$$
$$\hat{a}_A = a_A^0 + \frac{KD}{2}$$
$$\hat{b}_A = b_A^0 + \frac{1}{2}\sum_{kj}\mathbf{E}_q[\tilde{a}_{kj}^2]$$

The expectations of η_X and η_A are $\mathbf{E}_q[\eta_X] = \hat{a}_X/\hat{b}_X$ and $\mathbf{E}_q[\eta_A] = \hat{a}_A/\hat{b}_A$. For the expanded model, we change the variational updates (10) and (11) by replacing $1/\sigma_X$ and $1/\sigma_A$ with $\mathbf{E}_q[\eta_X]$ and $\mathbf{E}_q[\eta_A]$. We also modify the variational lower bound $L(q; K_+)$ accordingly.

3 Infinite nonnegative matrix factorization

For certain applications such as image decomposition, imposing nonnegativity constraints to the factorized matrices leads to clearer model interpretation and improved predictive power (Lee and Seung, 2001). To increase the utility of Bayesian matrix factorization, we extend the infinite matrix factorization model by imposing nonnegative constraints on $\mathbf{A}$. We call this new model Infinite Nonnegative Matrix Factorization (INMF).

For IMFs, we assign a factorized Gaussian prior over $\mathbf{A}$. For INMFs, we change this prior to a factorized truncated Gaussian distribution:

$$p(\mathbf{A}) \propto \prod_{k,j}\mathcal{N}(a_{kj}|0,\sigma_A^2)I(a_{kj} \geq 0) \tag{12}$$

where $I(\cdot)$ is an indicator function. This truncated Gaussian prior not only regularizes $\mathbf{A}$ to prevent overfitting as an L2-Regularizer but also effectively imposes nonnegative constraints on elements of $\mathbf{A}$.

Given the truncated Gaussian prior over $\mathbf{A}$, we cannot apply variational methods directly since the lower bound (6) becomes negatively infinite. To address this issue, we use the Power-Expectation Propagation (Power-EP) framework (Minka, 2004) to approximate the exact posterior.

3.1 Power-EP inference for INMF

For Power-EP, we choose the same form for the approximate posteriors q as before.Similar to the IMF case, we match $q(\mathbf{A}_0)q(\mathbf{Z}_0, \boldsymbol{\pi}_0)$ to the exact distributions. Therefore, we only need to approximate $q(\mathbf{Z}_+)q(\boldsymbol{\pi}_+)q(\mathbf{A}_+)$ based on $p(\mathbf{Z}_+, \mathbf{A}_+, \boldsymbol{\pi}_+|\mathbf{X})$. Now for $q(\mathbf{A}_+)$, we let

$$q(\mathbf{A}_+) \propto \tilde{p}_0(\mathbf{A}_+)\tilde{p}_X(\mathbf{A}_+) \tag{13}$$

where $\tilde{p}_X(\mathbf{A}_+)$ and $\tilde{p}_0(\mathbf{A}_+)$ are the Gaussian messages from the likelihood $p(\mathbf{X}|\mathbf{Z}_+, \mathbf{A}_+, \boldsymbol{\pi}_+)$ and the prior $p(\mathbf{A}_+)$ to the variable $\mathbf{A}_+$, respectively. Since both messages have the form of Gaussians, $q(\mathbf{A}_+)$ is a Gaussian distribution.

Power-EP (Minka, 2004) generalizes expectation propagation and variational inference (in particular, variational message passing (Winn and Bishop, 2004)) using a flexible $\alpha-$divergence. This divergence includes $KL(p\|q)$ and $KL(q\|p)$ as special cases. Using Power-EP, we have three steps for processing a factor in the joint distribution of the model: i) in the deletion step we compute the partial posterior/belief after removing the message from this factor; ii) in the projection step we minimize an α-divergence (e.g., $KL(p\|q)$ or $KL(q\|P)$) to obtain the new posterior q; and iii) in the message-update step, we update the message to be the ratio of the new posterior q and the partial belief computed in the deletion step. We then iteratively process all the factors in the joint distribution with these three steps. For INMFs, we minimize $KL(p\|q)$ when processing the prior factor $p(\mathbf{A}_+)$, and minimize $KL(q\|p)$ when processing the the likelihood and the other priors.

