Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 10-024

Generalized Probabilistic Matrix Factorizations for Collaborative Filtering

Hanhuai Shan and Arindam Banerjee

September 30, 2010

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 [10]. 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 [8], and later a probabilistic framework was developed, yielding probabilistic matrix factorization (PMF) [16] and its Bayesian generalization Bayesian PMF (BPMF) [17]. 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 [14] 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.

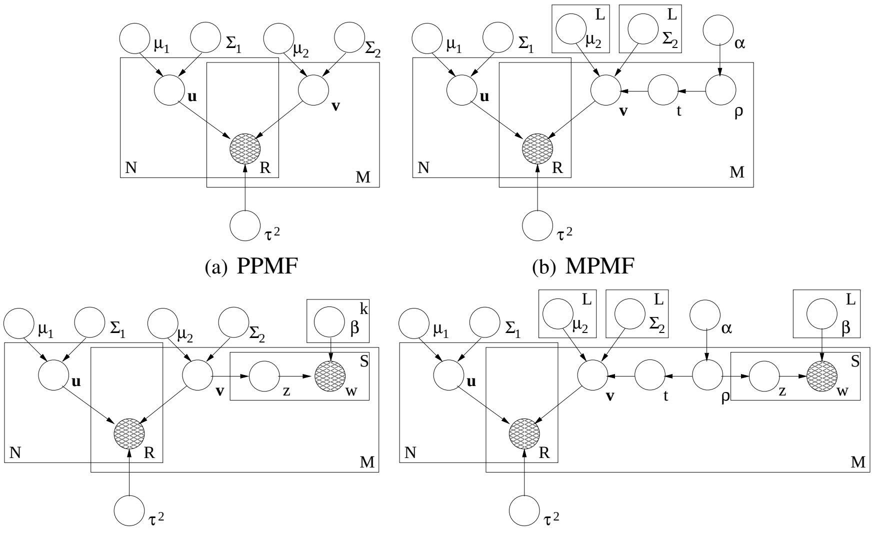
The rest of the paper is organized as follows: In Sections III-V, we propose GPMFs: Section III is for the models only on the rating matrix, Section IV is for the models on both the rating matrix and side information, and Section V is for the residual models. We present experimental results in Section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

II. Preliminaries and Related Work

In this section, we give a brief overview of PMF and BPMF, as well as two topic models LDA and CTM. We also give the recent related work on extensions to PMF models.

A. PMF and BPMF

PMF [16] is a probabilistic linear matrix factorization model. Given a matrix R with N rows and M columns, assuming each row i and column j has a latent feature vector $\mathbf{u}_i$ and $\mathbf{v}_j$, R_{ij} is then generated from a Gaussian



(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.

$N(\mathbf{u}_i^T\mathbf{v}_j, \tau^2)$. The priors for $\mathbf{u}_i$ and $\mathbf{v}_j$ are multivariate Gaussians $N(\mathbf{u}_i|\mathbf{0}, \sigma_1^2 I)$ and $N(\mathbf{v}_j|\mathbf{0}, \sigma_2^2 I)$. The model is learned by maximizing the posterior on latent feature vectors $\mathbf{u}_{1:N}$ and $\mathbf{v}_{1:M}$.

BPMF [17] introduces a full Bayesian prior to PMF. In particular, for $\mathbf{u}_{1:N}$, it has a Gaussian prior $N(\mathbf{u}_i|\boldsymbol{\mu}_1, \Sigma_1)$, and for parameters $(\boldsymbol{\mu}_1, \Sigma_1)$, there is a prior given by

$$p(\boldsymbol{\mu}_1, \Sigma_1|\boldsymbol{\mu}_0, \nu_0, W_0) = p(\boldsymbol{\mu}_1|\boldsymbol{\mu}_0, \Sigma_1)p(\Sigma_1|W_0, \nu_0) \ ,$$

where $p(\boldsymbol{\mu}_1|\boldsymbol{\mu}_0, \Sigma_1)$ is a multivariate Gaussian with mean $\boldsymbol{\mu}_0$ and covariance Σ_1, and $p(\Sigma_1|W_0, \nu_0)$ is a Wishart distribution with ν_0 and W_0 being degrees of freedom and scale matrix respectively. $\mathbf{v}_{1:M}$ have similar priors.

B. Topic models

Latent Dirichlet allocation (LDA) [5] is one of the most popular topic modeling algorithms. LDA assumes that each document has a separate discrete distribution $\boldsymbol{\pi}$ to generate the topics of words, and all documents share a Dirichlet prior $\boldsymbol{\alpha}$. The generative process for each document $\mathbf{w}$ is:

1) Draw $\boldsymbol{\pi} \sim$ Dirichlet($\boldsymbol{\alpha}$).
2) For each word w_h in $\mathbf{w}$:
   a) Draw a topic $z_h = c \sim$ discrete(π).
   b) Draw w_h from p($w_h|\boldsymbol{\beta}, z_h$).

$\boldsymbol{\beta} = \{\boldsymbol{\beta}_c, [c]_1^k\}$ ($[c]_1^k \equiv c = 1 \ldots k$) are the parameters for k discrete topic distributions over all words, and each time one of them is picked to generate the word given its topic.

Correlated topic models [4] (CTM) is similar to LDA, except that instead of the Dirichlet prior, it uses a Gaussian prior to capture the correlation among the topics.

C. Related work

Collaborative filtering with low-dimensional factors models have been widely explored: [14] improves regularized SVD. [9], [13] are probabilistic models with hidden factor variables connected to the ratings. In addition, as extensions to PMF: [11] proposes a non-linear matrix factorization algorithm using Gaussian processes. Bayesian clustered tensor factorization [20] and mixed membership matrix factorization [12] combine the clustering based models with the factorization based models. [6] and [19] generalize the idea of PMF to work on multi-way/multi-relational data. Moreover, [15] and [1] also use side information to help matrix factorization, but they differ from GPMFs either in the types of side information to work on, the basic mechanism of combining side information, or the inference algorithm.

