Generalized Probabilistic Matrix Factorizations for Collaborative Filtering

Hanhuai Shan
Dept. of Computer Science and Engineering
University of Minnesota, Twin Cities
shan@cs.umn.edu

Arindam Banerjee
Dept. of Computer Science and Engineering
University of Minnesota, Twin Cities
banerjee@cs.umn.edu

***Abstract*—Probabilistic matrix factorization (PMF) methods have shown great promise in collaborative filtering. In this paper, we consider several variants and generalizations of PMF framework inspired by three broad questions: Are the prior distributions used in existing PMF models suitable, or can one get better predictive performance with different priors? Are there suitable extensions to leverage side information? Are there benefits to taking into account row and column biases? We develop new families of PMF models to address these questions along with efficient approximate inference algorithms for learning and prediction. Through extensive experiments on movie recommendation datasets, we illustrate that simpler models directly capturing correlations among latent factors can outperform existing PMF models, side information can benefit prediction accuracy, and accounting for row/column biases leads to improvements in predictive performance.**

***Keywords*-probabilistic matrix factorization, topic models, variational inference**

I. Introduction

In recent years, matrix factorization methods have been successfully applied to collaborative filtering [7]. For example, in movie recommendation, given a rating matrix, the idea is to predict any missing entry (i, j) with the inner product of latent feature vectors for row (user) i and column (movie) j. The idea has been explored by Simon Funk [6], and later a probabilistic framework was developed, yielding probabilistic matrix factorization (PMF) [9] and its Bayesian generalization Bayesian PMF (BPMF) [10]. Both of them have achieved high accuracy in collaborative filtering.

In this paper, we propose generalized PMFs (GPMFs) based on the following three questions: First, *are the prior distributions used in PMF and BPMF suitable, or is it possible to get a better prediction and a simpler algorithm with different priors*? PMF assumes a diagonal covariance for the Gaussian prior, implying independent latent features; BPMF maintains a distribution over all possible covariance matrices. A model between PMF and BPMF is "parametric PMF" (PPMF), which allows a non-diagonal covariance matrix, but does not maintain distributions over all covariance matrices. In this paper, we first consider this PPMF model. The motivation is to avoid the independence assumption in PMF, and avoid the full Bayesian treatment in BPMF to simplify the learning process.

Second, *are there suitable extensions to PMF models to leverage side information for better collaborative filtering*? For example, in movie recommendation, there might be side information on movies, such as genre and cast. It would be interesting if the available side information could help the rating prediction. Therefore, we incorporate side information while performing matrix factorization. The side information we consider is in form of discrete tokens, such as genre and cast. The main idea is to use topic models over the side information and PMF over the ratings matrix. The coupling between two models come from the shared latent variables.

Third, *are there any benefits to take into account row and column biases in the PMF framework?* Certain rows (users) and columns (movies) may have significant biases, e.g., a critical user gives low ratings, and a popular movie gets high ratings. Therefore, the inner product of latent feature vectors might not be a good explanation for the full rating, but only for the residual rating after taking off the biases. While considering biases in SVD [8] improves prediction performance, we propose residual GPMFs which take the biases into the PMF framework.

By running experiments on movie recommendation datasets, we show that (1) PPMF performs better than PMF, BPMF, and co-clustering based algorithms; (2) incorporating side information improves prediction accuracy to a certain extent; and (3) residual models usually generate higher accuracy than the corresponding non-residual ones.

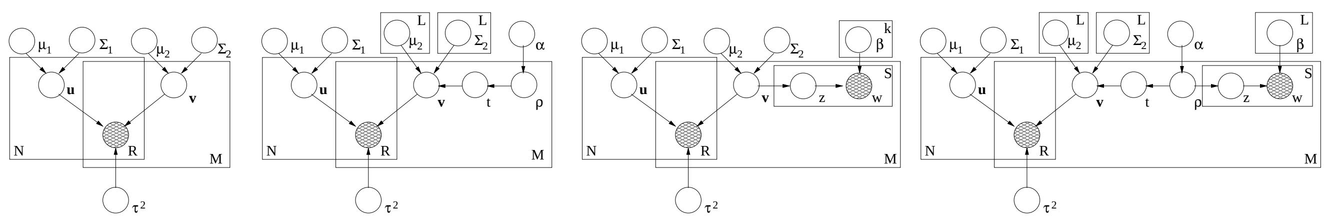
The rest of the paper is organized as follows: In Sections II-IV, we propose GPMFs: Section II is for the models only on the rating matrix, Section III is for the models on both the rating matrix and side information, and Section IV is for the residual models. We present experimental results in Section V and conclude in Section VI.

II. GPMFs on Rating Matrix

In this section, we propose PPMF and MPMF, which are two GPMFs working only on the rating matrix. For the ease of exposition, we assume that we are working on the movie rating matrix, where the rows represent the users and columns represent the movies.

A. Paramatric PMF (PPMF)

Given a matrix R with N users and M movies, assuming the k-dimensional latent feature vector for each user i is $\mathbf{u}_i$ and for each movie j is $\mathbf{v}_j$, the generative process of the matrix R following PPMF is given as follows (Figure 1(a)):



(a) PPMF (b) MPMF (c) CTM-PPMF (d) LDA-MPMF

Figure 1. Graphical models for GPMFs. (a) and (b) work on the rating matrix. (c) and (d) work on the rating matrix with side information.

