Nonparametric Bayesian Matrix Completion

Mingyuan Zhou, Chunping Wang, Minhua Chen, John Paisley, [1]David Dunson and Lawrence Carin
Electrical and Computer Engineering Department and [1]Statistics Department
Duke University
Durham, NC 27708-0291
Email: {mz31,chunping.wang,minhua.chen,jwp4,lawrence.carin}@duke.edu, dunson@stat.duke.edu

Abstract—**The Beta-Binomial processes are considered for inferring missing values in matrices. The model moves beyond the low-rank assumption, modeling the matrix columns as residing in a nonlinear subspace. Large-scale problems are considered via efficient Gibbs sampling, yielding predictions as well as a measure of confidence in each prediction. Algorithm performance is considered for several datasets, with encouraging performance relative to existing approaches.**

I. Introduction

There has been significant recent interest in collaborative filtering for matrix completion (see for example [1], [2], [3], [4], [5]). Like the cited papers, we view this from a Bayesian perspective. We note, however, that there has also been recent work viewing this problem from the standpoint of non-Bayesian optimization [6]. Theoretical guarantees have also been recently investigated, most of these based on non-Bayesian formulations [7]. Another aspect of our work that is gaining recent attention is the use of auxiliary information, beyond the matrix data, when performing inference [6], [5].

We wish to develop a nonlinear model, and do so in the form of a *union* of subspaces. To infer the union of supspaces, we employ the Indian Buffet process (IBP) [8], implemented using a truncated Beta-Bernoulli process construction [9], [10]. Further, we wish to consider auxiliary data, with the manner in which we do this related to the simplified form in [5]. We consider three different ways in which the IBP may be employed for this problem, one of which is related to but distinct from a previous use of the IBP for matrix analysis [11] (we discuss the relationships in detail below).

II. Low-Rank Representation

We consider a model very similar to the PMF developed in [2], [3], with an alternative method for inferring the latent-space dimension D, explicitly imposing a low-rank-favoring prior. The real matrix is assumed represented as

$$\boldsymbol{X} = \sum_{k=1}^{K} (\lambda_k z_k) \boldsymbol{u}_k \boldsymbol{v}_k^T + \boldsymbol{E} \tag{1}$$

where $\boldsymbol{u}_k \in \Re^N$, $\boldsymbol{v}_k \in \Re^M$, $\lambda_k \in \Re$, $\boldsymbol{E} \in \Re^{N\times M}$, and $z_k \in \{0,1\}$. Analogous to [2], [3], $\boldsymbol{u}_k \sim \mathcal{N}(0, \frac{1}{N}\boldsymbol{I}_N)$ and $\boldsymbol{v}_k \sim \mathcal{N}(0, \frac{1}{M}\boldsymbol{I}_M)$, where $\boldsymbol{I}_N$ is the $N \times N$ identity matrix. Note that the columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ have unit expected ℓ_2 norm, with the amplitudes absorbed in $\lambda_k \sim \mathcal{N}(0, \beta^{-1})$, with a gamma hyper-prior typically placed on β. It is possible to explicitly impose that the $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$ are orthonormal [12], but this has proven unnecessary (we have found in our experiments that using $\boldsymbol{U}$ and $\boldsymbol{V}$ to define a linear subspace is sufficient); the imposition of orthonormality comes at significant computational cost. Each component of $\boldsymbol{E}$ is drawn iid from $\mathcal{N}(0, \alpha_0^{-1})$, with a separate gamma hyper-prior employed for α_0.