III. GPMFs on Rating Matrix

In this section, we propose PPMF and MPMF, which are two GPMFs working only on the rating matrix. For 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. Parameterized PMF

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)):

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 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

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

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

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

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

where for each $\mathbf{u}_i$ and $\mathbf{v}_j$ we have a variational Gaussian distribution of k-dimensions. Given $q(\mathbf{u}_{1:N}, \mathbf{v}_{1:M}|\mathcal{P}')$, applying Jensen's inequality [5] yields a lower bound to the log-likelihood

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

The variational EM algorithm then iterates through E-step and M-step as follows:

E-step: 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 $\mathcal{P}'$, which gives

$$\begin{aligned} \boldsymbol{\lambda}_{1i} =& \Big(\Sigma_1^{-1} + \frac{1}{\tau^2}\sum_{j=1}^{M}\delta_{ij}(\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T + \mathrm{diag}(\boldsymbol{\nu}_{2j}^2))\Big)^{-1} \\ & \Big(\Sigma_1^{-1}\boldsymbol{\mu}_1 + \frac{1}{\tau^2}\sum_{j=1}^{M}\delta_{ij}R_{ij}\boldsymbol{\lambda}_{2j}\Big) \qquad (4) \end{aligned}$$

$$\nu_{1ic}^2 = \Big(\sum_{j=1}^{M}\delta_{ij}(\lambda_{2jc}^2 + \nu_{2jc}^2)/\tau^2 + \Sigma_{1,cc}^{-1}\Big)^{-1} \ , \qquad (5)$$

where $\Sigma_{1,cc}^{-1}$ is the cth element of Σ_1^{-1}'s diagonal. $\boldsymbol{\lambda}_{2j}$ and ν_{2jc}^2 have a similar form. Note that although the covariance matrices for variational Gaussians are diagonal, the model parameters Σ_1 and Σ_2 are not diagonal, so PPMF is able to capture the correlation among latent factors.

M-step: $\mathcal{P}'^*$ from the E-step gives us a surrogate objective function $L(\mathcal{P}, \mathcal{P}'^*)$, optimizing $L(\mathcal{P}, \mathcal{P}'^*)$ over $\mathcal{P}$ yields the estimate of the model parameters:

$$\boldsymbol{\mu}_1 = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\lambda}_{1i} \qquad (6)$$

$$\Sigma_1 = \frac{1}{N}\sum_{i=1}^{N}\big(\mathrm{diag}(\boldsymbol{\nu}_{1i}^2) + (\boldsymbol{\lambda}_1 - \boldsymbol{\mu}_1)(\boldsymbol{\lambda}_1 - \boldsymbol{\mu}_1)^T\big) \qquad (7)$$

$$\begin{aligned} \tau^2 =& \frac{1}{A}\sum_{i=1}^{N}\sum_{j=1}^{M}\delta_{ij}\Big(R_{ij}^2 + \boldsymbol{\lambda}_{1i}^T\mathrm{diag}(\boldsymbol{\nu}_{2j}^2)\boldsymbol{\lambda}_{1i} + \boldsymbol{\lambda}_{2j}^T\mathrm{diag}(\boldsymbol{\nu}_{1i}^2)\boldsymbol{\lambda}_{2j} \\ & -2R_{ij}\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{2j} + (\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{2j})^2 + \mathrm{Tr}(\mathrm{diag}(\boldsymbol{\nu}_{1i}^2)\mathrm{diag}(\boldsymbol{\nu}_{2j}^2))\Big) \qquad (8) \end{aligned}$$

where A is the total number of non-missing entries in R. The expressions for $\boldsymbol{\mu}_2$ and Σ_2 are similar with $\boldsymbol{\mu}_1$ and Σ_1.

To learn the model, the algorithm iterates through the E-step and M-step until convergence. In the E-step, the algorithm updates $\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$, and $\boldsymbol{\nu}_1$, $\boldsymbol{\nu}_2$ alternatively till convergence. The time complexity of each E-step is $O(k^2(kM + kN + MN)t_E)$, where k is the dimension of $\mathbf{u}$ and $\mathbf{v}$, and t_E is the number of iterations inside the E-step. The time complexity of each M-step is $O(k^2N + k^2M + kMN)$.

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 the log-likelihood of R, which integrates out *all possible* $\mathbf{u}_{1:N}$ and $\mathbf{v}_{1:M}$. For BPMF, it uses a full Bayesian treatment with hyperparameters on top of the Gaussian priors, which essentially keeps a distribution over *all possible* PPMF models. Therefore, PPMF lies between PMF and BPMF. Meanwhile, the variational inference for 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

$$\begin{aligned} \{\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_j\} &= \underset{(\mathbf{u}_i, \mathbf{v}_j)}{\mathrm{argmax}}\ p(\mathbf{u}_i, \mathbf{v}_j|R, \mathcal{P}) \\ &\approx \underset{(\mathbf{u}_i, \mathbf{v}_j)}{\mathrm{argmax}}\ q(\mathbf{u}_i, \mathbf{v}_j|\mathcal{P}') = \{\boldsymbol{\lambda}_{1i}, \boldsymbol{\lambda}_{2j}\} \ , \end{aligned}$$

so we have $\hat{R}_{ij} = \boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{2j}$.

B. Mixture PMF

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 $\boldsymbol{\rho}_j \sim \mathrm{Dirichlet}(\boldsymbol{\alpha})$.
   b) Pick a Gaussian $t_j \sim \mathrm{discrete}(\boldsymbol{\rho}_j)$.
   c) Generate $\mathbf{v}_j \sim p(\mathbf{v}_j|\boldsymbol{\mu}_{2,1:L}, \Sigma_{2,1:L}, t_j)$.

3) For each non-missing entry (i, j) in R, generate $R_{ij} \sim N(\mathbf{u}_i^T \mathbf{v}_j, \tau^2)$.

The likelihood of observing R is given by

$$p(R|\mathcal{P}) = \int_{\mathbf{u}_{1:N}} \int_{\mathbf{v}_{1:M}} \prod_{i=1}^{N} p(\mathbf{u}_i|\boldsymbol{\mu}_1, \Sigma_1) \Big(\prod_{j=1}^{M} p(\boldsymbol{\rho}_j|\boldsymbol{\alpha}) p(t_j|\boldsymbol{\rho}_j) \\ p(\mathbf{v}_j|\boldsymbol{\mu}_{2,1:L}, \Sigma_{2,1:L}, t_j) \Big) \prod_{i=1}^{N}\prod_{j=1}^{M} p(R_{ij}|\mathbf{u}_i^T \mathbf{v}_j, \tau^2)^{\delta_{ij}} \\ d\mathbf{u}_{1:N} d\mathbf{v}_{1:M}, \tag{9}$$

where $\mathcal{P} = \{\boldsymbol{\mu}_1, \Sigma_1, \boldsymbol{\mu}_{2l}, \Sigma_{2l}, \tau^2, \boldsymbol{\alpha}, [l]_1^L\}$.

There are 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 Dirichlet-discrete prior ($\boldsymbol{\alpha} \to \boldsymbol{\rho}_j$) becomes useful when we combine MPMF with LDA as discussed in Section IV-B.

