A Logistic Factorization Model for Recommender Systems with Multinomial Responses

Yu Wang
Electrical and Computer Engineering, Duke University
and
Xuan Bi
Carlson School of Management, University of Minnesota
and
Annie Qu
Department of Statistics, University of Illinois at Urbana-Champaign

June 13, 2019

Abstract

In this paper, we propose a two-way multinomial logistic model for recommender systems for categorical ratings. Specifically, we treat the possible ratings as mutually exclusive events, whose probability is determined by the latent factor of the users and the items through a two-way multinomial logistic function. The proposed method has a compatibility with categorical ratings and the advantage of incorporating both the covariate information and the latent factors of the users and items uniformly. We show numerically that the proposed method performs consistently better than five commonly used collaborative filtering methods, namely, the restricted singular value decomposition, the soft-impute matrix completion method, the regression-based latent factor models, the restricted Boltzmann machine, and the group-specific recommender system on various simulation setups and on MovieLens data.

Keywords: collaborative filter, cold-start problem, MovieLens

1 Introduction

The study of recommender systems (Aggarwal, 2016) has flourished during the past decades (Resnick and Varian, 1997; Lu et al., 2015; Ricci et al., 2011) stimulated by real-world challenges from Netflix (Bell and Koren, 2007; Gomez-Uribe and Hunt, 2015), Yahoo! Music (Koenigstein et al., 2011), and YouTube (Davidson et al., 2010), to name a few. Given a partial record of the ratings given by a set of users on a set of items, we would like to infer how a user will rate an item that has never been rated before based on the past behavior of the users and the rating history on the item (Adomavicius and Tuzhilin, 2005; Kim et al., 2005; Aciar et al., 2006; Ghosh et al., 1999).

One of the major challenges in developing recommender systems that work in the real world is the sparsity of the rating matrix, as the number of the users and the items in a data is usually large and the average number of the ratings of a user or an item is relatively small. This leads to the situation where for many users and items in the testing set, there is no previous rating record in the training set to learn from. For example, in the MovieLens 10M data (Miller et al., 2003; Harper and Konstan, 2015) collected by GroupLens Research (`http://grouplens.org/datasets/movielens`), 96% of the most recent ratings are from new users or on new items with no previous rating record (Bi et al., 2016). This phenomenon is called the "cold-start" problem. When tackling this problem, the covariate information on the new users and items, for example, the gender, sex, and age of a user and the features of an item can provide a relief.

Another feature of many datasets is that the ratings are categorical, taking values from a finite set of numbers. For example, the ratings are integers from one to ten for IMDB (Oghina et al., 2012), and from one to five for the MovieLens data (Miller et al., 2003; Harper and Konstan, 2015). This inspires using categorical data models (Agresti and Kateri, 2011) to analyze the data.

Previously, two common approaches to predict a new rating are content-based filtering and collaborative filtering. For content-based filtering (Mooney and Roy, 2000; Blanco-Fernandez et al., 2008; Iaquinta et al., 2008; Lops et al., 2011; Mirbakhsh and Ling, 2015), a new rating of an item is generated by comparing its content with a user's profile. This method has the capability to handle "cold-start" problems with respect to items. In other words, the rating of a new item which has never been rated by any user before can be generated with relatively high accuracy (Levi et al., 2012). However, it cannot handle the "cold-start" problems with respect to users, since a past history is crucial to constructing a user's profile. In addition, sufficient domain knowledge or covariate information is usually required for content-based filtering to discriminate items the user likes from items the user does not like.

For collaborative filtering, the requirement for covariate information is relaxed by modeling it as latent factors which are to be inferred from existing ratings (Melville et al., 2002; Srebro et al., 2005; Cacheda et al., 2011; Bobadilla et al., 2011; Ekstrand et al., 2011). Some of the most popular collaborative filtering approaches include restricted Boltzmann machines (Salakhutdinov et al., 2007) and various matrix factorization/completion methods: restricted singular value decomposition (Koren et al., 2009), the soft-impute matrix completion method (Mazumder et al., 2010), and the regression-based latent factor models (Agarwal and Chen, 2009a). Practically, the performance of all these methods is comparable in most applications. However, they still cannot solve the "cold-start" problem effectively for users and the items simultaneously (Park et al., 2006; Goldberg et al., 2000; Nguyen et al., 2007; Melville et al., 2002).

In this work, we propose a logistic-regression based collective filtering method that consistently unifies content-based filtering and collaborative filtering. Previously, for existing collaborative filtering methods, the covariate information of the users and items is either not easy to incorporate, or is used solely as an "off-set" to the ratings (Agarwal and Chen, 2009a; Cron et al., 2014), or to group users and the items (Bi et al., 2016). In our method, both the covariate information and the latent factors of the users and the items are uniformly incorporated in the model – they are combined together and contribute equally to the preference of a user giving or an item receiving a specific rating via a multinomial logistic method. This promises the proposed method a better performance in solving the cold-start problem.

In addition, in our method, we use latent factors to generate the *probability mass functions* of the ratings. This is rather different from the matrix factorization/completion based collaborative filtering methods (Koren et al., 2009; Mazumder et al., 2010; Zhu et al., 2016;

Agarwal and Chen, 2009a; Bi et al., 2016; Stern et al., 2009), where the latent factors are used to generate the values of the ratings directly.