In the deletion step, to process the likelihood and the priors on $\mathbf{Z}_+$ and $\boldsymbol{\pi}_+$, we compute the partial belief $q^{\setminus X}(\mathbf{A}_+) \propto q(\mathbf{A}_+)/\tilde{p}_x(\mathbf{A}_+)$. Because of (13), we have

$$q^{\setminus X}(\mathbf{A}_+) \propto \tilde{p}_0(\mathbf{A}_+) \tag{14}$$

Since $q(\mathbf{Z}_+)$ and $q(\boldsymbol{\pi}_+)$ are totally defined by the messages from the likelihood and the prior on $\mathbf{Z}$, the partial beliefs $q^{\setminus X}(\mathbf{Z}_+)$ and $q^{\setminus X}(\boldsymbol{\pi}_+)$ are 1 after removing these messages.

Then in the projection step, we minimize the exclusive KL divergence over the new approximate posteriors:

$$KL(q(\mathbf{Z}_+)q(\boldsymbol{\pi}_+)q(\mathbf{A}_+)\|p(\mathbf{X}|\mathbf{Z}_+, \mathbf{A}_+, \boldsymbol{\pi}_+)p(\mathbf{Z}_+, \boldsymbol{\pi}_+)q^{\setminus X}(\mathbf{A}_+)).$$

By replacing $p(\mathbf{A}_+)$ with its approximation $q^{\setminus X}(\mathbf{A}_+)$ in the variational updates described in Section 2, we can efficiently minimize the above KL divergence.

To update the message update, it normally requires the computation of the messages from the likelihood and the priors on $\mathbf{Z}_+$ and $\boldsymbol{\pi}_+$ to the variables $(\mathbf{A}, \mathbf{Z}_+, \&\boldsymbol{\pi}_+$. But since here we do not need the messages explicitly in the deletion step to obtain all the partial beliefs, we can save the computation of these messages.

When processing the truncated Gaussian prior $p(\mathbf{A}) = \prod p(a_{kj})$, we only update $q(\mathbf{A}_+)$ since this prior does

not involve $\mathbf{Z}_+$ and $\boldsymbol{\pi}_+$. The deletion step computes the partial belief

$$q^{\backslash kj(0)}(\tilde{\mathbf{a}}_{:,j}) = \mathcal{N}(\tilde{\mathbf{a}}_{:,j}|\mathbf{m}_{\backslash kj(0)}, \mathbf{V}_{\backslash kj(0)}) \propto q(\tilde{\mathbf{a}}_{:,j})/\tilde{p}_0(a_{kj})$$

where $\tilde{p}_0(a_{kj} \propto \mathcal{N}(\mathbf{m}_{kj(0)}, \mathbf{V}_{kj(0)})$ (i.e., this message is a $K_+ \times 1$ dimensional Gaussian from a_{kj} to $\tilde{\mathbf{a}}_{:,j}$) and

$$\mathbf{V}_{\backslash kj(0)} = (\mathbf{V}_j^{-1} - \mathbf{V}_{kj(0)})^{-1} \quad (15)$$

$$\mathbf{m}_{\backslash kj(0)} = \mathbf{V}_{\backslash kj(0)}(\mathbf{V}_j^{-1}\mathbf{m}_j - \mathbf{V}_{kj(0)}^{-1}\mathbf{m}_{kj(0)}) \quad (16)$$