1) For each user i, $[i]_1^N$ ($[i]_1^N \equiv i = 1 \ldots N$), generate $\mathbf{u}_i \sim N(\boldsymbol{\mu}_1, \Sigma_1)$.
2) For each movie j, $[j]_1^M$, generate $\mathbf{v}_j \sim N(\boldsymbol{\mu}_2, \Sigma_2)$.
3) For each non-missing entry (i,j) in R, generate $R_{ij} \sim N(\mathbf{u}_i^T \mathbf{v}_j, \tau^2)$.

The likelihood of R is given by

$$p(R|\mathcal{P}) = \int_{\mathbf{u}_{1:N}} \int_{\mathbf{v}_{1:M}} \prod_{i=1}^{N} p(\mathbf{u}_i|\boldsymbol{\mu}_1, \Sigma_1) \tag{1}$$

$$\prod_{j=1}^{M} p(\mathbf{v}_j|\boldsymbol{\mu}_2, \Sigma_2) \prod_{i=1}^{N}\prod_{j=1}^{M} p(R_{ij}|\mathbf{u}_i^T\mathbf{v}_j, \tau^2)^{\delta_{ij}} d\mathbf{u}_{1:N} d\mathbf{v}_{1:M} \ ,$$

where $\mathcal{P} = \{\boldsymbol{\mu}_1, \Sigma_1, \boldsymbol{\mu}_2, \Sigma_2, \tau^2\}$ are the model parameters, and δ_{ij} is 1 if R_{ij} is non-missing and 0 otherwise.

Given R, the learning task is to estimate model parameters $\mathcal{P}$ such that $p(R|\mathcal{P})$ is maximized. A general approach is to use the expectation maximization (EM) algorithm, where we calculate the posterior over latent variables $p(\mathbf{u}_{1:N}, \mathbf{v}_{1:M}|R, \mathcal{P})$ in the E-step and estimate model parameters $\mathcal{P}$ in the M-step. In this paper, we propose an efficient variational EM algorithm: First we introduce a tractable family of distributions $q(\mathbf{u}_{1:N}, \mathbf{v}_{1:M}|\mathcal{P}')$ as an approximation of the true posterior $p(\mathbf{u}_{1:N}, \mathbf{v}_{1:M}|R, \mathcal{P})$, where $\mathcal{P}'$ denotes the variational parameters. In particular, $\mathcal{P}' = \{\boldsymbol{\lambda}_{1i}, \boldsymbol{\nu}_{1i}^2, \boldsymbol{\lambda}_{2j}, \boldsymbol{\nu}_{2j}^2, [i]_1^N, [j]_1^M\}$ in PPMF, and the variational distribution q is given by

$$\begin{aligned} & q(\mathbf{u}_{1:N}, \mathbf{v}_{1:M}|\mathcal{P}') \\ =& \prod_{i=1}^{N} q(\mathbf{u}_i|\boldsymbol{\lambda}_{1i}, \mathrm{diag}(\boldsymbol{\nu}_{1i}^2)) \prod_{j=1}^{M} q(\mathbf{v}_j|\boldsymbol{\lambda}_{2j}, \mathrm{diag}(\boldsymbol{\nu}_{2j}^2)) \ , \end{aligned} \tag{2}$$

where for each $\mathbf{u}_i$ and $\mathbf{v}_j$ we have a variational Gaussian of k-dimensions. Given $q(\mathbf{u}_{1:N}, \mathbf{v}_{1:M}|\mathcal{P}')$, applying Jensen's inequality [4] yields a lower bound to the log-likelihood

$$\begin{aligned} \log p(R|\mathcal{P}) \geq & E_q[\log p(\mathbf{u}_{1:N}, \mathbf{v}_{1:M}, R|\mathcal{P})] \\ & + H(q(\mathbf{u}_{1:N}, \mathbf{v}_{1:M}|\mathcal{P}')) \ . \end{aligned} \tag{3}$$

Denoting the lower bound (3) with $L(\mathcal{P}, \mathcal{P}')$, the best lower bound can be found by maximizing $L(\mathcal{P}, \mathcal{P}')$ over $\mathcal{P}'$ in the variational E-step, yielding the best variational parameters $\mathcal{P}'^*$. $L(\mathcal{P}, \mathcal{P}'^*)$ then becomes a surrogate objective function, and optimizing $L(\mathcal{P}, \mathcal{P}'^*)$ over $\mathcal{P}$ in variational M-step gives the estimate of model parameters. The learning process iterates through the variational E-step and M-step to update $\mathcal{P}'$ and $\mathcal{P}$ alternatively until convergence.

We compare PPMF to PMF and BPMF: PMF only uses a zero mean and diagonal covariance Gaussian over $\mathbf{u}$ and $\mathbf{v}$ for regularization, and it uses MAP estimate to find the *best* $\mathbf{u}_{1:N}$ and $\mathbf{v}_{1:M}$ directly. Comparatively, PPMF has a Gaussian prior with an arbitrary mean and a full covariance matrix. Before finding the best $\mathbf{u}_{1:N}$ and $\mathbf{v}_{1:M}$, it first learns model parameters by maximizing $p(R|\mathcal{P})$, which integrates out *all possible* $\mathbf{u}_{1:N}$ and $\mathbf{v}_{1:M}$. For BPMF, it uses a full Bayesian treatment which essentially keeps a distribution over *all possible* PPMF models. Therefore, PPMF lies between PMF and BPMF. Meanwhile, the variational inference used in PPMF is a deterministic approximation algorithm, and the Markov chain Monte Carlo used in BPMF is a stochastic sampling based algorithm.