Admittedly, the matrix-factorization/completion-methods often boast nice theoretical guarantees on the optimality of the solutions in the sense of minimizing the root mean square error of the predicted ratings with respect to the true ratings when the ratings are generated by the proposed models; however, most of these results are only valid when the ratings take values in real numbers. In many applications, the ratings are categorical (Oghina et al., 2012; Miller et al., 2003; Harper and Konstan, 2015). In particular, the ratings may be binary in certain circumstances – for example, on a photo or video-sharing website like Youtube or Instagram, a photo or video is either liked or disliked by a user. In these cases, our logistic-regression-based collective filtering method is a more natural and suitable choice than the matrix-factorization/completion-methods, especially when the ratings are binary.

Compared to the restricted Boltzmann machine (Salakhutdinov et al., 2007) that takes a similar approach of treating possible ratings as mutually exclusive events whose probability distribution is generated by some hidden variable, our approach has two advantages. First, in our method, the latent factors are associated with a specific user or item instead of being shared within the whole model. Thus they have clearer interpretation and are more capable of distinguishing the different preferences of the users and the items. In addition, it is much more difficult to incorporate covariate information associated to a user or an item in the restricted Boltzmann machine than in our model.

The rest of the paper is organized as follows. In Section 2, we provide the notations and background that will be used in the rest of the paper. In Section 3, we introduce the two-way multinomial logistic model with or without covariate information and give the algorithm to fit the model into a given set of ratings. In Section 4, we implement the proposed method on simulated binary rating data under different missing rates and under different data sizes, simulated multi-categorical rating data under different missing rates and in the cold-start problem, and compare the results to the restricted singular value decomposition (Koren et al., 2009), the soft-impute matrix completion method (Mazumder et al., 2010), the regression-based latent factor models (Agarwal and Chen, 2009a), the restricted Boltzmann machine (Salakhutdinov et al., 2007), and the group-specific recommender system (Bi et al., 2016). In Section 5, we implement the proposed method on MovieLens data and compared the results to the same existing methods. Finally, we conclude this work in Section 6.

2 Notations and Background

In the rest of the paper, we denote the set of real numbers and natural numbers by $\mathbb{N}$ and $\mathbb{R}$ respectively. For $n \in \mathbb{N}$, let $[n] = \{1, \dots, n\}$. Let $|\cdot|$ be the cardinality of a set. The indicator function is denoted by $\mathbf{1}_{\{\cdot\}}$. Let e_n be the n^{th} natural basis in the Euclidean space, namely, all the entries in e_n are 0 except the n^{th} entry being 1.

Consider m users rating n items with possible ratings from $\{1, \dots, r\}$. The records, which may not be complete, can be represented by an $m \times n$ utility matrix $\mathbf{R}$, whose (i, j) entry $\mathbf{R}_{ij}$ gives the score of item j by user i, if it exists; and 0, otherwise. Let $\Omega = \{(i, j) \in [m] \times [n] \mid \mathbf{R}_{ij} \neq 0\}$ be the set of entries where the rating exists, $\Omega_{i\cdot} = \{j \in [n] \mid \mathbf{R}_{ij} \neq 0\}$ be the set of items rated by user i, and $\Omega_{\cdot j} = \{i \in [m] \mid \mathbf{R}_{ij} \neq 0\}$ be the set of users rating item j. By slightly modifying the notation, for $R \in \mathbb{R}^{m \times n}$, let $\Omega(R)$ be

$$\Omega(\mathbf{R})_{ij} = \begin{cases} \mathbf{R}_{ij}, & \text{if } (i, j) \in \Omega \\ 0, & \text{otherwise.} \end{cases} .$$

The matrix factorization/completion based collaborative filtering methods (Koren et al., 2009; Mazumder et al., 2010; Zhu et al., 2016; Agarwal and Chen, 2009a; Bi et al., 2016) are based on decomposing the utility matrix $\mathbf{R}$ approximately as

$$\Omega(\mathbf{R}) \approx \Omega(\mathbf{P}\mathbf{Q}^{\mathrm{T}}), \tag{1}$$

where $\mathbf{P}$ is an $m \times l$ user preference matrix, $\mathbf{Q}$ an $n \times l$ item preference matrix, and l is the given number of latent factors. Different proximity criteria lead to different methods. For example, (Zhu et al., 2016) suggests ℓ_0 and ℓ_1; and (Koren, 2010; Mazumder et al., 2010) use the residuals after a base line fit, similar to ANOVA. To estimate $\mathbf{P}$ and $\mathbf{Q}$ numerically, it is common to use alternating least square methods or the gradient descent methods (Koren et al., 2009). Finally, the predicted value of $\mathbf{R}_{ij}$ for $(i, j) \in \Omega$ is given by $\mathbf{P}_{ij}$.

On the other hand, the restricted Boltzmann machine method (Salakhutdinov et al., 2007) is based on the belief that the rating of a user i on movie j is determined by F binary or Gaussian hidden units by

$$\mathbb{P}\left[\mathbf{R}_{ij} = k\right] \propto e^{\mathbf{b}_{ij} + \sum_{f=1}^{F} h_f \mathbf{W}_{ijf}}, \tag{2}$$

for each possible rating $k = 1, \ldots, r$. Given a partial observation of the ratings, the parameters $\mathbf{b}_{jk}$, $\mathbf{W}_{ifk}$ and the posterior distribution of the value of the hidden units h_f can be estimated by MCMC or "Contrastive Divergence" (Salakhutdinov et al., 2007). Then the probability distribution of the missing ratings can be derived using 2 again, and the expectation can be taken as the predicted ratings.