The main distinction with [2], [3] is that here we explicitly impose the belief that $\boldsymbol{X}$ is approximately low rank (*approximate* because of the presence of $\boldsymbol{E}$). Specifically, $z_k \sim \pi_k$, with $\pi_k \sim \text{Beta}(a/K, b(K-1)/K)$. Through the choice of a and b we impose our prior belief about the rank of $\boldsymbol{X}$. Specifically, by marginalizing out the vector $\{\pi_1, \ldots, \pi_K\}$, one may show that the number of $\{z_k\}_{k=1,K}$ equal to one is distributed $\text{Binomial}(K, a/(a+b(K-1)))$, and the expected number of ones is $aK/[a+b(K-1)]$. As $K \to \infty$, the number of non-zero z_k is drawn from $\text{Poisson}(a/b)$. Hence, by setting, a, b, and K, one is making explicit prior statements about the approximate rank of $\boldsymbol{X}$, and posterior inference yields the estimated rank based on the observed data.

III. Sparseness & Union of Subspaces

Letting $\boldsymbol{X}_{:,m}$ represent the mth column of $\boldsymbol{X}$, from (1) we have

$$\boldsymbol{X}_{:,m} = \sum_{k=1}^{K} (\lambda_k z_k) v_{km} \boldsymbol{u}_k + \boldsymbol{E}_{:,m} \tag{2}$$

where v_{km} is the mth component of $\boldsymbol{v}_k$. This shows that each columns of $\boldsymbol{X}$ resides in a subspace spanned by the columns of $\boldsymbol{U}\boldsymbol{\Lambda}^{\frac{1}{2}}$, where $\boldsymbol{\Lambda} = \text{diag}\{\lambda_1^2 z_1, \ldots, \lambda_K^2 z_K\}$. Consequently, conditionally on $\boldsymbol{U}$, $\boldsymbol{\Lambda}$ and α_0, each column of $\boldsymbol{X}$ is drawn

$$\boldsymbol{X}_{:,m} \sim \mathcal{N}(\boldsymbol{0}, M^{-1}\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T + \alpha_0^{-1}\boldsymbol{I}_N) \tag{3}$$

A similar relationship holds with respect to the rows of $\boldsymbol{X}$ and the subspace defined by the columns of $\boldsymbol{V}\boldsymbol{\Lambda}^{\frac{1}{2}}$. A related form was used in the simplified model employed in [5], with each row/column drawn from a Gaussian with fixed covariance/subspace (but with the covariance drawn from an inverse-Wishart prior, rather than via a factor model).

Motivated by the goal of removing the assumption that each row/column of $\boldsymbol{X}$ resides in a fixed linear subspace, [5], [4] employed different GP methodologies to manifest a nonlinear model. Below we focus on a column expansion, recognizing we may also do this with the rows. Further, we consider an alternative approach for removing the linearity assumption: we assume that the columns of $\boldsymbol{X}$ are drawn from a *union* of linear subspaces, with the union manifesting a *nonlinear* subspace in $\Re^N$.

Motivated by an Indian buffet process (IBP) interpretation [8], assume we draw K "dishes" $\{\boldsymbol{u}_k\}_{k=1,K}$ in the manner discussed above. Further, we draw the K probabilities $\{\pi_k\}_{k=1,K}$ also as above. However, rather than having all columns of $\boldsymbol{X}$ expanded in the same subspace defined by $\boldsymbol{U}\boldsymbol{\Lambda}^{\frac{1}{2}}$, we consider a separate $\boldsymbol{\Lambda}_m$ for each column $m \in \{1, \dots, M\}$. Specifically, rather than drawing a single binary set $\{z_k\}_{k=1,K}$, we draw a separate such set for each column. Hence, for column $m \in \{1, \dots, M\}$ we draw $z_k^{(m)} \sim \text{Bern}(\pi_k)$, and $\boldsymbol{\Lambda}_m = \text{diag}(\lambda_1^2 z_1^{(m)}, \dots, \lambda_K^2 z_K^{(m)})$. The mth column is now represented

$$\boldsymbol{X}_{:,m} = \sum_{k=1}^{K} (\lambda_k z_k^{(m)}) v_{km} \boldsymbol{u}_k + \boldsymbol{E}_{:,m} \tag{4}$$

with $\lambda_k \sim \mathcal{N}(0, \beta_k^{-1})$, again with a gamma prior on β_k. To emphasize the distinction with the linear model, note

$$\boldsymbol{X}_{:,m} \sim \mathcal{N}(\boldsymbol{0}, M^{-1}\boldsymbol{U}\boldsymbol{\Lambda}_m\boldsymbol{U}^T + \alpha_0^{-1}\boldsymbol{I}_N) \tag{5}$$