The projection step gives the new posterior

$$q^{new}(\tilde{\mathbf{a}}_{:,j}) \propto \mathcal{N}(\tilde{\mathbf{a}}_j|\mathbf{m}_j^{new}, \mathbf{V}_j^{new}),$$

by minimizing the inclusive $KL(\hat{p}\|q)$ where $\hat{p} \propto q^{\backslash kj(0)}(\tilde{\mathbf{a}}_{:,j})p(a_{kj})$. The following moment matching equations solve the minimization problem:

$$\mathbf{m}_j^{new} = \mathbf{m}_w + \alpha \mathbf{V}_w \mathbf{e}_k \quad (17)$$

$$\mathbf{V}_j^{new} = \mathbf{V}_w - \frac{\alpha m_j^{new}(k)}{v_w(k,k)} \mathbf{V}_w \mathbf{e}_k \mathbf{e}_k^T \mathbf{V}_w^T \quad (18)$$

where

$$\mathbf{V}_w = ([\mathbf{V}_{\backslash kj(0)}]^{-1} + \frac{1}{\sigma_A^2}\mathbf{e}_k\mathbf{e}_k^T)^{-1}$$

$$\mathbf{m}_w = \mathbf{V}_w([\mathbf{V}_{\backslash kj(0)}]^{-1}\mathbf{m}_{\backslash kj(0)})$$

$$\alpha = \frac{1}{v_w(k,k)^{1/2}} \frac{\mathcal{N}(v_w(k,k)^{-1/2}m_w(k)|0,1)}{\int_{-\infty}^{v_w(k,k)^{-1/2}m_w(k)} \mathcal{N}(z|0,1)\mathrm{d}z}$$

To update the message, we take the ratio between $q^{new}(\tilde{\mathbf{a}}_{:,j})$ and $q^{\backslash kj(0)}(\tilde{\mathbf{a}}_{:,j})$ and obtain the new message $\tilde{p}_0(a_{kj}$ with

$$\mathbf{V}_{kj(0)}^{new} = ((\mathbf{V}_j^{new})^{-1} - (\mathbf{V}_{\backslash kj(0)})^{-1})^{-1} \quad (19)$$

$$\mathbf{m}_{kj(0)}^{new} = \mathbf{V}_{kj(0)}^{new}((\mathbf{V}_j^{new})^{-1}\mathbf{m}_j^{new} - (\mathbf{V}_{\backslash kj(0)})^{-1}\mathbf{m}_{\backslash kj(0)}) \quad (20)$$

Note that the deletion step and message update can be further simplified by low-rank matrix operations. To learn K_+ for the INMF model, we maximize the approximate marginal likelihood $L(q; K_+)$. We also extend the model to estimate the posteriors of σ_X and σ_A, just like what we have described in Section 2.4. The IMF and INMF are summarized in Algorithm 1.

4 Experiments

To evaluate the proposed methods, we test them on three tasks: synthetic image decomposition, collaborative filtering, and role mining.

4.1 Image Decomposition

First, we compare IMFs and INMFs with Gibbs sampling (Wood and Griffiths, 2007), Particle Filtering (PF) (Wood and Griffiths, 2007), Variational Finite Matrix Factorization (VFMF), and the variational infinite matrix factorization (TVIMF) (Doshi-Velez et al., 2009) on a synthetic dataset.

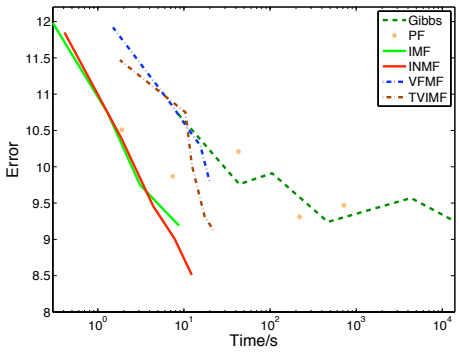

Figure 3: Performance comparison on image data.

We use a similar data generation process as described in Section 2.3 and by (Griffiths and Ghahramani, 2005). The only difference is that now we randomly generate each of the 6-by-6 latent images that contain 8 elements of 1's (the other elements are 0's).