For prediction, we are using a MAP estimate. In particular, for the estimate of the $(i,j)^{th}$ entry, $\hat{R}_{ij} = \hat{\mathbf{u}}_i^T \hat{\mathbf{v}}_j$, where

$$\{\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_j\} = \underset{(\mathbf{u}_i, \mathbf{v}_j)}{\mathrm{argmax}}\ p(\mathbf{u}_i, \mathbf{v}_j|R, \mathcal{P}) \approx \underset{(\mathbf{u}_i, \mathbf{v}_j)}{\mathrm{argmax}}\ q(\mathbf{u}_i, \mathbf{v}_j|\mathcal{P}') \ .$$

In PPMF, $\underset{(\mathbf{u}_i, \mathbf{v}_j)}{\mathrm{argmax}}\ q(\mathbf{u}_i, \mathbf{v}_j|\mathcal{P}') = \{\boldsymbol{\lambda}_{1i}, \boldsymbol{\lambda}_{2j}\}$, so we have $\hat{R}_{ij} = \boldsymbol{\lambda}_{1i}^T \boldsymbol{\lambda}_{2j}$.

Due to the page limit, the readers are referred to [12] for details and update equations for PPMF and all other models.

B. Mixture PMF (MPMF)

In PPMF, $\mathbf{u}_i$ and $\mathbf{v}_j$ are generated from a single Gaussian distribution. We could generalize the model by allowing $\mathbf{u}_i$ and/or $\mathbf{v}_j$ to be generated from a mixture of Gaussians, yielding mixture PMF.

Assuming one Gaussian $N(\boldsymbol{\mu}_1, \Sigma_1)$ to generate $\mathbf{u}_i$, but a mixture of L Gaussians $\{N(\boldsymbol{\mu}_{2l}, \Sigma_{2l}), [l]_1^L\}$ to generate $\mathbf{v}_j$, the generative process for MPMF is given by (Figure 1(b)):

1) For each user i, $[i]_1^N$, generate $\mathbf{u}_i \sim N(\boldsymbol{\mu}_1, \Sigma_1)$.
2) For each movie j, $[j]_1^M$:
   a) Generate $\boldsymbol{\rho}_j \sim$ Dirichlet$(\boldsymbol{\alpha})$.
   b) Pick a Gaussian $t_j \sim$ discrete$(\boldsymbol{\rho}_j)$.
   c) Generate $\mathbf{v}_j \sim p(\mathbf{v}_j|\boldsymbol{\mu}_{2t_j}, \Sigma_{2t_j})$.
3) For each non-missing entry (i,j) in R, generate $R_{ij} \sim N(\mathbf{u}_i^T \mathbf{v}_j, \tau^2)$.

MPMF assumes that $\mathbf{v}_j$ for movies in a same cluster are generated from a same Gaussian, hence are similar. A few things to note for MPMF: First, for the model in Figure 1(b), $\mathbf{v}_j$ is generated from a mixture of Gaussians, but $\mathbf{u}_i$ is still generated from a single Gaussian. In principle, both

$\mathbf{u}_i$ and $\mathbf{v}_j$ could be generated from a mixture of Gaussians. Second, the Dirichlet-discrete prior ($\boldsymbol{\alpha} \to \boldsymbol{\rho}_j$) over t_j seems unnecessary. Following the standard mixture model, we only need one discrete($\boldsymbol{\rho}$) to generate all $\{t_j, [j]_1^M\}$. However, we are proposing MPMF as an intermediate model from PPMF to LDA-MPMF. The prior $\boldsymbol{\alpha} \to \boldsymbol{\rho}_j$ becomes useful when we combine MPMF with LDA as discussed in Section III-B.

III. GPMFs on Rating Matrix with Side Information

PPMF and MPMF work on the matrix only. In this section, we propose two models to incorporate side information. We use matrix factorization on the rating matrix, meanwhile, we hope that the latent feature vectors $\mathbf{v}$ for two movies are close to each other if the movies are similar on side information, e.g. they have similar plots. In this paper, we assume that we only have side information on movies.

Topic modeling algorithms such as correlated topic models (CTM) [3] and latent Dirichlet allocation (LDA) [4] work on documents, and matrix factorization algorithms work on the rating matrix. Therefore, given data such as rating matrix with movie plots, it is a natural idea to combine matrix factorization with topic models. In particular, we propose CTM-PPMF and LDA-MPMF, with CTM-PPMF a combination of CTM and PPMF, and LDA-MPMF a combination of LDA and MPMF. Other than the plots, in general, the model works on other side information in form of discrete tokens, such as cast, genre, etc., but for ease of exposition, we will still use "document", "words", and "topics" when describing the model.

A. *CTM-PPMF*

The main idea in CTM-PPMF is as follows: For each movie j, $\mathbf{v}_j$ not only serves as PPMF's latent feature vector for its ratings, but also serves as CTM's membership vector over topics (after logistic transformation) for its side information. Therefore the common $\mathbf{v}_j$ for both the ratings and side information of movie j becomes the glue to combine PPMF and CTM together.