For inference and learning, MPMF follows a similar approach as in PPMF, except that it uses a more complicated variational distribution q. Let $\mathcal{P}' = \{\boldsymbol{\lambda}_{1i}, \boldsymbol{\nu}_{1i}^2, \boldsymbol{\lambda}_{2jl}, \boldsymbol{\nu}_{2jl}^2, \boldsymbol{\gamma}_j, \boldsymbol{\phi}_j, [j]_1^M, [l]_1^L\}$, the variational distribution q for MPMF is given by

$$q(\mathbf{u}_{1:N}, \mathbf{v}_{1:M}, t_{1:M}, \boldsymbol{\rho}_{1:M}|\mathcal{P}') = \prod_{i=1}^{N} q(\mathbf{u}_i|\boldsymbol{\lambda}_{1i}, \text{diag}(\boldsymbol{\nu}_{1i}^2)) \\ \Big(\prod_{j=1}^{M} q(\boldsymbol{\rho}_j|\boldsymbol{\gamma}_j) q(t_j|\boldsymbol{\phi}_j) q(\mathbf{v}_j|\boldsymbol{\lambda}_{2j,1:L}, \text{diag}(\boldsymbol{\nu}_{2j,1:L}^2), t_j) \Big) \tag{10}$$

where for each $\mathbf{u}_i$ we still have a k-dimensional Gaussian, and for each $\mathbf{v}_j$ we have an L-dimensional Dirichlet($\boldsymbol{\gamma}_j$) to generate $\boldsymbol{\rho}_j$, an L-dimensional discrete($\boldsymbol{\phi}_j$) to generate t_j, and a mixture of k-dimensional Gaussians to generate $\mathbf{v}_j$.

The updating equations are given by:

E-step:

$$\boldsymbol{\lambda}_{1i} = \Big(\Sigma_1^{-1} + \frac{1}{\tau^2} \sum_{j=1}^{M} \delta_{ij} \sum_{l=1}^{L} \phi_{jl} (\boldsymbol{\lambda}_{2jl} \boldsymbol{\lambda}_{2jl}^T + \text{diag}(\boldsymbol{\nu}_{2jl}^2)) \Big)^{-1} \\ \Big(\Sigma_1^{-1} \boldsymbol{\mu}_1 + \frac{1}{\tau^2} \sum_{j=1}^{M} \delta_{ij} R_{ij} \sum_{l=1}^{L} \phi_{jl} \boldsymbol{\lambda}_{2jl} \Big) \tag{11}$$

$$\boldsymbol{\lambda}_{2jl} = \Big(\Sigma_{2l}^{-1} + \frac{1}{\tau^2} \sum_{i=1}^{N} \delta_{ij} (\boldsymbol{\lambda}_{1i} \boldsymbol{\lambda}_{1i}^T + \text{diag}(\boldsymbol{\nu}_{1i}^2)) \Big)^{-1} \\ \Big(\Sigma_{2l}^{-1} \boldsymbol{\mu}_2 + \frac{1}{\tau^2} \sum_{i=1}^{N} \delta_{ij} R_{ij} \boldsymbol{\lambda}_{1i} \Big) \tag{12}$$

$$\nu_{1ic}^2 = \Big(\sum_{j=1}^{M} \delta_{ij} \sum_{l=1}^{L} \phi_{jl} (\lambda_{2jlc}^2 + \nu_{2jlc}^2)/\tau^2 + \Sigma_{1,cc}^{-1} \Big)^{-1} \tag{13}$$

$$\nu_{2jlc}^2 = \Big(\sum_{i=1}^{N} \delta_{ij} (\lambda_{1ic}^2 + \nu_{1ic}^2)/\tau^2 + \Sigma_{2l,cc}^{-1} \Big)^{-1} \tag{14}$$

$$\phi_{jl} \propto \exp\Big(\Psi(\gamma_{jl}) - \Psi(\sum_{l'=1}^{L} \gamma_{jl}) - \frac{1}{2}\text{Tr}(\Sigma_{2l}^{-1} \text{diag}(\nu_{2jl}^2)) \\ - \frac{1}{2} (\boldsymbol{\lambda}_{2jl} - \boldsymbol{\mu}_{2l})^T \Sigma_{2l}^{-1} (\boldsymbol{\lambda}_{2jl} - \boldsymbol{\mu}_{2l}) + \frac{1}{2} \log|\Sigma_{2l}^{-1}| \\ - \frac{1}{2\tau^2} \sum_{i=1}^{N} \delta_{ij} \big(-2R_{ij} \boldsymbol{\lambda}_{1i}^T \boldsymbol{\lambda}_{2jl} + (\boldsymbol{\lambda}_{1i}^T \boldsymbol{\lambda}_{2jl})^2 \\ + \boldsymbol{\lambda}_{1i}^T \text{diag}(\boldsymbol{\nu}_{2jl}^2) \boldsymbol{\lambda}_{1i} + \boldsymbol{\lambda}_{2jl}^T \text{diag}(\boldsymbol{\nu}_{1i}^2) \boldsymbol{\lambda}_{2jl} \\ + \text{Tr}(\text{diag}(\nu_{1i}^2) \text{diag}(\boldsymbol{\nu}_{2jl}^2)) \big) + \frac{1}{2} \log(\nu_{2jlc}^2) \Big) \tag{15}$$

$$\gamma_{jl} = \alpha_l + \phi_{jl} \tag{16}$$

where Ψ is the digamma function.

M-step:

$$\boldsymbol{\mu}_1 = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\lambda}_{1i} \tag{17}$$

$$\boldsymbol{\mu}_{2l} = \sum_{j=1}^{M} \phi_{jl} \boldsymbol{\lambda}_{2jl} / \sum_{j=1}^{M} \phi_{jl} \tag{18}$$

$$\Sigma_1 = \frac{1}{N} \sum_{i=1}^{N} \big(\text{diag}(\boldsymbol{\nu}_{1i}^2) + (\boldsymbol{\lambda}_1 - \boldsymbol{\mu}_1)(\boldsymbol{\lambda}_1 - \boldsymbol{\mu}_1)^T \big) \tag{19}$$

$$\Sigma_{2l} = \sum_{j=1}^{M} \phi_{jl} \big(\text{diag}(\boldsymbol{\nu}_{2jl}^2) + (\boldsymbol{\lambda}_{2jl} - \boldsymbol{\mu}_2)(\boldsymbol{\lambda}_{2jl} - \boldsymbol{\mu}_2)^T \big) / \sum_{j=1}^{M} \phi_{jl} \tag{20}$$

$$\tau^2 = \frac{1}{A} \sum_{i=1}^{N} \sum_{j=1}^{M} \delta_{ij} \Big(R_{ij}^2 - 2R_{ij} \boldsymbol{\lambda}_{1i}^T \sum_{l=1}^{L} \phi_{jl} \boldsymbol{\lambda}_{2jl} \\ + \sum_{l=1}^{L} \phi_{jl} \big((\boldsymbol{\lambda}_{1i}^T \boldsymbol{\lambda}_{2jl})^2 + \boldsymbol{\lambda}_{1i}^T \text{diag}(\boldsymbol{\nu}_{2jl}^2) \boldsymbol{\lambda}_{1i} + \boldsymbol{\lambda}_{2jl}^T \text{diag}(\boldsymbol{\nu}_{1i}^2) \boldsymbol{\lambda}_{2jl} \\ + \text{Tr}(\text{diag}(\boldsymbol{\nu}_{1i}^2) \text{diag}(\boldsymbol{\nu}_{2jl}^2)) \big) \Big) \tag{21}$$