3 Method

3.1 The Proposed Model

In this model, each user i has latent factors $\mathbf{u}_i = (\mathbf{u}_{i1}, \ldots, \mathbf{u}_{ir})$, and each item j has latent factors $\mathbf{v}_j = (\mathbf{v}_{j1}, \ldots, \mathbf{v}_{jr})$, where $\mathbf{u}_{i1}, \ldots, \mathbf{u}_{ir}, \mathbf{v}_{j1}, \ldots, \mathbf{v}_{jr} \in \mathbb{R}^l$. In addition, there are latent factors $\mathbf{c}_k$ and $\mathbf{d}_k$ that describe the overall preferences of the users and the items, respectively. We note that each $\mathbf{u}_{ik}$, $\mathbf{v}_{jk}$, $\mathbf{c}_k$ and $\mathbf{d}_k$ are vectors of the same dimension l. The rating of item j by user i is assumed to follow the probability mass function

$$\mathbb{P}\left[\mathbf{R}_{ij} = k\right] = \frac{e^{(\mathbf{u}_{ik}+\mathbf{c}_k)(\mathbf{v}_{jk}+\mathbf{d}_k)}}{\sum_{k=1}^{r} e^{(\mathbf{u}_{ik}+\mathbf{c}_k)(\mathbf{v}_{jk}+\mathbf{d}_k)}}. \tag{3}$$

For user i with latent factors $(\mathbf{u}_{i1} + c_1, \ldots, \mathbf{u}_{ir} + c_r)$, the probability that it rates k on the item j is only proportional to the exponential of the k-th coefficient of the item; and the same for the items.

Therefore, this model can be viewed as a two-way multinomial logistic model. The model (3) is similar to the restricted Boltzmann machine (2) in the sense that the possible ratings are treated as mutually exclusive events, whose probability distribution is determined by the latent factors. Therefore, it is a more natural and suitable choice than the matrix-factorization/completion-methods for categorical ratings.

When the covariate information of the users and the items are available, we can incorporate them into the model by augmentation. If each user i has m_{CI} covariate information $(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{m_{CI}})$, and each item j has n_{CI} covariate information $(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{n_{CI}})$, then we can augment the latent factors by

$$\tilde{\mathbf{u}}_{ik} = (\mathbf{u}_{ik}, \boldsymbol{\alpha}_{i_{CI}}, \mathbf{b}_{i_{CI}k}), \quad \tilde{\mathbf{v}}_{ik} = (\mathbf{v}_{ik}, \mathbf{a}_{i_{CI}k}, \boldsymbol{\beta}_{i_{CI}}), \tag{4}$$

where $\mathbf{a}_{i_{CI}}, \mathbf{b}_{j_{CI}}$ are the latent factors of proper dimensions associated with the covariate information. In this case, we have

$$p_{ijk} = \mathbb{P}\left[\mathbf{R}_{ij} = k\right] = \frac{e^{(\tilde{\mathbf{u}}_{ik}+\mathbf{c}_k)(\tilde{\mathbf{v}}_{jk}+\mathbf{d}_k)}}{\sum_{k=1}^{r} e^{(\tilde{\mathbf{u}}_{ik}+\mathbf{c}_k)(\tilde{\mathbf{v}}_{jk}+\mathbf{d}_k)}}, \quad k \in [r]. \tag{5}$$

Since in this model the latent factors are associated with a specific user or item instead of being shared within the whole model, thus they are more capable of telling the different preferences of the users and the items. More importantly, this makes it possible to uniformly incorporate covariate information, unlike the restricted Boltzmann machine.

Given the augmented latent factors $\tilde{\mathbf{u}}_{ik}$, $\tilde{\mathbf{v}}_{jk}$, $\mathbf{c}_k$ and $\mathbf{d}_k$, the predicted ratings are generated depending on the evaluation criteria (Gunawardana and Shani, 2015). Similar to previous works (Bi et al., 2016; Agarwal and Chen, 2009b; Salakhutdinov et al., 2007; Mazumder et al., 2010), we use the root mean square error (RMSE) to evaluate the predictions and consider the following objective function

$$\mathcal{S} = \sum_{(i,j)\in\Omega} (\bar{\mathbf{R}}_{ij} - \mathbf{R}_{ij})^2, \tag{6}$$

where the predicted ratings are the expectation of all possible ratings

$$\bar{\mathbf{R}}_{ij} = \sum_{k=1}^{r} k p_{ijk}. \tag{7}$$

Compared to the maximum likelihood estimation (MLE) approach, the RMSE approach is more robust to irregular data, such as in the examples where binary covariates can completely separate binary responses. In addition, the RMSE approach is more computationally effective in handling missing data.