Given a matrix R with N users and M movies, for each movie j, we have side information $\mathbf{w}_j$ as a collection of words. Denoting the side information of M movies as $W = \{\mathbf{w}_j, [j]_1^M\}$, the generative process of (R, W) for CTM-PPMF is given as follows (Figure 1(c)):

1) For each user i in R, $[i]_1^N$, generate $\mathbf{u}_i \sim N(\boldsymbol{\mu}_1, \Sigma_1)$.
2) For each movie j in R, $[j]_1^M$, generate $\mathbf{v}_j \sim N(\boldsymbol{\mu}_2, \Sigma_2)$.
3) For the h^{th} word in $\mathbf{w}_j$,$[j]_1^M$:
   a) Generate a topic $z_{jh} \sim$ discrete(logistic($\mathbf{v}_j$)).
   b) Generate the word $w_{jh} \sim p(w_{jh}|\boldsymbol{\beta}_{z_{jh}})$.
4) For each non-missing entry (i, j) in R, generate $R_{ij} \sim N(\mathbf{u}_i^T \mathbf{v}_j, \tau^2)$.

Here logistic$(\mathbf{v}_j) = \frac{\exp(\mathbf{v}_j)}{\sum_{c=1}^{k} \exp(v_{jc})}$ is a logistic function to transform $\mathbf{v}_j$ to a discrete distribution, and $\boldsymbol{\beta} = \{\boldsymbol{\beta}_c, [c]_1^k\}$ are k discrete distributions for k topics over all words in the dictionary.

B. *LDA-MPMF*

The main idea in LDA-MPMF is as follows: For MPMF on the rating matrix, each movie has a Dirichlet-discrete prior ($\boldsymbol{\alpha} \to \boldsymbol{\rho}_j$). Meanwhile, if we use LDA on side information, each movie also needs a Dirichlet-discrete prior to generate the topics for the words in side information. Therefore, letting MPMF and LDA share the Dirichlet-discrete prior, we can combine MPMF and LDA together.

Given a rating matrix R with N users and M movies, where each movie j has the side information $\mathbf{w}_j$, the generative process of (R, W) for LDA-MPMF is (Figure 1(d)):

1) For each user i in R, $[i]_1^N$, generate $\mathbf{u}_i \sim N(\boldsymbol{\mu}_1, \Sigma_1)$.
2) For each movie j in R, $[j]_1^M$:
   a) Generate $\boldsymbol{\rho}_j \sim$ Dirichlet($\boldsymbol{\alpha}$).
   b) Pick a Gaussian $t_j \sim$ discrete($\boldsymbol{\rho}_j$).
   c) Generate $\mathbf{v}_j \sim p(\mathbf{v}_j|\boldsymbol{\mu}_{2t_j}, \Sigma_{2t_j})$.
3) For the h^{th} word in $\mathbf{w}_j$,$[j]_1^M$:
   a) Generate a topic $z_{jh} \sim$ discrete($\boldsymbol{\rho}_j$).
   b) Generate the word $w_{jh} \sim p(w_{jh}|\boldsymbol{\beta}_{z_{jh}})$.
4) For each non-missing entry (i, j) in R, generate $R_{ij} \sim N(\mathbf{u}_i^T \mathbf{v}_j, \tau^2)$.

Here $\boldsymbol{\beta} = \{\boldsymbol{\beta}_l, [l]_1^L\}$ are L discrete distributions for L topics over all words in the dictionary.

In Figure 1, LDA-MPMF and CTM-PPMF show different ways to incorporate side information. In CTM-PPMF, the topics $\mathbf{z}$ for side information are *generated from* the membership vector logistic($\mathbf{v}$). Therefore, similar logistic($\mathbf{v}$) may lead to similar documents apriori, i.e., similar documents indicate similar $\mathbf{v}$ hence similar ratings for movies aposteriori. In LDA-MPMF, the ratings and the side information for a particular movie *share* the membership vectror $\boldsymbol{\rho}$. Therefore, conditioned on similar side information of movies, their $\boldsymbol{\rho}$ would be similar, so the model would probably pick a same Gaussian in the mixture to generate their $\mathbf{v}$, leading to similar $\mathbf{v}$ and further the similar ratings for these movies. Due to their different strategies, in CTM-PPMF, the dimension of $\mathbf{v}$ is the same with the number of topics in the documents, but in LDA-MPMF, the dimension of $\mathbf{v}$ has nothing to do with the number of topics.

IV. Residual GPMFs

There are usually biases in the ratings. For example, a popular movie receives high ratings, and a critical user gives low ratings. Therefore, it may be unwise to explain the full rating R_{ij} using the inner product of $\mathbf{u}_i$ and $\mathbf{v}_j$. Instead, a certain part of R_{ij} could be explained by the user and movie biases, hence we use $\mathbf{u}_i^T \mathbf{v}_j$ to explain R_{ij} with the biases taken off, i.e., the residue of R_{ij}, yielding residual GPMFs.