$\boldsymbol{\alpha}$ could be updated using Newton-Raphson algorithm [5] as

$$\alpha_l' = \alpha_l - \zeta(x_l - y)/r_l \ , \tag{22}$$

where ζ is the learning rate and

$$x_l = M \Big(\Psi(\sum_{l'=1}^{L} \alpha_{l'}) - \Psi(\alpha_l) \Big) + \sum_{j=1}^{M} \Big(\Psi(\gamma_{jl}) - \Psi(\sum_{l'=1}^{L} \gamma_{jl}) \Big)$$

$$r_l = -M \Psi'(\alpha_l)$$

$$y = (\sum_{l=1}^{L} x_l / r_l) / (1/e + \sum_{l=1}^{L} 1/r_l)$$

$$e = M \Psi'(\sum_{l=1}^{L} \alpha_l) \ .$$

As in PPMF, the algorithm iterates through E-step and M-step until convergence, and in the E-step, the algorithm updates all variational parameters alternatively until convergence. The complexity for each E-step is $O((k^3 N +$

$k^2NML + k^3ML + ML^2)t_E$). For each M-step, the complexity for updating α is $O(MLt_\alpha)$ given t_α the number of iterations to update α, and the complexity for updating other model parameters is $O(k^2N + k^2ML + kNML)$.

For the prediction of MPMF, we use MAP estimate as in PPMF, i.e., $\hat{R}_{ij} = \hat{\mathbf{u}}_i^T \hat{\mathbf{v}}_j$, where $\hat{\mathbf{u}}_i = \boldsymbol{\lambda}_{1i}$, and $\hat{\mathbf{v}}_j = \sum_{l=1}^{L} \phi_{jl} \boldsymbol{\lambda}_{2jl}$. Therefore, $\hat{R}_{ij} = \boldsymbol{\lambda}_{1i}^T \sum_{l=1}^{L} \phi_{jl} \boldsymbol{\lambda}_{2jl}$.

IV. GPMFs on Rating Matrix with Side Information

PPMF and MPMF work on the rating matrix only. In this section, we propose two models to incorporate the side information while performing matrix factorization. The idea of using side information to help matrix factorization could be considered as a combination of two basic algorithms in collaborative filtering—matrix factorization based algorithms and neighborhood based algorithms. We use matrix factorization on the rating matrix, meanwhile, we hope that the low-dimensional latent feature vectors $\mathbf{v}$ for two movies are close to each other if they are neighbors based on side information, e.g. they have similar casts. In this paper, we assume that we only have side information on movies, such as movie's cast, plot, genre, etc..

Topic modeling algorithms such as CTM and LDA work on documents, and matrix factorization algorithms such as PPMF and MPMF work on the rating matrix. Therefore, given the data like "ratings+movie plots", it is a natural idea to combine topic models with matrix factorization, in order to help rating prediction using movie plots. 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 actor/actress names, movie 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 corresponding 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 $z_{jh} \sim \text{discrete}(\text{logistic}(\mathbf{v}_j))$.
   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 $\text{logistic}(\mathbf{v}_j) = \frac{\exp(\mathbf{v}_j)}{\sum_{c=1}^{k} \exp(v_{jc})}$ 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. Let $\mathcal{P} = \{\boldsymbol{\mu}_1, \Sigma_1, \boldsymbol{\mu}_2, \Sigma_2, \tau^2, \boldsymbol{\beta}_c, [c]_1^k\}$, the likelihood function of observing R and W is

$$
\begin{aligned}
&p(R, W|\mathcal{P}) \qquad (23)\\
&= \int_{\mathbf{u}_{1:N}} \int_{\mathbf{v}_{1:M}} \left(\prod_{j=1}^{M} p(\mathbf{v}_j|\boldsymbol{\mu}_2, \Sigma_2) p(\mathbf{z}_j|\mathbf{v}_j) p(\mathbf{w}_j|\mathbf{z}_j, \boldsymbol{\beta}_{1:k}) \right)\\
&\prod_{i=1}^{N} p(\mathbf{u}_i|\boldsymbol{\mu}_1, \Sigma_1) \prod_{i=1}^{N} \prod_{j=1}^{M} p(R_{ij}|\mathbf{u}_i^T \mathbf{v}_j, \tau^2)^{\delta_{ij}} d\mathbf{u}_{1:N} d\mathbf{v}_{1:M} \ .
\end{aligned}
$$

The inference and learning for CTM-PPMF is similar to that of PPMF, except that we need to introduce discrete$(\boldsymbol{\phi}_j)$ in the variational distribution to generate $\mathbf{z}_j$, so we have

$$
\begin{aligned}
q(\mathbf{u}_{1:N}, \mathbf{v}_{1:M}|\mathcal{P}') = &\prod_{i=1}^{N} q(\mathbf{u}_i|\boldsymbol{\lambda}_{1i}, \text{diag}(\boldsymbol{\nu}_{1i}^2)) \qquad (24)\\
&\prod_{j=1}^{M} \Big(q(\mathbf{v}_j|\boldsymbol{\lambda}_{2j}, \text{diag}(\boldsymbol{\nu}_{2j}^2)) \prod_{h=1}^{S_j} q(z_{jh}|\boldsymbol{\phi}_j) \Big),
\end{aligned}
$$

where S_j is the number of total words in $\mathbf{w}_j$ and $\mathcal{P}' = \{\boldsymbol{\lambda}_{1i}, \boldsymbol{\nu}_{1i}^2, \boldsymbol{\lambda}_{2j}, \boldsymbol{\nu}_{2j}^2, \boldsymbol{\phi}_j, [i]_1^N, [j]_1^M\}$.