3.2 Estimating the Latent Factors

Given a record of ratings $\mathbf{R}_{ij}$ for $(i,j) \in \Omega$, the latent factors $\mathbf{u}_{ik}$, $\mathbf{v}_{jk}$, $\mathbf{c}_k$ and $\mathbf{d}_k$ can be estimated by minimizing the mis-predication rate or the root mean square error by applying the gradient descent method iteratively on the latent factors. The gradients of the objective function (7) with respect to the latent factors $\mathbf{u}_{ik}$, $\mathbf{v}_{jk}$, $\mathbf{c}_k$, and $\mathbf{d}_k$ are

$$\begin{aligned} \frac{\partial \mathcal{S}}{\partial \tilde{\mathbf{u}}_{ik}} &= \sum_{j \in \Omega_{i\cdot}} P_{ijk}(\tilde{\mathbf{v}}_{jk} + \mathbf{d}_k), \quad & \frac{\partial \mathcal{S}}{\partial \tilde{\mathbf{v}}_{jk}} &= \sum_{i \in \Omega_{\cdot j}} P_{ijk}(\tilde{\mathbf{u}}_{ik} + \mathbf{c}_k), \\ \frac{\partial \mathcal{S}}{\partial \mathbf{c}_k} &= \sum_{(i,j) \in \Omega} P_{ijk}(\tilde{\mathbf{v}}_{jk} + \mathbf{d}_k), \quad & \frac{\partial \mathcal{S}}{\partial \mathbf{d}_k} &= \sum_{(i,j) \in \Omega} P_{ijk}(\tilde{\mathbf{u}}_{ik} + \mathbf{c}_k), \end{aligned} \tag{8}$$

where

$$P_{ijk} = 2(\bar{\mathbf{R}}_{ij} - \mathbf{R}_{ij})(k - \bar{\mathbf{R}}_{ij})p_{ijk}. \tag{9}$$

To implement the gradient descent, we set the initial values for the latent factors as

$$\begin{aligned} &\mathbf{u}_{ik}, \mathbf{v}_{ik} \sim N(0, \eta I_D), \quad \mathbf{a}_{i_{CI}k}, \mathbf{b}_{j_{CI}k} = 0, \\ &\mathbf{c}_k = \mathbf{d}_k = \frac{1}{2} \log \left(\sum_{(i,j) \in \Omega} \mathbf{1}_{\{\mathbf{R}_{ij} = k\}} / l \right), \quad k \in [r]. \end{aligned} \tag{10}$$

where l is the dimension of each $\mathbf{u}_{ik}$, $\mathbf{v}_{jk}$, $\mathbf{c}_k$ and $\mathbf{d}_k$, and $\eta \ll \min \mathbf{c}_k$. In the simulations, we choose $\eta = \min \mathbf{c}_k / 10$. The choice of $\mathbf{c}_k$ and $\mathbf{d}_k$ is to ensure that when $\mathbf{u}_{ik}$ and $\mathbf{v}_{jk}$ are small, namely, the users and items have no strong preference, and the probability mass function of the predicted rating is approximated by the empirical distribution of all the existing ratings.

3.3 Implementation Issues

We use the stochastic gradient descent method with decreasing step sizes to fit the model with existing data (Spall, 2005). The purpose is to ensure accurate small-step searching near the local minimum, while allowing for occasional big steps for a faster convergence. The step sizes of the gradient descent process for the latent factors $\mathbf{u}_{ik}$, $\mathbf{v}_{jk}$, $\mathbf{c}_k$ and $\mathbf{d}_k$ are randomly generated respectively by

$$\begin{aligned}
\delta_{\mathbf{u}_{ik}} &= \Delta_K(\partial_{\mathbf{u}_{ik}}\mathcal{S}(t)), & \delta_{\mathbf{v}_{jk}} &= \Delta_K(\partial_{\mathbf{v}_{jk}}\mathcal{S}(t)), \\
\delta_{\mathbf{c}_k} &= \Delta_K(\partial_{\mathbf{c}_k}\mathcal{S}(t)), & \delta_{\mathbf{d}_k} &= \Delta_K(\partial_{\mathbf{d}_k}\mathcal{S}(t)), \\
\delta_{\mathbf{a}_{i_{CI}k}} &= \Delta_K(\partial_{\mathbf{a}_{i_{CI}k}}\mathcal{S}(t)), & \delta_{\mathbf{b}_{j_{CI}k}} &= \Delta_K(\partial_{\mathbf{b}_{j_{CI}k}}\mathcal{S}(t)),
\end{aligned} \tag{11}$$

where the $\Delta_K(x)$ are sampled from the exponential distribution $\text{EXP}(\max\{x, K\})$ independently with a rate of $\max\{x, K\}$ and K controlling the step size. Consequently, the latent factors update by

$$\begin{aligned}
\mathbf{u}_{ik} &\leftarrow \mathbf{u}_{ik} - \delta_{\mathbf{u}_{ik}}\partial_{\mathbf{u}_{ik}}\mathcal{S}(t), & \mathbf{v}_{jk} &\leftarrow \mathbf{v}_{jk} - \delta_{\mathbf{v}_{jk}}\partial_{\mathbf{v}_{jk}}\mathcal{S}(t), \\
\mathbf{c}_k &\leftarrow \mathbf{c}_k - \delta_{\mathbf{c}_k}\partial_{\mathbf{c}_k}\mathcal{S}(t), & \mathbf{d}_k &\leftarrow \mathbf{d}_k - \delta_{\mathbf{d}_k}\partial_{\mathbf{d}_k}\mathcal{S}(t), \\
\mathbf{a}_{i_{CI}k} &\leftarrow \mathbf{a}_{i_{CI}k} - \delta_{\mathbf{a}_{i_{CI}k}}\partial_{\mathbf{a}_{i_{CI}k}}\mathcal{S}(t), & \mathbf{b}_{j_{CI}k} &\leftarrow \mathbf{b}_{j_{CI}k} - \delta_{\mathbf{b}_{j_{CI}k}}\partial_{\mathbf{b}_{j_{CI}k}}\mathcal{S}(t).
\end{aligned} \tag{12}$$