We have a corresponding residual model for each of the four models we have proposed. In particular, instead of

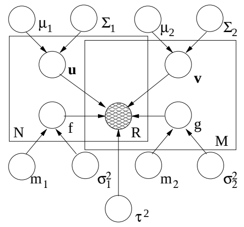


Figure 2. Graphical model for residual PPMF.

generating R_{ij} from $N(\mathbf{u}_i^T\mathbf{v}_j, \tau^2)$, in the residual models, R_{ij} is generated from $N(\mathbf{u}_i^T\mathbf{v}_j + f_i + g_j, \tau^2)$, where f_i and g_j are the row and column biases, and are assumed to be generated from $N(m_1, \sigma_1^2)$ and $N(m_2, \sigma_2^2)$ respectively. Therefore, the matrix factorization is performed on R after the effects of f_i and g_j removed. For brevity, we only discuss residual PPMF (rsPPMF) as an example, the other residual models can be obtained in a similar way. We report experimental results on all residual models in Section V.

As in Figure 2, the generative process of rsPPMF for the matrix R with N users and M movies is as follows:

1) For each user i, $[i]_1^N$, generate $\mathbf{u}_i \sim N(\boldsymbol{\mu}_1, \Sigma_1)$.
2) For each movie j, $[j]_1^M$, generate $\mathbf{v}_j \sim N(\boldsymbol{\mu}_2, \Sigma_2)$.
3) For each user i, $[i]_1^N$, generate $f_i \sim N(m_1, \sigma_1^2)$.
4) For each movie j, $[j]_1^M$, generate $g_j \sim N(m_2, \sigma_2^2)$.
5) For each non-missing entry (i, j) in R, generate $R_{ij} \sim N(\mathbf{u}_i^T\mathbf{v}_j + f_i + g_j, \tau^2)$.

The likelihood of observing R is therefore

$$p(R|\mathcal{P}) = \int_{\mathbf{u}_{1:N}}\int_{\mathbf{v}_{1:M}}\int_{f_{1:N}}\int_{g_{1:M}} \Big(\prod_{i=1}^{N} p(\mathbf{u}_i|\boldsymbol{\mu}_1, \Sigma_1) p(f_i|m_1, \sigma_1^2)\Big)$$
$$\Big(\prod_{j=1}^{M} p(\mathbf{v}_j|\boldsymbol{\mu}_2,\Sigma_2) p(g_j|m_2,\sigma_2^2)\Big) \prod_{i=1}^{N}\prod_{j=1}^{M} p(R_{ij}|\mathbf{u}_i^T\mathbf{v}_j + f_i + g_j, \tau^2)^{\delta_{ij}}$$
$$d\mathbf{u}_{1:N} d\mathbf{v}_{1:M} df_{1:N} dg_{1:M} \tag{4}$$

where $\mathcal{P} = \{\boldsymbol{\mu}_1, \Sigma_1, \boldsymbol{\mu}_2, \Sigma_2, \tau^2, m_1, \sigma_1^2, m_2, \sigma_2^2\}$.

For inference, we introduce two new terms to the variational distribution: a variational Gaussian $N(\theta_{1i}, \eta_{1i}^2)$ for f_i and $N(\theta_{2j}, \eta_{2j}^2)$ for g_j. The variational distribution becomes:

$$q(\mathbf{u}_{1:N}, \mathbf{v}_{1:M}|\mathcal{P}') = \Big(\prod_{i=1}^{N} q(\mathbf{u}_i|\boldsymbol{\lambda}_{1i}, \text{diag}(\boldsymbol{\nu}_{1i}^2)) q(f_i|\theta_{1i}, \eta_{1i}^2)\Big)$$
$$\Big(\prod_{j=1}^{M} q(\mathbf{v}_j|\boldsymbol{\lambda}_{2j}, \text{diag}(\boldsymbol{\nu}_{2j}^2)) q(g_j|\theta_{2j}, \eta_{2j}^2)\Big) \ , \tag{5}$$

where $\mathcal{P}' = \{\boldsymbol{\lambda}_{1i}, \boldsymbol{\nu}_{1i}^2, \boldsymbol{\lambda}_{2j}, \boldsymbol{\nu}_{2j}^2, \theta_{1i}, \eta_{1i}^2, \theta_{2j}, \eta_{2j}^2, [i]_1^N, [j]_1^M\}$.

For the other three models, their corresponding residual models also generate R_{ij} from $N(\mathbf{u}_i^T\mathbf{v}_j + f_i + g_j, \tau^2)$, and they also need a variational Gaussian $N(\theta_{1i}, \eta_{1i}^2)$ for f_i and $N(\theta_{2j}, \eta_{2j}^2)$ for g_j while doing inference.

For prediction of residual GPMFs following MAP estimate, we have $\hat{R}_{ij}^{rs} = \hat{R}_{ij} + \hat{f}_i + \hat{g}_j$, where $\hat{R}_{ij} = \hat{\mathbf{u}}_i^T\hat{\mathbf{v}}_j$ according to the corresponding original models, $\hat{f}_i = \theta_{1i}$ and $\hat{g}_j = \theta_{2j}$.