For the update equations, in the E-step, the equations for $\boldsymbol{\lambda}_{1i}$ and $\boldsymbol{\nu}_{1ic}$ are the same as in PPMF, and the equations for $\boldsymbol{\lambda}_{2j}$, $\boldsymbol{\nu}_{2jc}$ and ϕ_{jc} are given by

$$
\begin{aligned}
\boldsymbol{\lambda}_{2j} = &\Big(\Sigma_2^{-1} + \frac{1}{\tau^2} \sum_{i=1}^{N} \delta_{ij} (\boldsymbol{\lambda}_{1i} \boldsymbol{\lambda}_{1i}^T + \text{diag}(\boldsymbol{\nu}_{1i}^2)) + S_j H(\boldsymbol{\xi}_j) \Big)^{-1}\\
&\Big(\Sigma_2^{-1} \boldsymbol{\mu}_2 + \frac{1}{\tau^2} \sum_{i=1}^{N} \delta_{ij} R_{ij} \boldsymbol{\lambda}_{1i} - \frac{S_j \exp(\boldsymbol{\xi}_j)}{\sum_{c=1}^{k} \exp(\xi_{jc})}\\
&+ S_j H(\boldsymbol{\xi}_j) \boldsymbol{\xi}_j + S_j \boldsymbol{\phi}_j \Big) \qquad (25)
\end{aligned}
$$

$$
\begin{aligned}
\nu_{2jc} = &\Big(S_j \exp(\xi_{jc}) (\sum_{c'=1}^{k} \exp(\xi_{jc'}) - \exp(\xi_{jc})) + \Sigma_{2,cc}^{-1}\\
&+ \frac{1}{\tau^2} \sum_{i=1}^{N} \delta_{ij} (\lambda_{1ic}^2 + \nu_{1ic}^2) \Big)^{-1} \qquad (26)
\end{aligned}
$$

$$
\phi_{jc} \propto \exp\Big(\lambda_{2jc} + \frac{1}{S_j} \sum_{h=1}^{S_j} \sum_{d=1}^{D} \mathbf{1}(w_{jh} \sim d) \log \beta_{cd} \Big) \qquad (27)
$$

Here $\boldsymbol{\xi}_j$ is a new variational parameter, and in each iteration of EM, it takes the value of $\boldsymbol{\lambda}_{2j}$ from the last iteration. $\mathbf{1}(w_{jh} \sim d)$ is an indicator taking value 1 if w_{jh} is the d^{th} word in the dictionary and 0 otherwise. $H(\boldsymbol{\xi}_j)$ is given by

$$
H(\boldsymbol{\xi}_j) = \frac{\text{diag}(\exp(\boldsymbol{\xi}_j))}{\sum_{c=1}^{k} \exp(\xi_{jc})} - \frac{\exp(\boldsymbol{\xi}_j) \exp(\boldsymbol{\xi}_j)^T}{\Big(\sum_{c=1}^{k} \exp(\xi_{jc}) \Big)^2} \ .
$$

The update equations for $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, Σ_1, and Σ_2 in the M-step are the same as in PPMF. For $\boldsymbol{\beta}$, we have

$$\beta_{cd} \propto \sum_{j=1}^{M} \phi_{jc} \sum_{h=1}^{S_j} \mathbf{1}(w_{jh} \sim d) \ . \tag{28}$$

For each E-step, the time complexity for updating $\boldsymbol{\phi}$ is $O(kMSDt_E)$, where $S = \max\{S_j, [j]_1^M\}$, and the complexity for updating the rest variational parameters is $O(k^2(kM + kN + NM)t_E)$. For each M-step, the time complexity is $O(kMSD)$ for updating $\boldsymbol{\beta}$ and $O(k^2N + k^2M + kMN)$ for updating other model parameters.

For CTM-PPMF, the prediction of the missing entry is the same as PPMF, i.e., $\hat{R}_{ij} = \boldsymbol{\lambda}_{1i}^T \boldsymbol{\lambda}_{2j}$.

B. LDA-MPMF

The main idea of LDA-MPMF is as follows: For MPMF on the rating matrix, each movie has a Dirichlet-discrete prior ($\boldsymbol{\alpha} \rightarrow \boldsymbol{\rho}_j$). Meanwhile, if we use LDA on side information, each movie also needs a Dirichlet-discrete prior to generate the topics of words in its 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 $\mathbf{v}_j \sim p(\mathbf{v}_j|\boldsymbol{\mu}_{2,1:L}, \Sigma_{2,1:L}, t_j)$.
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. Letting $\mathcal{P} = \{\boldsymbol{\mu}_1, \Sigma_1, \boldsymbol{\mu}_{2l}, \Sigma_{2l}, \tau^2, \boldsymbol{\alpha}, , \boldsymbol{\beta}_l, [l]_1^L\}$, the likelihood function of observing R and W is given by

$$p(R, W|\mathcal{P}) = \int_{\mathbf{u}_{1:N}} \int_{\mathbf{v}_{1:M}} \prod_{i=1}^{N} p(\mathbf{u}_i|\boldsymbol{\mu}_1, \Sigma_1) \tag{29}$$

$$\Big(\prod_{j=1}^{M} p(\boldsymbol{\rho}_j|\boldsymbol{\alpha}) p(t_j|\boldsymbol{\rho}_j) p(\mathbf{v}_j|\boldsymbol{\mu}_{2,1:L}, \Sigma_{2,1:L}, t_j) p(\mathbf{z}_j|\boldsymbol{\rho}_j)$$

$$p(\mathbf{w}_j|\mathbf{z}_j, \boldsymbol{\beta}_{1:L})\Big) \prod_{i=1}^{N} \prod_{j=1}^{M} p(R_{ij}|\mathbf{u}_i^T \mathbf{v}_j, \tau^2)^{\delta_{ij}} d\mathbf{u}_{1:N} d\mathbf{v}_{1:M} \ .$$

Looking at the graphical models of MPMF and LDA-MPMF in Figure 1, as we have mentioned, MPMF is the intermediate model between PPMF and LDA-MPMF, so although the prior $\boldsymbol{\alpha} \rightarrow \boldsymbol{\rho}_j$ seems to be redundant in MPMF, it becomes useful when combining with LDA.

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 of movies 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 $\mathbf{v}$ for movies aposteriori, hence similar ratings. In LDA-MPMF, the ratings and 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 their $\mathbf{v}$ would probably be generated from a same Gaussian in the mixture, hence are similar. 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.

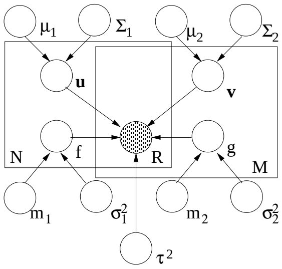


Figure 2. Graphical model for residual PPMF.