Figure 3 shows the performance of all these six algorithms. For the evaluation, we interpret each row of the latent matrix $\mathbf{A}$ as a base (i.e.,$\mathbf{a}_{k,:}$). The errors in the figure are measured by the mean square difference between each latent base, $\mathbf{a}_{k,:}$, and the estimated base that is closest to it. As shown in Figure 3, our new methods converge much faster than the alternative methods, demonstrating their high computational efficiency. The results of particle filtering (for which we vary the number of particles from 100 to 400, 1000, 2500, 5000) fluctuate heavily. Sometimes the particle filtering method even ends up with over 50 bases as the final estimation, while the true number of bases

Algorithm 1 IMF and INMF

1. Initialize $K_+ = 1$.
2. Initialize all approximate factors.
3. For each variational inference iteration:
   a) Loop over $k = 1, \ldots, K_+$:
      Update $q(\pi_k)$ via (7)
   b) Loop over $i = 1, \ldots, N, k = 1, \ldots, K_+$:
      Update $q(\mathbf{z}_{ik})$ via (8)
   c) Loop over $j = 1, \ldots, D$:
      For IMF: update $q(\mathbf{a}_j)$ via (9)
   For INMF:
      Loop over $j = 1, \ldots, D, k = 1, \ldots, K_+$:
      Update $q(\mathbf{a}_j)$ via (15) to (20)
   d) Compute the evidence $L(q; K_+)$
   e) Increase K_+ if $L(q; K_+)$ increases.

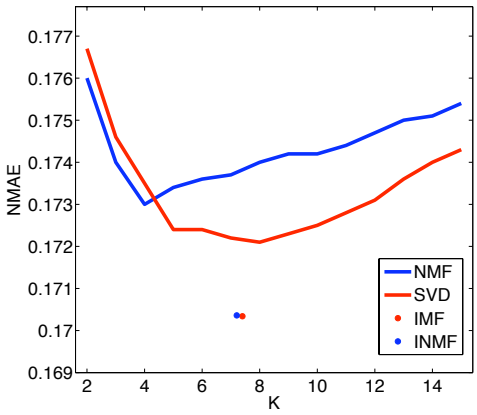

Figure 4: Prediction errors for collaborative filtering. The plot shows Normalized Mean Absolute Error (NMAE) of four models on the Jester data. The results are averaged over 10 experiments.

is only 4. For TVIMF and VFMF, the truncation level T has to be sufficiently large (we varied it from 4 to 10) to achieve low estimate error. This makes this variational approach less efficient compared to our method. The performance of TVIMF and VFMF are not stable in practice. Therefore, we run them multiple times and pick the best solution from these runs. By contrast, the IMF and INMF work stably and the quality of the approximate lower bound is very good: IMF and INMF never over-estimate K_+ in our experiments. IMF achieves the lowest prediction error among all the approximate inference methods (except INMF) for IBP models. Furthermore, the INMF leads to highest prediction accuracy among all the methods, demonstrating the effectiveness of the nonnegativity constraints on latent matrices.

4.2 Collaborative Filtering

We apply the new methods to a collaborative filtering task to test how accurate they are when predicting the preference of a specific user given his previous ratings. We use a subset of the Jester dataset Goldberg et al. (2001). The dataset contains 100 jokes with 73421 user ratings. The density (or fraction of the rating matrix that is filled) of the Jester set is about 0.5. We select 1000 users and for each selected user, 30 ratings are held out for testing. We select 1000 users and for each selected user, 30 ratings are held out for testing. The experiment is repeated for 10 times, each time with a different user subset.

IMF automatically discovers 7.8 latent bases averaged over the 10 experiments. INMF gives 7.6 latent bases on average. For comparison, VFMF, TVIMF, classical nonnegative matrix factorization (NMF), and singular value decomposition (SVD) are applied to the same training and testing data. We did not apply Gibbs samplers and particle filtes because of their limited scalability for large datasets. NMF and SVD require a predefined number of bases. We vary the number of bases from 2 to 15 and choose this number by optimizing their performance. The truncation level of VFMF and TVIMF is 30. Empirically, a smaller truncation level leads to much worse prediction performance.