The convergence in the gradient descent is checked by comparing the relative change in the objective function within T steps. Specifically, if the objective function changes less than ϵ, relatively, in T steps, then we stop the algorithm. The above statements are concluded in Algorithm 1.

4 Simulation

In this section, we implement the two-way multinomial logistic model with or without covariate information and compare it to other existing collaborative filtering methods, namely the restricted singular value decomposition (Koren et al., 2009), the soft-impute matrix completion method (Mazumder et al., 2010), the regression-based latent factor models (Agarwal and Chen, 2009a), the restricted Boltzmann machine (Salakhutdinov et al., 2007), and the group-specific recommender system (Bi et al., 2016) on the simulated binary rating data in Section 4.1 and on the simulated multi-categorical rating data in Section 4.2 to study the performance of our method under different missing rates and in the cold-start problem.

For the restricted singular value decomposition (Koren et al., 2009), we use the R package. For the soft-impute matrix completion method (Mazumder et al., 2010), we apply the R package "softImpute". For the regression-based latent factor models (Agarwal and Chen,

Algorithm 1 Infer the latent factors by minimizing the mis-prediction rate or the root mean square error

```
Input: rating matrix $\mathbf{R}$, convergence threshold $T$,$\epsilon$, parameter $K$, $t = 1$
Initialize the latent factors $\tilde{\mathbf{u}}_{ik}, \tilde{\mathbf{v}}_{jk}, \mathbf{c}_k, \mathbf{d}_k$ by (10)
Compute the value of the objective function $\mathcal{S}(t)$ by (7)
repeat
    Compute the new latent factors by gradient descent by (12)
    Compute the new value of the objective function $\mathcal{S}'$ by (7)
    if $\mathcal{S}' < \mathcal{S}(t)$ then
        Update the latent factors $\tilde{\mathbf{u}}_{ik}, \tilde{\mathbf{v}}_{jk}, \mathbf{c}_k, \mathbf{d}_k$ and set $\mathcal{S}(t+1) = \mathcal{S}'$
    else
        Set $\mathcal{S}(t+1) = \mathcal{S}(t)$ and continue
    end if
    $t = t + 1$
until $t > T$ and $\frac{|\mathcal{S}(t) - \mathcal{S}(t-T)|}{\mathcal{S}(t-T)} < \epsilon$.
```

2009a), we use the default 10 iterations. For the group-specific recommender system (Bi et al., 2016), we use the Matlab code available at `https://sites.google.com/site/xuanbigts/software`. For the restricted Boltzmann machine (Salakhutdinov et al., 2007), we use the python package TensorFlow available at `https://github.com/tensorflow/tensorflow`. The two-way multinomial logistic model with or without covariate information is implemented in Matlab available at `https://bitbucket.org/yuwang0531/tml`.

4.1 Binary Classification

In this section, we study the performance of the proposed method on the simulated binary rating data. This is the kind of problem that the matrix factorization/completion based methods are not specialized for. The simulated binary rating data is generated by the two-way multinomial logistic model with covariate information introduced in Section 3 with the number of category $r = 2$. Due to similarity in the performance of the competing models in fitting real data (the best one is only 10% better than the worst one), and the relatively high missing rate, the trends of the results on simulated data are the same no matter which model is used to generate the ratings. (We have performed an additional simulation with 5-category ratings generated by 1500 users and 1500 items under the RSVD model in the Supplementary Material. The trends are similar to the results derived from the simulated data generated by the proposed model.) Using the simulated data, we show the advantage of our method compared to the other methods under different missing rates in Section 4.1.1 and under different data sizes in Section 4.1.2.

4.1.1 Missing Rate

To study the influence of the missing rate, we generate the ratings for 300 users and 300 items under the missing rates 0.7, 0.8, 0.9, 0.95. The dimension of latent factors is $l = 3$. The numbers of covariate information for both the items and the users are $m_{CI} = 2$ and $n_{CI} = 2$ respectively. The latent factors $\mathbf{u}_{ik}, \mathbf{v}_{jk}, \mathbf{c}_k, \mathbf{d}_k$ are independently generated from the Gaussian distribution $N(0, 0.1)$. The covariate information for the users $\boldsymbol{\alpha}_{i_{CI}}$ and the times $\boldsymbol{\beta}_{i_{CI}}$ is independently generated from the Bernoulli distribution $B(0.5)$ in $\{0, 1\}$. The corresponding latent factors $\mathbf{a}_{i_{CI}k}$ and $\mathbf{b}_{i_{CI}k}$ are independently generated from the Gaussian distribution $N(0, 0.1)$. To generate the ratings at the desired missing rate, we first generate all possible $m \times n$ ratings between the users and the items and then pick out a given portion of the ratings. Also, all the ratings are randomly permuted to simulate random log-ins of users in real applications and divided by 60% for training, 15% for tuning and 25% for testing, to mimic the setting of MovieLens data (see Section 5).