The variational distribution q for LDA-MPMF is

$$q(\mathbf{u}_{1:N}, \mathbf{v}_{1:M}, t_{1:M}, \boldsymbol{\rho}_{1:M}, \mathbf{z}_{1:M}|\mathcal{P}') = \prod_{i=1}^{N} q(\mathbf{u}_i|\boldsymbol{\lambda}_{1i}, \text{diag}(\boldsymbol{\nu}_{1i}^2))$$

$$\Big(\prod_{j=1}^{M} q(\boldsymbol{\rho}_j|\boldsymbol{\gamma}_j) q(t_j|\boldsymbol{\phi}_j) q(\mathbf{v}_j|\boldsymbol{\lambda}_{2j,1:L}, \text{diag}(\boldsymbol{\nu}_{2j,1:L}^2), t_j) q(\mathbf{z}_j|\boldsymbol{\phi}_j)\Big), \tag{30}$$

where $\mathcal{P}'$ is the same as in MPMF. For the updating equations, in the E-step, the equations for $\boldsymbol{\lambda}_{1i}$, $\boldsymbol{\lambda}_{2jl}$, ν_{1ic} and ν_{2jlc} are the same as in MPMF, and the equations for ϕ_{jl} and γ_{jl} are

$$\phi_{jl} \propto \exp\Big\{\big[\big(\Psi(\gamma_{jl}) - \Psi(\sum_{l'=1}^{L} \gamma_{jl})\big)(S_j + 1) + \frac{1}{2} \log |\Sigma_{2l}^{-1}|$$

$$-\frac{1}{2}\text{Tr}(\Sigma_{2l}^{-1} \text{diag}(\nu_{2jl}^2)) - \frac{1}{2}(\boldsymbol{\lambda}_{2jl} - \boldsymbol{\mu}_{2l})^T \Sigma_{2l}^{-1} (\boldsymbol{\lambda}_{2jl} - \boldsymbol{\mu}_{2l})$$

$$+\sum_{h=1}^{S_j}\sum_{d=1}^{D} \mathbf{1}(w_{jh} \sim d) \log \beta_{ld} - \frac{1}{2\tau^2}\sum_{i=1}^{N} \delta_{ij}\Big(-2R_{ij} \boldsymbol{\lambda}_{1i}^T \boldsymbol{\lambda}_{2jl}$$

$$+(\boldsymbol{\lambda}_{1i}^T \boldsymbol{\lambda}_{2jl})^2 + \boldsymbol{\lambda}_{1i}^T \text{diag}(\boldsymbol{\nu}_{2jl}^2) \boldsymbol{\lambda}_{1i} + \boldsymbol{\lambda}_{2jl}^T \text{diag}(\boldsymbol{\nu}_{1i}^2) \boldsymbol{\lambda}_{2jl}$$

$$+\text{Tr}(\text{diag}(\nu_{1i}^2) \text{diag}(\boldsymbol{\nu}_{2jl}^2))\Big) + \frac{1}{2} \log(\nu_{2jlc}^2)\big] / (1 + S_j)\Big\} \tag{31}$$

$$\gamma_{jl} = \alpha_l + (1 + S_j)\phi_{jl} \ . \tag{32}$$

In the M-step, the updating equation for $\boldsymbol{\beta}$ is given by

$$\beta_{ld} \propto \sum_{j=1}^{M} \phi_{jl} \sum_{h=1}^{S_j} \sum_{d=1}^{D} \mathbf{1}(w_{jh} \sim d) \ , \tag{33}$$

where D is the total number of words in the dictionary. The equations for other parameters are the same as in MPMF.

In each E-step, the time complexity is $O(ML(L+k^3+SD+Nk)t_E)$ for updating ϕ and $O((k^3N+k^2MNL+k^3ML)t_E)$ for updating the rest of the variational parameters. In each M-step, the complexity for updating $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is $O(MLt_\alpha)$ and $O(MSD^2L)$ respectively, and the complexity for updating the rest model parameters is $O(k^2N+k^2ML+kNML)$.

For prediction of missing entry in LDA-MPMF, the prediction is the same as MPMF, i.e., $\hat{R}_{ij} = \boldsymbol{\lambda}_{1i}^T\sum_{l=1}^L \phi_{jl}\boldsymbol{\lambda}_{2jl}$.

V. Residual GPMFs

There are usually biases in the ratings. For example, a popular movie usually receives high ratings, and a critical user usually 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}, which gives residual GPMFs.

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

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

$$p(R|\mathcal{P})=\int_{\mathbf{u}_{1:N}}\int_{\mathbf{v}_{1:M}}\int_{f_{1:N}}\int_{g_{1:M}}\Big(\prod_{i=1}^N p(\mathbf{u}_i|\boldsymbol{\mu}_1,\Sigma_1)p(f_i|m_1,\sigma_1^2)\Big)$$
$$\Big(\prod_{j=1}^M p(\mathbf{v}_j|\boldsymbol{\mu}_2,\Sigma_2)p(g_j|m_2,\sigma_2^2)\Big)\prod_{i=1}^N\prod_{j=1}^M p(R_{ij}|\mathbf{u}_i^T\mathbf{v}_j+f_i+g_j,\tau^2)^{\delta_{ij}}$$
$$d\mathbf{u}_{1:N}d\mathbf{v}_{1:M}df_{1:N}dg_{1:M} \tag{34}$$

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:

$$q(\mathbf{u}_{1:N},\mathbf{v}_{1:M}|\mathcal{P}') = \Big(\prod_{i=1}^N q(\mathbf{u}_i|\boldsymbol{\lambda}_{1i},\text{diag}(\boldsymbol{\nu}_{1i}^2))q(f_i|\theta_{1i},\eta_{1i}^2)\Big)$$
$$\Big(\prod_{j=1}^M q(\mathbf{v}_j|\boldsymbol{\lambda}_{2j},\text{diag}(\boldsymbol{\nu}_{2j}^2))q(g_j|\theta_{2j},\eta_{2j}^2)\Big)\ , \tag{35}$$

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

The updating equations for $\boldsymbol{\nu}_1$, $\boldsymbol{\nu}_2$, $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ and Σ_1, Σ_2 are the same as in PPMF. For the rest parameters, we have:
E-step:

$$\boldsymbol{\lambda}_{1i} =\Big(\Sigma_1^{-1}+\frac{1}{\tau^2}\sum_{j=1}^M\delta_{ij}(\boldsymbol{\lambda}_{2j}\boldsymbol{\lambda}_{2j}^T+\text{diag}(\boldsymbol{\nu}_{2j}^2))\Big)^{-1}$$
$$\Big(\Sigma_1^{-1}\boldsymbol{\mu}_1+\frac{1}{\tau^2}\sum_{j=1}^M\delta_{ij}(R_{ij}-\theta_{1i}-\theta_{2j})\boldsymbol{\lambda}_{2j}\Big) \tag{36}$$

$$\theta_{1i} =\Big(\frac{1}{\sigma_1^2}+\frac{1}{\tau^2}\sum_{j=1}^M\delta_{ij}\Big)^{-1}\Big(\frac{m_1}{\sigma_1^2}+\frac{1}{\tau^2}\sum_{j=1}^M(R_{ij}-\theta_{2j}-\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{2j})\Big) \tag{37}$$

$$\eta_{1i}^2 =\Big(1/\sigma_1^2+\sum_{j=1}^M\delta_{ij}/\tau^2\Big)^{-1}\ . \tag{38}$$