Consider a random binary matrix $\boldsymbol{Z} \in \{0,1\}^{M \times K}$, with the mth row defined by the draws $\{z_1^{(m)}, \dots, z_K^{(m)}\}$. The M rows are "customers" at an "Indian buffet", and if $z_k^{(m)} = 1$ customer m tastes dish k, which here corresponds to column vector $\boldsymbol{u}_k$. This is the IBP model [8], with an implementation via a Beta-Bernoulli construction; the reader is referred to [9] for a detailed discussion of the relationship between the IBP and Beta-Bernoulli processes, with a related discussion in [10]. An advantage of the IBP construction is that theory exists for the statistics of $\boldsymbol{Z}$, and hence for the size of each subspace used to represent the columns of $\boldsymbol{X}$ and the diversity of these subspaces. The model may also be generalized further by considering a power-law IBP [13]; that is left for future research, but it shows the general modeling flexibility of the above construction.

There are additional ways in which the model may be extended for collaborative filtering. For example, let $\boldsymbol{z}^{(m)} = (z_1^{(m)}, \dots, z_K^{(m)})$. We may draw $\boldsymbol{z}^{(m)} \sim G$, with $G \sim \text{DP}(\gamma G_0)$, with $\text{DP}(\gamma G_0)$ representing a Dirichlet process (DP) with innovation parameter $\gamma > 0$ and base measure G_0 [14]. The base measure G_0 may be a Beta-Bernoulli prior of the form discussed above, manifesting a mixture model $G = \sum_{i=1}^{\infty} w_i \delta_{\boldsymbol{z}_i^*}$ with each $\boldsymbol{z}_i^* \sim \text{Bern}(\boldsymbol{\pi}_i)$ and $\pi_{ik} \sim \text{Beta}(a/K, b(K-1)/K)$ and $\sum_{i=1}^{\infty} w_i = 1$. In this construction the columns of $\boldsymbol{X}$ are drawn from a mixture of low-rank Gaussians. In our collaborative-filtering experiments (discussed further below) we found that the performance with the DP construction was essentially the same as that based on the simpler IBP construction above. However, the use of DP-like constructions may be of interest if one has prior information concerning the properties of the rows/columns of $\boldsymbol{X}$, of particular interest for columns/rows with very few observations [3]. One way to do this is to employ a *kernel* stick-breaking process [15]; we do not consider this here, but it is also an area for further exploration.

Before proceeding, we note that while this paper focuses on collaborative filtering, research related to the above model has been considered for image analysis, specifically for inpainting, denoising, and compressive sensing [16], [17]. In that case rather than considering rows/columns of a matrix $\boldsymbol{X}$, a dictionary analogous to $\{\boldsymbol{u}_k\}_{k=1,K}$ is learned for (typically) 8×8 blocks in the image. In [17] it was demonstrated that the IBP construction outlined above, applied to image processing, yields state-of-the-art performance. We here extend that model to collaborative filtering and related matrix applications, again demonstrating state-of-the-art performance (discussed in Section V). We believe this represents a statement about the general power of the IBP construction for learning dictionaries or (here) subspaces.

IV. Unifying Low-Rank & Sparseness

In some applications $M \gg N$ or $N \gg M$ (*e.g.*, the number of users is typically much larger than the number of movies in a rating system), and therefore the row/column based construction discussed above is computationally attractive (the union of subspaces is implemented with vectors corresponding to the smaller of M and N). However, for some problems it is desirable to treat the rows and columns symmetrically.