Table I
MSE FROM PPMF AND CO-CLUSTERING BASED ALGORITHMS.

k	5	10	15	20	25	30
specCoc s2	0.1212 ±0.0012	0.1170 ±0.0010	0.1141 ±0.0001	0.1136 ±0.0001	0.1128 ±0.0001	0.1115 ±0.0001
specCoc s5	0.0948 ±0.0022	0.0987 ±0.0063	0.0953 ±0.0014	0.0987 ±0.0035	0.0968 ±0.0035	0.0977 ±0.0037
BregCoc s2	0.0902 ±0.0001	0.0885 ±0.0001	0.0881 ±0.0001	0.0878 ±0.0001	0.0874 ±0.0004	0.0875 ±0.0004
BregCoc s5	0.0978 ±0.0020	0.0987 ±0.0030	0.0967 ±0.0031	0.1007 ±0.0099	0.0953 ±0.0022	0.0955 ±0.0022
BCC	0.1079 ±0.0014	0.1060 ±0.0001	0.1044 ±0.0001	0.1037 ±0.0001	0.1036 ±0.0006	0.1035 ±0.0005
PPMF	**0.0800 ±0.0005**	**0.0784 ±0.0005**	**0.0785 ±0.0005**	**0.0784 ±0.0006**	**0.0785 ±0.0005**	**0.0783 ±0.0006**

Table II
MSE FROM PMF, BPMF AND PPMF.

k	5	10	15	20	25	30
PMF	**0.07996 ±0.00044**	0.08040 ±0.00036	0.08238 ±0.00070	0.08343 ±0.00067	0.08362 ±0.00047	0.08431 ±0.00059
BPMF	0.08517 ±0.00055	0.08685 ±0.00058	0.08970 ±0.00062	0.09382 ±0.00063	0.09879 ±0.00067	0.10525 ±0.00076
PPMF	0.08003 ±0.00057	**0.07837 ±0.00051**	**0.07849 ±0.00050**	**0.07840 ±0.00055**	**0.07855 ±0.00054**	**0.07832 ±0.00059**

V. Experimental Results

In this section, we run collaborative filtering on movie rating data using multiple algorithms, including PMF, BPMF, co-clustering algorithms and GPMFs we have proposed[1].

We use two movielens datasets[2]. One contains 100,000 ratings for 1682 movies by 943 users, the other contains 1 million ratings for 3900 movies by 6040 users. Due to the page limit, we only discuss results on the larger dataset hereafter. Please refer to [12] for results on the smaller one. For each movie we extract three types of side information from IMDB[3]—cast, genre, and plot. For genre, there are 25 movie types. For cast, we only use the top-10 ranked most important actors/actresses in each movie, and there are totally 13924 actors/actresses. For plot, we use the plots written by IMDB users, and there are 2693 words in the dictionary after preprocessing. We then remove the movies with one or more types of side information missing.

We use mean square error (MSE) as the measurement of prediction accuracy on the rating matrix. A small part of ratings is held out as the validation set to determine the stopping time of the learning process. In particular, we stop the variational EM when the number of iterations is larger than 3 (because we do not want the iteration to stop too early) and the MSE on the validation set is larger than the last iteration. Unless otherwise specified, we use this "early stopping" strategy for all matrix factorization based algorithms. For the rest of the ratings, we use a 10-fold cross validation: We divide all ratings evenly into 10 parts, each of which is picked as the test set in turn, and the rest are used as the training set. We then take the average MSE over 10 folds on the test set. Before running the algorithm, we

[1]For algorithms we compare with, the code for spectral co-clustering is from http://adios.tau.ac.il/SpectralCoClustering/, and the code for other algorithms are from the authors of original papers.

[2]www.movielens.umn.edu

[3]www.imdb.com

Table III
MSE FROM ORIGINAL AND RESIDUAL GPMFS WITH $k = 30$.

	original	residual
PPMF	0.07832±0.00059	**0.07786±0.00056**
CTM-PPMF (cast)	0.07800±0.00056	**0.07734±0.00055**
CTM-PPMF (plot)	0.07942±0.00055	**0.07859±0.00058**
CTM-PPMF (genre)	0.07817±0.00047	**0.07727±0.00054**
MPMF	0.07906±0.00054	**0.07819±0.00052**
LDA-MPMF (cast)	0.07897±0.00055	**0.07839±0.00053**
LDA-MPMF (plot)	0.07905±0.00052	**0.07824±0.00053**
LDA-MPMF (genre)	0.07923±0.00045	**0.07824±0.00053**

transform each rating R_{ij} to $\sqrt{6 - R_{ij}}$, such that the ratings are closer to a Gaussian distribution [1].

A. *PPMF vs. Co-clustering Algorithms*

We first compare PPMF with three co-clustering algorithms: spectral co-clustering (specCoc) [5], Bregman co-clustering (BregCoc) [2], and Bayesian co-clustering (BCC) [11]. We use two schemes (s2 and s5) for specCoc and BregCoc, with different schemes keeping different types of statistics [2]. We run k-means on the low-rank vectors from imputed SVD to initialize the membership vectors in co-clustering algorithms. We use random initialization for PPMF. The MSE with k from 5 to 30 are presented in Table I, where k is the dimension of $\mathbf{u}$ and $\mathbf{v}$ for PPMF and the number of row/column clusters for the co-clustering algorithms. We can see that PPMF clearly generates a smaller MSE compared to the co-clustering algorithms. While co-clustering algorithms can learn the clustering structures of a matrix, their matrix approximation results are usually not as good as the matrix factorization based algorithms.