The expressions for $\boldsymbol{\lambda}_{2j}$, θ_{2j} and η_{2j}^2 are similar.
M-step:

$$\tau^2 =\frac{1}{A}\sum_{i=1}^N\sum_{j=1}^M\delta_{ij}\Big(R_{ij}^2+\boldsymbol{\lambda}_{1i}^T\text{diag}(\boldsymbol{\nu}_{2j}^2)\boldsymbol{\lambda}_{1i}+\boldsymbol{\lambda}_{2j}^T\text{diag}(\boldsymbol{\nu}_{1i}^2)\boldsymbol{\lambda}_{2j}$$
$$-2R_{ij}(\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{2j}^T+\theta_{1i}+\theta_{2j})+(\boldsymbol{\lambda}_{1i}^T\boldsymbol{\lambda}_{2j}+\theta_{1i}+\theta_{2j})^2$$
$$+\text{Tr}(\text{diag}(\boldsymbol{\nu}_{1i}^2)\text{diag}(\boldsymbol{\nu}_{2j}^2))+\eta_{1i}^2+\eta_{2j}^2\Big) \tag{39}$$

$$m_1 =\frac{1}{N}\sum_{i=1}^N\theta_{1i} \tag{40}$$

$$\sigma_1^2 =\frac{1}{N}\sum_{i=1}^N((\theta_{1i}-m_1)^2+\eta_{1i}^2)\ . \tag{41}$$

The expressions for m_2, and σ_2^2 are similar.

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. The update equations could be easily derived then.

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 $\hat{g}_j=\theta_{2j}$.

VI. 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].

[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.

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

(a) On movielens

k	5	10	15	20	25	30
specCoc s2	0.1178 ±0.0016	0.1160 ±0.0020	0.1163 ±0.0025	0.1157 ±0.0028	0.1103 ±0.0013	0.1103 ±0.0014
specCoc s5	0.0983 ±0.0011	0.0958 ±0.0017	0.0962 ±0.0023	0.0959 ±0.0021	0.0942 ±0.0013	0.0943 ±0.0011
BregCoc s2	0.0943 ±0.0015	0.0949 ±0.0014	0.0953 ±0.0014	0.0959 ±0.0013	0.0966 ±0.0020	0.0973 ±0.0016
BregCoc s5	0.0993 ±0.0012	0.1021 ±0.0017	0.1030 ±0.0020	0.1049 ±0.0017	0.1063 ±0.0021	0.1078 ±0.0018
BCC	0.1139 ±0.0011	0.1116 ±0.0015	0.1119 ±0.0011	0.1119 ±0.0015	0.1106 ±0.0018	0.1114 ±0.0022
PPMF	**0.0862** **±0.0011**	**0.0863** **±0.0010**	**0.0859** **±0.0011**	**0.0854** **±0.0013**	**0.0853** **±0.0011**	**0.0855** **±0.0012**

(b) On million-movielens

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.

(a) On movielens

k	5	10	15	20	25	30
PMF	0.09172 ±0.00230	0.09702 ±0.00179	0.10048 ±0.00194	0.10106 ±0.00209	0.10235 ±0.00182	0.10350 ±0.00123
BPMF	0.08954 ±0.00104	0.09040 ±0.00117	0.09269 ±0.00129	0.09638 ±0.00144	0.10121 ±0.00152	0.10664 ±0.00152
PPMF	**0.08620** **±0.00112**	**0.08629** **±0.00103**	**0.08587** **±0.00111**	**0.08544** **±0.00137**	**0.08533** **±0.00112**	**0.08551** **±0.00122**

(b) On million-movielens

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**

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

(a) On movielens

	original	residual
PPMF	0.08551 ±0.00122	**0.08502** **±0.00119**
CTM-PPMF cast	0.08695 ±0.01486	**0.08615** **±0.00120**
CTM-PPMF plot	0.08913 ±0.00153	**0.08803** **±0.00127**
CTM-PPMF genre	0.08589 ±0.00139	**0.08538** **±0.00108**
MPMF	**0.08628** **±0.00115**	0.08753 ±0.00110
LDA-MPMF cast	0.08774 ±0.00164	**0.08739** **±0.00105**
LDA-MPMF plot	**0.08747** **±0.00121**	0.08750 ±0.00110
LDA-MPMF genre	**0.08693** **±0.00167**	0.08754 ±0.00156

(b) On million-movielens

	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**

We use two movielens datasets[2]. One (movielens) contains 100,000 ratings for 1682 movies by 943 users, the other (million-movielens) contains 1 million ratings for 3900 movies by 6040 users. For each movie we extract three types of side information from IMDB[3]—cast, genre, and plot. For genre, there are 25 movie types in both datasets. For cast, we only use the top-10 ranked most important actors/actresses in each movie, and there are totally 7099 and 13924 actors/actresses in movielens and million-movielens respectively. For plot, we use the plots written by imdb users. After preprocessing the text, there are totally 2791 and 2693 words in the dictionary of movielens and million-movielens respectively. 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 the ratings is held out as the validation set, which is used in the training process to decide the stopping time for variational EM iterations. 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, one of which is picked as the test set, and the remaining 9 parts are used as the training set. The process is repeated for 10 times, with each part used once as the test set. We then take the average MSE over 10 folds on the test set. Before running the algorithm, we transform each rating R_{ij} to $\sqrt{6 - R_{ij}}$, such that the ratings are closer to a Gaussian distribution [2].

A. PPMF vs. Co-clustering Algorithms

We first compare PPMF to co-clustering based algorithms. Performing co-clustering on a matrix gives the membership vectors for rows/columns over row/column clusters, as well as certain statistics for each row-column cluster, i.e., co-cluster. For a certain entry (i, j) in the matrix R, after learning the membership vectors for row i and column j, as well as the statistics for all co-clusters, we can predict R_{ij}. In our experiment, we compare PPMF with three co-clustering algorithms: spectral co-clustering (specCoc) [7], Bregman co-clustering (BregCoc) [3], and Bayesian co-clustering (BCC) [18]. We use two schemes (s2 and s5) for specCoc and BregCoc, with different schemes keeping different types of statistics [3]. For initialization of co-clustering algorithms, we run k-means on the low-rank vectors from imputed singular value decomposition (SVD), and use the result membership vectors for initialization. For PPMF, we use random initialization. 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

[2] www.movielens.umn.edu

[3] www.imdb.com

Table IV
MSE FROM GENERALIZED PMFS.