The column-based expansion discussed above may be represented $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}_R^T + \boldsymbol{E}$, where $\boldsymbol{U} \in \Re^{N \times K}$ represents the K column dictionary elements as above, and the sparse matrix $\boldsymbol{S}_R \in \Re^{M \times K}$, with mth column defined by $(z_1^{(m)} v_{1m}, \dots, z_K^{(m)} v_{Km})^T$. We may also manifest a similar construction with respect to the rows of $\boldsymbol{X}$, which implies that it should be possible to constitute a symmetric construction, with sparseness imposed simultaneously in the column and row expansion.

Toward that end, we assume the rows of $\boldsymbol{U}$ may also be expanded in a *distinct* union of subspaces, again implemented via the Beta-Bernoulli IBP construction. Specifically, we assume $\boldsymbol{U} = \boldsymbol{S}_L\boldsymbol{F}$, where $\boldsymbol{S}_L \in \Re^{N \times K'}$ and $\boldsymbol{F} \in \Re^{K' \times K}$. The sparse matrix $\boldsymbol{S}_L$ is constructed in the same manner as $\boldsymbol{S}_R$, with details omitted for brevity. We therefore have

$$\boldsymbol{X} = \boldsymbol{S}_L\boldsymbol{F}\boldsymbol{S}_R^T + \boldsymbol{E} \tag{6}$$

This model is closely related to that in [11], with two distinctions: (*i*) in [11] $\boldsymbol{S}_L$ and $\boldsymbol{S}_R$ were binary, where here they are real (but still sparse); and (*ii*), more importantly, we place an additional requirement on the "feature" matrix $\boldsymbol{F}$.

In [11], within the prior the elements of $\boldsymbol{F}$ are drawn iid from a Gaussian. We have found that we achieve better results on the collaborative-filtering data (movie ratings) if we add the additional assumption that $\boldsymbol{F}$ is low rank. To do this, we model $\boldsymbol{F}$ using the low-rank construction in Section II. Therefore, this symmetric model employs both sparseness (for $\boldsymbol{S}_L$ and $\boldsymbol{S}_R$) and low-rank (on $\boldsymbol{F}$) properties, unifying two distinct lines of research for collaborative filtering. The sparseness is imposed via the IBP, and the low-rank of $\boldsymbol{F}$ is imposed by a related "spike-slab" construction.

The representation $\boldsymbol{X} = \boldsymbol{S}_L\boldsymbol{F}\boldsymbol{S}_R^T + \boldsymbol{E}$ may be interpreted as follows. The sparse matrix $\boldsymbol{S}_L \in \Re^{N \times K}$ may be viewed as representing latent features for the rows of $\boldsymbol{X}$: row n of $\boldsymbol{S}_L$ corresponds to a sparse set of latent features associated with the entity associated with the nth row of $\boldsymbol{X}$. Similarly, the mth row of $\boldsymbol{S}_R \in \Re^{M \times K}$ represents a sparse set of latent features associated with the entity associated with the mth column of $\boldsymbol{X}$. The coordinates of these features are defined by $\boldsymbol{U}$ and $\boldsymbol{V}$, but neither of these is *explicitly* used here. The (typically small) matrix $\boldsymbol{F}$ provides a mapping from the nth row of $\boldsymbol{S}_L$ and the mth column of $\boldsymbol{S}_R$ to component (n, m) of $\boldsymbol{X}$.

A. Auxiliary information

In many collaborative-filtering problems we may have feature vectors $\boldsymbol{r}_n \in \Re^{J_r}$, for $n \in \{1, \ldots, N\}$, with $\boldsymbol{r}_n$ representing observed covariates associated with the nth row of $\boldsymbol{X}$. Similarly, we may have $\boldsymbol{c}_m \in \Re^{J_c}$ for $m \in \{1, \ldots, M\}$, representing features of the M columns. We would like to use this auxiliary information when building the model. Several methods were considered for employing this auxiliary information, and for brevity we only discuss the method that yielded consistently best results. Specifically, consider the model

$$\boldsymbol{X} = \boldsymbol{S}_L \boldsymbol{F} \boldsymbol{S}_R^T + \boldsymbol{C}_L \boldsymbol{G}_L^T + \boldsymbol{G}_R \boldsymbol{C}_R^T + \boldsymbol{E} \tag{7}$$

where $\boldsymbol{G}_L \in \Re^{M \times J_c}$ and $\boldsymbol{G}_R \in \Re^{N \times J_r}$ are matrices to be inferred. The term $\boldsymbol{S}_L \boldsymbol{F} \boldsymbol{S}_R^T$ may be replaced with any of the methods discussed above for matrix analysis.