For the regularized singular value decomposition method (Koren et al., 2009), we set the tuning parameter $\lambda = 3$. For the regression-based latent factor model (Agarwal and Chen, 2009a), we use the default of 10 iterations. For the soft-impute method (Mazumder et al., 2010), we choose the default $\lambda = 0, K = 4$ to achieve convergence for the local minimum. For the group-specific method (Bi et al., 2016), we choose $K = 3$. For the restricted Boltzmann machine (Salakhutdinov et al., 2007), 100 hidden units are used. The two-way multinomial logistic model with or without covariate information is implemented with dimension of latent factors $l = 3$ for the best trade-off between accuracy and computational cost.

The simulation results for the above methods are shown in Table 1, where the mean is calculated from 5000 runs, and the standard error of the mean is less than 0.002. It shows that the advantage of the two-way logistic model with covariate information is generally more significant under high missing rate. For example, under the missing rate of 95%, the root mean square error of the two-way logistic model with covariate information is 4.3% smaller than the same model without covariate information, 10.5% smaller than that of the regularized singular value decomposition method (Koren et al., 2009), 7.4% smaller than the restricted Boltzmann machine (Salakhutdinov et al., 2007), 7.0% smaller than the regression-based latent factor model (Agarwal and Chen, 2009a), 6.0% smaller than the soft-impute method (Mazumder et al., 2010), and 5.5% smaller than the group-specific recommender system (Bi et al., 2016). This indicates that the covariate information is more important under high missing rate and the two-way logistic model utilizes the covariate information more effectively.

4.1.2 Data Size

To study the influence of the data size, we generate ratings for 300 users and 300 items, and for 1500 users and 1500 items, under the missing rates 0.8, 0.95 from the model described in Section 3. The dimension of latent factors is $l = 3$. The numbers of covariate information for both the items and the users are $m_{CI} = 2$ and $n_{CI} = 2$, respectively. The latent factors for the users and items and ratings in the training, tuning and testing sets are generated in the same way as in Section 4.1.1. The parameter settings of the regularized singular value decomposition method (Koren et al., 2009), the regression-based latent factor model (Agarwal and Chen, 2009a), the soft-impute method (Mazumder et al., 2010), the group-specific method (Bi et al., 2016), the restricted Boltzmann machine (Salakhutdinov et al., 2007), and two-way multinomial logistic model with or without covariate information are also the same as given in Section 4.1.1.

The simulation results for the above methods are shown in Table 2, where the mean is calculated from 5000 runs, and the standard error of the mean is less than 0.002. It shows that the advantage of the two-way logistic model with covariate information is consistent for different data sizes. For all the methods, the performance is better on the larger data than in the smaller data. This is because under the same missing rate, the number of ratings for each user and each item increases as the data size increases, thus it is easier to make predictions on the larger data size.

Table 2 also shows that the two-way logistic model with covariate information is less sensitive to data size change than other methods. Namely, this method is better at handling smaller data than the other methods. For example, under a missing rate of 95%, the increase in the root mean square error after changing the data size from 1500×1500 to 300×300 for the two-way logistic model with covariate information is 3.80%, while the increments are $6.95\%, 5.51\%, 5.84\%, 5.84\%, 6.79\%, 6.27\%$ for the regularized singular value decomposition method, the restricted Boltzmann machine, the regression-based latent factor model, the soft-impute method, the group-specific recommender system, and the two-way logistic model without covariate information, respectively.

4.2 Multi-categorical Recommendation with Cold-Start

In this section, we study the performance of the proposed method on multi-categorical data. This is the most common setup for recommender systems in real applications. The simulated multi-categorical rating data is generated by the two-way multinomial logistic model with covariate information introduced in Section 3, with five categories $\{1, 2, 3, 4, 5\}$ in the same fashion as the MovieLens data. Using simulated data, we show the advantage of our method compared to the other existing methods, under different missing rates again in Section 4.2.1, and in the cold-start problem 4.2.2.

4.2.1 Missing Rate

We generate ratings for 1500 users and 1500 items under the missing rates 0.7, 0.8, 0.9, 0.95. The dimension of latent factors is $l = 2$. The numbers of covariates for both the items and the users are $m_{CI} = 1$ and $n_{CI} = 1$ respectively. The latent factors $\mathbf{u}_{ik}, \mathbf{v}_{jk}, \mathbf{c}_k, \mathbf{d}_k$ are independently generated from the Gaussian distribution $N(0, 0.1)$. The covariate information for the users $\boldsymbol{\alpha}_{i_{CI}}$ and the times $\boldsymbol{\beta}_{i_{CI}}$ is independently generated from the Bernoulli distribution $B(0.5)$ in $\{0, 1\}$. The corresponding latent factors $\mathbf{a}_{i_{CI}k}$ and $\mathbf{b}_{i_{CI}k}$ are independently generated from the Gaussian distribution $N(0, 0.1)$. To generate the ratings at the desired missing rate, we first generate all possible $m \times n$ ratings between the users and the items and then pick out a given portion of the ratings. All the ratings are randomly permuted to simulate random log-ins of the users in the real case, and divided by 60% for training, 15% for tuning and 25% for testing.