B. *PPMF vs. PMF and BPMF*

We then compare PPMF with PMF and BPMF. For BPMF, we do not use the early stopping strategy since it hurts the performance. The MSE of these three algorithms using random initialization are presented in Table II. On both datasets, PPMF performs better than PMF. We are surprised to see that PPMF performs even better than BPMF, which gives us some supportive evidence to go with PPMF instead of a full Bayesian model as in BPMF, but more rigorous experiments will be needed to fully compare the performance of these algorithms.

C. *GPMFs vs. residual GPMFs*

To compare GPMFs with residual GPMFs, we present the result with $k = 30$ in Table III. We can see that the residual models always have high accuracy than original models. Such observation could also be obtained from Table IV when we provide all results for both GPMFs and residual GPMFs with k from 5 to 30.

D. *GPMFs with side information*

We use three types of side information—cast, plot and genre. To see whether incorporating side information helps improve the accuracy, we show the results for GPMFs with k from 5 to 30 in Table IV, with Table IV(a) for original models and Table IV(b) for residual models respectively (We put "rs" before the model name to denote residual models.). In each table, the top part is the results of the pair of PPMF and CTM-PPMF, and the bottom part is the results of the pair of MPMF and LDA-MPMF. In each part, we put a $\bullet$ under the results of CTM-PPMF (LDA-MPMF) if the MSE is lower than corresponding PPMF (MPMF) which does not use side information. Also, for each choice of k, we use bold for the best result. For MPMF and LDA-MPMF, we have tried different values for L but it does not affect the result much, so we only use $L = 5$.

For LDA-MPMF compared to MPMF, incorporating side information seems to help to a certain extent, especially for residual models. For CTM-PPMF compared to PPMF, the advantage of taking side information is more clear for both the original and residual models, especially when incorporating cast and genre. Overall, PPMF and CTM-PPMF perform better than MPMF and LDA-PPMF.

Among three types of side information, genre seems to be the most informative, then comes cast, and plots are more hurting the result than helping, especially for CTM-PPMF. For the reasons of bad performance using plots, the plots are quite subjective and highly compressed, and two movies with similar plots may be completely different in their quality. As for the cast, it may help prediction if it contains famous movie stars, but for most actors/actresses, whether he/she shows up in a movie does not seem to affect the rating that much. Also, most actors/actresses only appear in one or two movies, making it difficult to discover the relationship between a certain actor/actress and the movie ratings. In comparison, there are only 25 movie types in genre, so a large number of movies would be assigned to a same movie type, making it easier to find out the relationship between the rating and genre, which could be one reason of genre's usefulness in prediction. However, intuitively, genre is not that informative. A movie will not necessarily get a high or low rating just because it belongs to a certain type. Due to all reasons above, although we have expected better performance, the side information we have used may not be powerful enough to generate a distinct improvement on accuracy, at least through the ways we have considered.

Although LDA-MPMF does not show advantage in prediction accuracy, it generates several interesting cast groups after running on rating+cast. Table V shows two examples of top 10 actor/actress names in two clusters. Table V(a) is a list of actors/actresses acting in Star Trek or other science fiction movies, and we also give the movie names they performed in. Table V(b) is another cluster of actors/actresses. Given the year of their movies, we can see that they are mostly active as early as in 40's-60's, which is a distinct cluster since most movies in movielens are in 80's or 90's.

VI. CONCLUSION

In this paper, we have generalized probabilistic matrix factorizations along several directions: We have proposed

Table IV
MSE FOR GENERALIZED PMFS.

(a) GPMFs

k	5	10	15	20	25	30
PPMF	0.08003 ±0.00057	0.07837 ±0.00051	0.07849 ±0.00050	0.07840 ±0.00055	0.07855 ±0.00054	0.07832 ±0.00059
CTM -PPMF cast	0.08108 ±0.00053	0.07875 ±0.00047	**0.07829 ±0.00042** ●	0.07810 ±0.00053 ●	0.78090 ±0.00053 ●	**0.07800 ±0.00056** ●
CTM -PPMF plot	0.08166 ±0.00048	0.07991 ±0.00046	0.07945 ±0.00049	0.07915 ±0.00040	0.07923 ±0.00052	0.07942 ±0.00055
CTM -PPMF genre	0.08062 ±0.00040	**0.07831 ±0.00045** ●	0.07830 ±0.00057 ●	**0.07800 ±0.00052** ●	**0.07808 ±0.00048** ●	0.07817 ±0.00047 ●
MPMF	**0.07983 ±0.00061**	0.07874 ±0.00071	0.07886 ±0.00067	0.07874 ±0.00060	0.07915 ±0.00043	0.07906 ±0.00054
LDA -MPMF cast	0.07986 ±0.00066	0.07861 ±0.00053 ●	0.07924 ±0.00048	0.07877 ±0.00050	0.07908 ±0.00061 ●	0.07897 ±0.00055 ●
LDA -MPMF plot	0.08000 ±0.00059	0.07869 ±0.00060 ●	0.07916 ±0.00057	0.07884 ±0.00039	0.07904 ±0.00047 ●	0.07905 ±0.00052 ●
LDA -MPMF genre	0.08013 ±0.00049	0.07902 ±0.00053	0.07908 ±0.00053	0.07938 ±0.00044	0.07972 ±0.00049	0.07923 ±0.00045