B. Inference methods

The three methods discussed above are implemented using Gibbs sampling. The conditional density functions used to implement the sampling are analytic for each of the models. The analysis is relatively computationally efficient (run on PCs using non-optimized Matlab software), with further details on computation times discussed below when presenting results for each of the datasets considered in our experiments.

V. Example Results

A. Movie ratings

For a fair comparison, with the same test settings, we conducted both the Weak and Strong generalization leave-one-out-per-user tests on both the 1M MovieLens and EachMovie datasets, as defined by Marlin [18] and followed by [1], [19], [4]. In [4] GPLVM was demonstrated to yield superior performance relative to many of the algorithms in the literature, and therefore for brevity here we only compare to the GPLVM results. We also compared our results with the GPLVM on other test settings. Both the root mean squared error (RMSE) and normalized mean absolute error (NMAE) are used as performance measures. The NMAE is calculated based on the predicted discrete ratings.

In [4] the movie genre was considered. In [5] missingness feature vectors were introduced, by the factorization of the binary matrix indicating which ratings are observed. We consider four kinds of auxiliary information: the movie genre; movie missingness features; user metadata which includes age, gender and occupation; and the user missingness features. These auxiliary features were included using the framework in (7). In the following, the vector (b_1, b_2, b_3, b_4), $b_i \in \{0, 1\}$, denotes whether each of the four forms of auxiliary information is used. For example, if the second and fourth types of auxiliary information are used, this is denoted (0101).

TABLE I

Weak generalization comparison between the GPLVM and our approaches. The low-rank model is discussed in Section II, the user/movie profile models are discussed in Section III (based on rows/columns), and the dual-sparse model is discussed in Section IV. The auxiliary data are handled as in (7).

1M MovieLens

Methods	Settings	NMAE	RMSE
GPLVM	linear	0.4052 ± 0.0011	0.8791 ± 0.0080
	nonlinear	0.4026 ± 0.0020	0.8801 ± 0.0082
User Profile	0000	0.3983 ± 0.0020	0.8681 ± 0.0074
	0011	0.3993 ± 0.0064	0.8660 ± 0.0024
	1100	0.3981 ± 0.0032	0.8667 ± 0.0058
	1111	0.3957 ± 0.0065	0.8628 ± 0.0064
Movie Profile	0000	0.3952 ± 0.0012	0.8621 ± 0.0037
	0011	0.3920 ± 0.0019	0.8614 ± 0.0034
	1100	0.3937 ± 0.0030	0.8611 ± 0.0049
	1111	0.3916 ± 0.0021	0.8598 ± 0.0076
Low Rank	0000	0.3919 ± 0.0049	0.8621 ± 0.0081
	0011	0.3933 ± 0.0037	0.8598 ± 0.0082
	1100	0.3914 ± 0.0056	0.8588 ± 0.0067
	1111	0.3922 ± 0.0052	0.8584 ± 0.0070
Dual Sparse	0000	0.3981 ± 0.0047	0.8642 ± 0.0075
	0011	0.3970 ± 0.0059	0.8632 ± 0.0063
	1100	0.3927 ± 0.0017	0.8612 ± 0.0064
	1111	0.3923 ± 0.0055	0.8631 ± 0.0063