The parameters for the competing methods are tuned for their best performance, as follows. Specifically, we set $K = 4$ and $\lambda = 6$ for the regularized singular value decomposition method, choose $\lambda = 0$ and 10 iterations for the regression-based latent factor model, , select $K = 4$ for the soft-impute method, and choose 100 hidden units for the restricted Boltzmann machine. The two-way multinomial logistic model with or without covariate information is implemented with dimension of latent factors $l = 2$ for the objective function (6).

The simulation results for the above methods are shown in Table 3, where the mean is calculated from 10000 runs, and the standard error of the mean is less than 0.002. Similar to the trend shown in Table 1, the advantage of the two-way logistic model with covariate information is generally more significant under high missing rates. For example, under the missing rate of 95%, the root mean square error of the two-way logistic model with covariate information is 2.3% smaller than the same model without covariate information, 12.9% smaller than that of the regularized singular value decomposition method, 11.5% smaller than the restricted Boltzmann machine, 14.8% the regression-based latent factor model, 11.0% smaller than the soft-impute method, and 1.8% smaller than the group-specific recommender system.

4.2.2 Cold-Start Problem

In this simulation study, we consider the “cold-start” problem where the testing set contains a large portion of new users and new items that have not appeared in the training set. We generate ratings for 1500 users and 1500 items under a missing rate of 0.99. The dimension of latent factors is $l = 2$. The numbers of covariates for both the items and the users are $m_{CI} = 1$ and $n_{CI} = 1$ respectively. The latent factors for the users and the items are generated in the same way as in Section 4.2.1. The parameter settings of the regularized singular value decomposition method, the regression-based latent factor model, the soft-impute method, the group-specific method, the restricted Boltzmann machine, and two-way multinomial logistic model with or without covariant are also the same as given in Section 4.2.1.

The ratings are generated at a missing rate of 0.99 and randomly permuted to simulate random log-ins of the users in the real case. They are divided by 75% for training, 10% for tuning and 15% for testing. To create a cold-start setup, we then exchange the ratings in the testing set with the ratings in the training and tuning sets, such that 50% of the ratings in the testing set are either from new users or for new items.

The simulation results for the above methods are shown in Table 4, where the mean is calculated from 10000 runs, and the standard error of the mean is less than 0.002. It shows that the two-way logistic model with or without covariate information outperforms the other methods in the cold-start problem. Specifically, on the whole testing set, the root mean square error of the two-way logistic model with covariate information is 1.3% smaller than the same model without covariate information, 9.8% smaller than that of the regularized singular value decomposition method, 9.3% smaller than the restricted Boltzmann machine, 12.0% the regression-based latent factor model, 8.8% smaller than the soft-impute method, and 2.2% smaller than the group-specific recommender system. For the new ratings, the root mean square error of the two-way logistic model with covariate information is 1.7% smaller than the same model without covariate information, 11.1% smaller than that of the regularized singular value decomposition method, 12.4% smaller than the restricted Boltzmann machine, 14.4% the regression-based latent factor model, 9.7% smaller than the soft-impute method, and 2.3% smaller than the group-specific recommender system.

5 MovieLens Data

We apply the proposed method to the MovieLens 1M and 10M data (Miller et al., 2003; Harper and Konstan, 2015), collected by GroupLens Research (`http://grouplens.org/datasets/movielens`). The MovieLens 1M data contains $1,000,209$ integer ratings of $3,883$ movies by $6,040$ users ranging from $\{1, 2, \ldots, 5\}$, at a missing rate of 96%. In addition, it provides demographic information including the age, gender, occupation, and zip code of the users, and the genres of the movies. In the MovieLens 10M data, $10,000,054$

ratings are collected from $71,567$ users over $10,681$ items, at a missing rate of 99%. For the 10M data, the user's demographic information is not available.

The categorical covariate information of the users and the items is binary-coded. To avoid using too many variables, we divide users' ages by $0-9$, $10-19$, ..., $60-69$ into 7 groups, considering that users of similar ages tend to have similar preference for movies. The users' zip codes are divided into 10 groups by $0-9,999$, ..., $90,000-99,999$, since users of similar zip codes tend to live closer geographically. The users' gender is either male or female, hence encoded by $\{0,1\}$. In addition, each user has one of 21 possible occupations. Consequently, the users' covariate information is encoded into 39 binary variables. Each item is labeled by a subset of 18 possible genres; therefore, the items' covariate information is encoded by 18 binary variables.

The simulation results for the proposed methods and the other existing methods are shown in Table 5. The data is divided by 60% for training, 15% for tuning and 25% for testing, sorted by timestamps. For the proposed two-way multinomial logistic model with or without covariate information, we apply the loss function (7) and set the parameter $l = 5$ to achieve the best performance with the smallest value for l, after searching over $l = 1, 2, \dots, 10$. The parameters for competing models are tuned for their best performance as follows. For the RSVD, we use $K = 4$ and $\lambda = 8$ for the 1M data, and $K = 4$ and $\lambda = 6$ for the 10M data. For the regression-based latent factor model, we let $K = 1$ for both the 1M and 10M data and take 25 and 10 iterations, respectively. For the soft-impute method, we select $\lambda = 0, K = 4$ and $\lambda = 0, K = 9$ for the 1M and 10M data, respectively. For the restricted Boltzmann machine, we choose 200 hidden units for both the 1M and 10M data.