The results of NMF, SVD, IMF and INMF are shown in figure 4. We do not plot the results of VFMF and TVIMF since their results are out of the boundary of this figure. VFMF achieves 0.1767 with 30 effective bases, and TVIMF achieves 0.1761 with 30 effective bases. IMF and INMF not only learn a more compact representation of the data, but also give more accurate predictions than the other methods.

4.3 Role Mining

Finally, we apply our new methods to the role mining problem. Role mining is an active research area in information security that aims at discovering a set of roles for role-based access control (RBAC) from an existing user-permission assignment relation. In RBAC, instead of assigning permissions directly to users, an administrator assigns permissions to roles and authorizes users to roles. To simplify administration, it is desirable to keep the number of roles small.

We use two datasets, "Domino" and "Firewall" from researchers at HP Labs. The "Domino" dataset, from a Lotus Domino server, is a 79x231 binary matrix where each row is a user and each column is a permission for a given level of access to files, databases, and custom applications. The "Firewall" dataset is a 709x365 binary matrix from a Cisco firewall used to provide external users access to internal resources. In our experiment, we first transpose the original matrices, such that Z matrix becomes a meaningful binary role matrix, which assigns permissions to roles.

We randomly hold out 20% elements of each matrix for testing. Since the matrix is extremely sparse, we report the Area Under the Curve (AUC) of IMF, INMF, VFMF, TVIMF, NMF and SVD in Table 1. The truncation level T of VFMF and TVIMF is 20. NMF and SVD are chosent to have the optimal number of bases that gives to their best AUC value.

We find that TFMF, TVIMF, IMF and SVD have inferior performance on this dataset. By contrast, the AUC comparison shown in Table 1 suggests that INMF outperforms all the other algorithms on both datasets. For better illustration, we plot the ROC curve of "domino" dataset. We observe that INMF achieves the distinguishable True-Positive ratio when

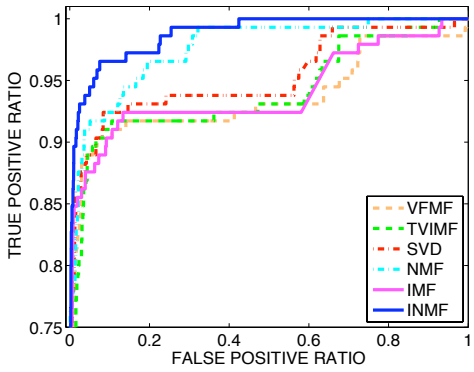

Figure 5: The ROC curve for the "Domino" dataset.

Algorithm	Domino		Firewall	
	K	AUC	K	AUC
VFMF	20	0.9339	20	0.9748
TVIMF	20	0.9382	20	0.9732
SVD	5	0.9518	10	0.9745
NMF	4	0.9755	6	0.9801
IMF	8	0.9400	7	0.9823
INMF	9	**0.9869**	9	**0.9853**

Table 1: AUC of VFMF, TVIMF, SVD, NMF, IMF and INMF on "Domino" and "Firewall" datasets.

the False-Positive ratio is small, which indicates the desirable performance on the top of the list.

5 Conclusions

Using the new equivalence classes of infinite binary matrices, we have developed two novel approximate inference algorithms for infinite matrix factorization and infinite nonnegative matrix factorization, respectively. These two methods are unified in the power-EP framework. They learn the model dimensionality automatically from the data, as well as all the model hyper-parameters. Unlike the previous variational method (Doshi-Velez et al., 2009), the new methods do not use a truncated stick-breaking representation (so no truncation levels to be set) and achieve faster convergence and lower prediction error rates compared to alternative inference methods.

References

Griffiths TL, Ghahramani Z (2005) Infinite latent feature models and the Indian buffet process. Technical Report 2005-001, Gatsby Computational Neuroscience Unit, University College London.

Doshi-Velez F, Miller KT, Gael JV, Teh YW (2009) Variational inference for the Indian buffet process. In Proceedings of AISTATS.