EachMovie

Methods	Settings	NMAE	RMSE
GPLVM	linear	0.4209 ± 0.0017	1.1110 ± 0.0028
	nonlinear	0.4179 ± 0.0018	1.1118 ± 0.0022
User Profile	0000	0.4168 ± 0.0028	1.1138 ± 0.0080
	0101	0.4172 ± 0.0018	1.1102 ± 0.0047
Movie Profile	0000	0.4134 ± 0.0027	1.1028 ± 0.0047
	0101	0.4109 ± 0.0005	1.0936 ± 0.0050

B. Weak and Strong generalization results

Weak generalization tests the model's ability for prediction based on the currently observed ratings. Table I shows the weak generalization results for 1M MovieLens, which are averaged over the same 3 partitions used in [18], [1], [19], [4]. The baselines (0000, without auxiliary information) of our four approaches already give better performance and they all improve with adding the user auxiliary information (0011), the movie auxiliary information (1100), or all the available auxiliary information (1111). All four approaches developed here yield similar performance on this dataset.

The Weak generalization results for EachMovie, which are averaged over 3 random partitions as done in [18], [1], [19], [4], are also shown in Table I. Since neither the user metadata nor movie genre on EachMovie are available to the authors, we report the results of both the user profile and movie profile approaches using the baseline (0000) and both missingness features (0101).

Strong generalization tests the model's ability to adjust for new users, after the initial model has been learned. This is

TABLE II
STRONG GENERALIZATION COMPARISON BETWEEN THE GPLVM AND OUR APPROACHES.

1M MovieLens

Methods	Settings	NMAE	RMSE
GPLVM	linear	0.4071 ± 0.0081	0.8775 ± 0.0239
	nonlinear	0.3994 ± 0.0145	0.8748 ± 0.0268
User	0000	0.4062 ± 0.0134	0.8635 ± 0.0254
Profile	1111	0.4057 ± 0.0138	0.8631 ± 0.0250
Movie	0000	0.3988 ± 0.0099	0.8658 ± 0.0240
Profile	1111	0.3992 ± 0.0127	0.8617 ± 0.0202

EachMovie

Methods	Settings	NMAE	RMSE
GPLVM	linear	0.4171 ± 0.0054	1.0981 ± 0.0077
	nonlinear	0.4134 ± 0.0049	1.1008 ± 0.0080
User	0000	0.4162 ± 0.0010	1.1113 ± 0.0025
Profile	0101	0.4152 ± 0.0014	1.1080 ± 0.0012
Movie	0000	0.4113 ± 0.0014	1.0983 ± 0.0019
Profile	0101	0.4091 ± 0.0005	1.0904 ± 0.0015

TABLE III
RMSE OF OUR APPROACHES ON THE 10M MOVIELENS.

Methods	r_a partition	r_b partition
User Profile	0.8749 ± 0.0009	0.8328 ± 0.0004
Movie Profile	0.8676 ± 0.0006	0.8323 ± 0.0002

addressed by sequential learning in our Bayesian approaches, performed by training the model on the ratings of currently observed users with Gibbs sampling, and when new users are introduced into the model; their feature vectors are sampled conditioned on the movie feature vectors, and then their ratings are naturally put into the full likelihood to influence the whole model through Gibbs sampling.

We choose the same test settings as that in [18], [1], [19], [4] for Strong generalization. The comparison between the GPLVM and our user profile and movie profile approaches for the 1M Movielens and EachMovie are shown in Tables II, respectively.

We also tested our user profile and movie profile approaches on the 10M MovieLens. Table III shows the results for both the r_a and r_b partitions provided with the dataset, in which 10 ratings per user are held out for testing. Averaged over both partitions, the GPLVM reports the RMSE of 0.8740 ± 0.0278 using a 10 dimensional latent space, while the baselines of our approaches achieve average RMSEs of 0.8539 ± 0.0298 and 0.8499 ± 0.0250.