(a) On movielens

k	5	10	15	20	25	30
PPMF	**0.08620** **±0.00112**	**0.08629** **±0.00103**	**0.08587** **±0.00111**	**0.08544** **±0.00137**	**0.08533** **±0.00112**	**0.08551** **±0.00122**
CTM -PPMF cast	0.08741 ±0.00125	0.08663 ±0.00121	0.08633 ±0.00127	0.08650 ±0.00169	0.08686 ±0.01262	0.08695 ±0.01486
CTM -PPMF plot	0.08849 ±0.00124	0.08812 ±0.00123	0.08817 ±0.00142	0.08846 ±0.00148	0.08849 ±0.00120	0.08913 ±0.00154
CTM -PPMF genre	0.08694 ±0.00112	0.08728 ±0.00143	0.08607 ±0.00121	0.08615 ±0.00141	0.08578 ±0.00132	0.08589 ±0.00139
MPMF	0.08765 ±0.00079	0.08683 ±0.00112	0.08641 ±0.00127	0.08598 ±0.00106	0.08613 ±0.00126	0.08628 ±0.00115
LDA -MPMF cast	0.08766 ±0.00065	0.08826 ±0.00099	0.08763 ±0.00140	0.08749 ±0.00134	0.08737 ±0.00119	0.08774 ±0.00164
LDA -MPMF plot	0.08788 ±0.00096	0.08790 ±0.00081	0.08813 ±0.00118	0.08756 ±0.00169	0.08735 ±0.00104	0.08747 ±0.00121
LDA -MPMF genre	0.08719 ±0.00104 ●	0.08737 ±0.00134	0.08770 ±0.00104	0.08683 ±0.00093	0.08649 ±0.00122	0.08693 ±0.00167

(b) On million-movielens

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

Table V
MSE FROM RESIDUAL MODELS OF GENERALIZED PMFS.

(a) On movielens

k	5	10	15	20	25	30
rsPPMF	0.08605 ±0.00100	0.08671 ±0.00129	**0.08559** **±0.00109**	0.08517 ±0.00111	**0.08506** **±0.00117**	**0.08502** **±0.00119**
rsCTM -PPMF cast	**0.08576** **±0.00077** ●	0.08634 ±0.00134 ●	0.08589 ±0.00096	0.08621 ±0.00112	0.08629 ±0.00108	0.08615 ±0.00120
rsCTM -PPMF plot	0.08678 ±0.00110	0.08685 ±0.00073	0.08699 ±0.00119	0.08731 ±0.00102	0.08730 ±0.00128	0.08803 ±0.00127
rsCTM -PPMF genre	0.08590 ±0.00092 ●	**0.08599** **±0.00106** ●	0.08565 ±0.00078	**0.08509** **±0.00116** ●	0.08521 ±0.00116	0.08538 ±0.00108
rsMPMF	0.08764 ±0.00079	0.08683 ±0.00112	0.08640 ±0.00127	0.08598 ±0.00106	0.08724 ±0.00116	0.08753 ±0.00110
rsLDA -MPMF cast	0.08877 ±0.00136	0.08952 ±0.00170	0.08867 ±0.00123	0.08807 ±0.00146	0.08783 ±0.00141	0.08739 ±0.00105 ●
rsLDA -MPMF plot	0.08894 0.00128	0.08949 ±0.00150	0.08842 ±0.00145	0.08776 ±0.00111	0.08765 ±0.00112	0.08750 ±0.00110 ●
rsLDA -MPMF genre	0.08881 ±0.00154	0.08930 ±0.00134	0.08860 ±0.00117	0.08765 ±0.00104	0.08745 ±0.00130	0.08754 ±0.00156

(b) On million-movielens

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

see that PPMF clearly generates a smaller MSE compared to the co-clustering based algorithms. Co-clustering algorithms represent the neighborhood based algorithms since similar rows/columns will have similar membership vectors. While they 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. For PPMF and CTM-PPMF, the residual models generate higher accuracy in almost all cases; for MPMF and LDA-MPMF, the residual models also generate high accuracy on the larger dataset million-movielens. Such observation could also be obtained from Table IV and V when we provide the 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 for GPMFs—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 and V, with Table IV for original models and Table V 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 ● under the results of CTM-PPMF (LDA-MPMF) if

Table VI
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 Glengarry Glen Ross; The Right Stuff; Nixon Milk Money; Eye for an Eye; Just Cause
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

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 on movielens hurts the prediction most of the times, but the side information seems to help on the larger dataset million-movielens, especially for residual models. For CTM-PPMF compared to PPMF, the advantage of taking side information is more distinct. For both the original model and residual model of CTM-PPMF on million-movielens, and for the residual model of CTM-PPMF on movielens, we can see the increase of prediction accuracy, 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 one, 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. Meanwhile, the overlapping of cast among different movies is very small, i.e., 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 for the ratings. 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 VI shows two examples of top 10 actor/actress names in two cast groups. Table VI(a) is a list of actors/actresses acting in Star Trek or other science fiction movies. For better presentation, we also give the movie names they performed in. Table VI(b) is another group of cast. Given the year of movies they performed in, we can see these are actors/actresses mostly active as early as in 40's-60's, which is a distinct group since most movies in movielens are in 80's or 90's. For the word list of topics from plots, since the plots are highly subjective as we have discussed, we do not see very coherent topics.

VII. CONCLUSION

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

REFERENCES

[1] D. Agarwal and B. Chen. fLDA: Matrix factorization through latent Dirichlet allocation. In *WSDM*, 2010.

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

[3] 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.

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

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

[6] W. Chu and Z. Ghahramani. Probabilistic models for incomplete multi-dimensional arrays. In *AISTATS*, 2009.

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

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

[9] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval*, pages 50–57, Berkeley, California, August 1999.

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

[11] N. Lawrence and R. Urtasun. Non-linear matrix factorization with Gaussian processes. In *ICML*, 2009.

[12] L. Mackey, D. Weiss, and M. Jordan. Mixed membership matrix factorization. In *ICML*, 2010.

[13] B. Marlin. Modeling user rating profiles for collaborative filtering. In *NIPS*, 2003.

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

[15] I. Porteous, A. Asuncion, and M. Welling. Bayesian matrix factorization with side information and Dirichlet process mixtures. In *AAAI*, 2010.

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

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

[18] H. Shan and A. Banerjee. Bayesian co-clustering. In *ICDM*, pages 530–539, 2008.

[19] A. Singh and G. Gordon. A Bayesian matrix factorization model for relational data. In *UAI*, 2010.

[20] I. Sutskever, R. Salakhutdinov, and J. Tenenbaum. Modelling relational data using Bayesian clustered tensor facotrization. In *NIPS*, 2009.