(b) Residual GPMFs

k	5	10	15	20	25	30
rsPPMF	**0.07902 ±0.00061**	**0.07795 ±0.00074**	0.07776 ±0.00056	0.07780 ±0.00061	0.07783 ±0.00053	0.07786 ±0.00056
rsCTM -PPMF cast	0.07916 ±0.00078	0.07818 ±0.00084	0.07799 ±0.00044	0.07754 ±0.00051 ●	0.07727 ±0.00058 ●	0.07734 ±0.00055 ●
rsCTM -PPMF plot	0.07957 ±0.00072	0.07851 ±0.00064	0.07860 ±0.00101	0.07838 ±0.00047	0.07843 ±0.00052	0.07859 ±0.00058
rsCTM -PPMF genre	0.07903 ±0.00075	0.07814 ±0.00054	**0.07768 ±0.00054** ●	**0.07750 ±0.00053** ●	**0.07722 ±0.00059** ●	**0.07727 ±0.00054** ●
rsMPMF	0.07982 ±0.00061	0.07874 ±0.00070	0.07886 ±0.00066	0.07875 ±0.00060	0.07843 ±0.00057	0.07819 ±0.00052
rsLDA -MPMF cast	0.07954 ±0.00060 ●	0.07871 ±0.00061 ●	0.07840 ±0.00049 ●	0.07856 ±0.00054 ●	0.07859 ±0.00060	0.07839 ±0.00053
rsLDA -MPMF plot	0.07962 ±0.00053 ●	0.07875 ±0.00061	0.07840 ±0.00049 ●	0.07850 ±0.00054 ●	0.07856 ±0.00060	0.07824 ±0.00053
rsLDA -MPMF genre	0.07962 ±0.00057 ●	0.07848 ±0.00042 ●	0.07856 ±0.00061 ●	0.07835 ±0.00058 ●	0.07837 ±0.00054 ●	0.07824 ±0.00053

Table V
TWO CAST CLUSTERS FROM LDA-MPMF.

(a)

cast	movie names
Carrey, Jim	Batman Forever; Ace Ventura: Pet Detective; The Mask; The Cable Guy; Liar Liar; The Truman Show; Ace Ventura: When Nature calls; Dumb & Dumber
Doohan, Jams	Star Trek series
Kelley, DeForest	Star Trek series
Koenig, Walter	Star Trek series
Nimony, Lenard	Star Trek series
Shatner, William	Star Trek series
Takei, George	Star Trek series
Nichols, Nichelle	Star Trek series
Harris, Ed	Apollo 13; The Firm; The Rock; The Abyss; Milk Money; Just Cause Glengarry Glen Ross; The Right Stuff; Nixon; Eye for an Eye
Gough, Michael	Batman series

(b)

cast	year of movies
Grant, Cary	1940, 1959, 1946, 1955, 1940, 1938, 1944, 1963, 1941
Stewart, James	1939, 1940, 1958, 1946, 1954
Bogart, Humphrey	1942, 1941, 1954, 1951, 1948, 1946
Balsam, Martin	1961, 1957, 1960, 1991, 1962
Hepburn, Audrey	1961, 1964, 1954, 1953, 1963, 1957, 1957
Mitchell, Thomas	1939, 1939, 1946, 1937, 1952, 1943
Rains, Claude	1939, 1942, 1946, 1938, 1962, 1939, 1946
Kelly, Grace	1955, 1954, 1954, 1952
Coburn Jams	1994, 1996, 1963, 1996, 1997, 1990
Newman, Paul	1994, 1973, 1969, 1958, 1967, 1998, 1994

PPMF which is more flexible than PMF but simpler than BPMF. We have incorporated the side information to help matrix factorization. We have also proposed residual models which are able to account for the row and column biases. We show the following results in the experiments: PPMF generates higher accuracy than PMF and BPMF, as well as the co-clustering based algorithms. Also, the residual models usually have better performance than the corresponding original models. Moreover, incorporating side information does help prediction to a certain extent. The future work includes generalizing PMF to work on a series of matrices with different time stamps, and incorporating the side information for multiple entities such as for both users and movies.

Acknowledgements: This research was supported by NSF grants IIS-0812183, IIS-0916750, CNS-1017647, IIS-1029711, NSF CAREER grant IIS-0953274, and NASA grant NNX08AC36A.

REFERENCES

[1] D. Aggarwal and S. Merugu. Predictive discrete latent factor models for large scale dyadic data. In *KDD*, 2007.

[2] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. Modha. A generalized maximum entropy approach to Bregman co-clustering and matrix approximation. *JMLR*, 2007.

[3] D. Blei and J. Lafferty. Correlated topic models. In *NIPS*, 2005.

[4] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[5] I. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD*, 2001.

[6] S. Funk. http://sifter.org/∼simon/journal/20061211.html.

[7] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 2009.

[8] A. Paterek. Improving regularized singular value decomposition for collaborative filtering. In *KDD Cup and Work Shop*, 2007.

[9] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS*, 2007.

[10] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *ICML*, 2008.

[11] H. Shan and A. Banerjee. Bayesian co-clustering. In *ICDM*, 2008.

[12] H. Shan and A. Banerjee. Generalized probabilistic matrix factorizations for collaborative filtering. Technical Report 10-024, University of Minnesota, Twin Cities, 2010.