It is well-known that the ensembles of different predictions could lead to improved performance [19], [4]. Simply averaging the user profile and movie profile approaches' results shown in Table I, the 1M MovieLens Weak NMAE and RMSE are reduced to 0.3914 ± 0.0028 and 0.8525 ± 0.0052, respectively. Improvements of this type on other tests are also observed, which are not shown here for brevity.

C. Parameter Settings & Computations

For the baseline movie profile approach with the preset $K = 256$ latent space dimension, one Gibbs iteration in the Weak generalization test takes about 0.25, 8, 25 and 150 seconds on the 100K MovieLens, 1M MovieLens, EachMovie and 10M MovieLens, respectively. All computations were performed on a 2.53GHz E5540 Xeon processor, using non-optimized Matlab software. The first 200 Gibbs iterations are used as burn-in and the following 800 samples are collected for prediction for both the 100K MovieLens and 1M MovieLens. The burn-in & collection iterations for the EachMovie, 10M MovieLens and MLB datasets are 200 & 300, 50 & 100 and 1000 & 500, respectively.

VI. CONCLUSIONS

Three Bayesian approaches are developed for collaborative-filtering problems, along with a principled means of handling auxiliary information. Employing the Indian buffet process, implemented via a Beta-Bernoulli construction, the dimensionality of the underlying latent space is inferred. With the same hyper-parameter settings in all examples, our approaches yields superior performance on widely considered benchmark data sets.

REFERENCES

[1] N. Srebro, J. Rennie, and T. Jaakkola, "Maximum-margin matrix factorization," in *Proc. Neural Information Processing Systems*, 2005.

[2] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization with MCMC," in *Advances in Neural Information Processing Systems*, 2008.

[3] ——, "Probabilistic matrix factorization," in *Advances in Neural Information Processing Systems*, 2008.

[4] N. Lawrence and R. Urtasun, "Non-lineaar matrix factorization with Gaussian processes," in *Proc. Int. Conf. Machine Learning*, 2009.

[5] K. Yu, J. Lafferty, S. Zhu, and Y. Gong, "Large-scale collaborative prediction using a nonparametric random effects model," in *Proc. Int. Conf. Machine Learning*, 2009.

[6] J. Abernethy, F. Bach, T. Evgeniou, and J.-P. Vert, "A new approach to collaborative filtering: operator estimation with spectral regularization," *J. Machine Learning Research*, 2009.

[7] E. J. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Trans. Inform. Theory*, 2010.

[8] T. Griffiths and Z. Ghahramani, "Infinite latent feature models and the indian buffet process," in *NIPS*, 2005, pp. 475–482.

[9] R. Thibaux and M. Jordan, "Hierarchical beta processes and the Indian buffet process," in *Proc. AISTAT*, 2007.

[10] J. Paisley and L. Carin, "Nonparametric factor analysis with beta process priors," in *Proc. Int. Conf. Machine Learning*, 2009.

[11] E. Meeds, Z. Ghahramani, R. Neal, and S. Roweis, "Modeling dyadic data with binary latent factors," in *Neural Information Processing Systems*, 2006.

[12] P. Hoff, "Simulation of the matrix Bingham-von Mises-Fisher distribution, with applications to multivariate and relational data," *J. Comp. Graph. Statistics*, 2009.

[13] Y. Teh and D. Gorur, "Indian buffet process with power-law behavior," in *NIPS*, 2009.

[14] C. Antoniak, "Mixtures of Dirichlet processes with applications to bayesian nonparametric problems," *The Annals of Statistics*, no. 2, pp. 1152–1174, 1974.

[15] D. Dunson and J.-H. Park, "Kernel stick-breaking processes," *Biometrika*, 2009.

[16] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. ICML*, 2009.

[17] M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro, and L. Carin, "Non-parametric bayesian dictionary learning for sparse image representations," in *NIPS*, 2009.

[18] B. Marlin, "Modeling user rating profiles for collaborative filtering," in *NIPS*, 2003.

[19] D. DeCoste, "Collaborative prediction using ensembles of maximum margin matrix factorizations," in *Proc. ICML*, 2006.