Lee DD, Seung HS (2001) Algorithms for non-negative matrix factorization. In Advances in Nerual Information Processing 13. MIT Press.

Devarajan K (2008) Nonnegative matrix factorization: An analytical and interpretive tool in computational biology. PLoS Computational Biology 4: 1–12.

Minka T (2004) Power EP. Technical Report 2004-149, Microsoft Research Ltd.

Teh YW, Gorur D, Ghahramani Z (2007) Stick-breaking construction for the Indian buffet process. In Proceedings of the International Conference on Artificial Intelligence and Statistics. volume 11.

Winn J, Bishop CM (2004) Variational message passing. Journal of Machine Learning Research 5.

Wood F, Griffiths TL (2007) Particle filtering for nonparametric Bayesian matrix factorization. In Advances in Nerual Information Processing 19. MIT Press.

Goldberg K, Roeder T, Gupta D, Perkins C (2001) Eigentaste: A constant time collaborative filtering algorithm. Information Retrieval 4: 133–151.

A Lower bound of marginal likelihood $L(q; K_+)$

The lower bound $L(q; K_+)$ is given here.

$$L(q; K_+) = T_1 + T_2 + T_3 - H_1 - H_2 - H_3$$

where

$$T_1 = \sum_{i=1}^{N}\sum_{j=1}^{D}\Big[-\frac{1}{2}\ln(2\pi\sigma_x^2) + \frac{x_{ij}}{2\sigma_x^2}\sum_{k=1}^{K_+} q(z_{ik}=1)m_j(k) - \frac{1}{2\sigma_x^2}\sum_{r=1}^{K_+}\sum_{k=1}^{K_+}\mathbf{E}_{q(\mathbf{Z}_+)}(z_{ir}z_{ik})\Lambda^j_{rk} - \frac{1}{2\sigma_x^2}x_{ij}^2\Big]$$

$$T_2 = \sum_{j=1}^{D}\Big[-\frac{K_+}{2}\ln(2\pi\sigma_A^2) - \frac{1}{2\sigma_A^2}\sum_{k=1}^{K_+}\Lambda^j_{kk}\Big]$$

$$T_3 = \sum_{k\le K_+}\Big[(\sum_i q(z_{ik}=0))\big(\Psi(\hat\beta_k) - \Psi(\hat\alpha_k+\hat\beta_k)\big) + (\sum_i q(z_{ik}=1)-1)\big(\Psi(\hat\alpha_k) - \Psi(\hat\alpha_k+\hat\beta_k)\big) + \ln\frac{\alpha}{k}\Big]$$

$$H_1 = \sum_{j=1}^{D}\Big[-\frac{1}{2}\ln|\mathbf{V}_j| - \frac{K_+}{2}(1+\ln(2\pi))\Big]$$

$$H_2 = \sum_{k\le K_+}\Big[\sum_i q(z_{ik}=1)\ln q(z_{ik}=1) + q(z_{ik}=0)\ln q(z_{ik}=0)\Big]$$

$$H_3 = \sum_{k\le K_+}\int q(\pi_k)\ln q(\pi_k)\mathrm{d}\pi_k = \sum_{k\le K_+}\Big[(\hat\alpha_k-1)\{\psi(\hat\alpha_k) - \psi(\hat\alpha_k+\hat\beta_k)\} + (\hat\beta_k-1)\{\psi(\hat\beta_k) - \psi(\hat\alpha_k+\hat\beta_k)\} - \ln\Gamma(\hat\alpha_k+\hat\beta_k) + \ln\Gamma(\hat\alpha_k) + \ln\Gamma(\hat\beta_k)\Big]$$

where $\Lambda^j = \mathbf{V}_j + \mathbf{m}_j\mathbf{m}_j^T$, $\mathbf{m}_j$ and $\mathbf{V}_j$ are the mean and the covariance matrix of $q(\tilde{\mathbf{a}}_j)$ respectively, and $m_j(k)$ is the k-th element of $\mathbf{m}_j$.