Compared to the collaborative filtering methods based on matrix factorization/completion, the root mean square error of the two-way logistic model with covariate information is 10.1% smaller than that of the regularized singular value decomposition method, 20.8% smaller than the regression-based latent factor model 11.6% smaller than the soft-impute method, and 1.6% smaller than the group-specific recommender system. On the MovieLens 10M data, the root mean square error of the two-way logistic model with covariate information is 7.6% smaller than that of the regularized singular value decomposition method, 5.4% smaller than the regression-based latent factor model, 9.5% smaller than the soft-impute method, and 0.9% smaller than the group-specific recommender system.

Due to the loss of covariate information, the two-way logistic model without covariate information performs 2.2% worse on the MovieLens 1M data and 0.4% worse on the MovieLens 10M data than the same model with covariate information. Compared to the group-specific recommender system (Bi et al., 2016) that also utilizes covariate information to group up the users and the items, the two-way logistic model without covariate information perform 0.5% worse on the MovieLens 1M data, but 0.4% better on the MovieLens 10M data. As for the other matrix-factorization-based collaborative filtering methods, the root mean square error of the two-way logistic model without covariate information is 8.2% smaller than that of the regularized singular value decomposition method, 19.1% smaller than the regression-based latent factor model, and 9.7% smaller than the soft-impute method on the MovieLens 1M data. On the MovieLens 10M data, it is 7.3% smaller than the regularized singular value decomposition method, 5.1% smaller than the regression-based latent factor model, and 9.2% smaller than the soft-impute method. This shows that the two-way logistic model is more consistent with the categorical nature of the ratings.

One the other hand, compared to the restricted Boltzmann method that also generates categorical predicted ratings, the two-way multinomial logistic model with covariate information also performs 6.3% better on the MovieLens 1M data and 5.8% better on the MovieLens 10M data than the restricted Boltzmann machine. Even without the covariate information, the two-way multinomial logistic model without covariate information also performs 4.3% better on the MovieLens 1M data and 5.4% better on the MovieLens 10M data than the restricted Boltzmann machine.

From the above comparison, we observe that the improvement the two-way logistic model with covariate information over the same model without covariate information is more significant for the MovieLens 1M data than the MovieLens 10M data. This is consistent with the fact that the MovieLens 1M data contains more covariate information than the MovieLens 10M data. In addition, the improvement of both the two-way logistic model with and without covariate information is generally more significant in the MovieLens 1M data than in the MovieLens 10M data. This is because in both the MovieLens 1M and 10M data, each user rates about 150 movies on average, while the average number of ratings for the movies is 250 for the 1M data and 900 for the 10M data. The availability of more past ratings makes predicting the new ratings easier on the MovieLens 10M data than the 1M data.

6 Conclusion

In this paper, we propose a two-way multinomial logistic model for recommender systems on categorical ratings. Specifically, instead of factorizing the ratings immediately into the product of the latent factors of the users and the items, we use a multi-class logistic model where the probability mass function of the ratings of a user on an item is factorized into a function of the latent factor of that user and item. Each user and each item have latent factors representing their a priori absolute preference on giving/receiving a specific rating, and the actual ratings are generated from a probability distribution of the ratings determined by the user's and item's latent factors together with their covariate information through a logistic function.

Compared to the existing collaborative filtering methods, the proposed method can incorporate both the covariate information and the latent factors of the users and the items uniformly in the model. Compared to the matrix factorization/completion based collaborative filtering methods, where the ratings are treated as the discretization of the inner products of the latent factors from the users and the items, we treat the possible ratings as categorical random variables where the probability mass function is determined by the latent factors and the covariate information of the users and the items in the proposed method.

These give the proposed method an advantage in handling categorical ratings and the cold-start problems. As shown by the numerical experiments, the proposed method performs significantly better than the restricted singular value decomposition (Koren et al., 2009), the soft-impute matrix completion method (Mazumder et al., 2010), the regression-based latent factor models (Agarwal and Chen, 2009a), the restricted Boltzmann machine (Salakhutdinov et al., 2007), and the group-specific recommender system (Bi et al., 2016) on the simulated binary rating data under different missing rates and under different data sizes, the simulated multi-categorical rating data under different missing rates and in the cold-start problem, and the MovieLens 1M and 10M data.

Acknowledgement

We greatly appreciate the comments and suggestions provided by the three anonymous reviewers, which improve the manuscript significantly. We would like to acknowledge support for this project from the National Science Foundation grants DMS-1415308, DMS-1613190, DMS-1821198, CPS-1329991 and AFOSR Grant FA9550-15-1-0059.

References

Aciar, S., D. Zhang, S. Simoff, and J. Debenham (2006). Recommender system based on consumer product reviews. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '06, Washington, DC, USA, pp. 719–723. IEEE Computer Society.

Adomavicius, G. and A. Tuzhilin (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering 17*(6), 734–749.

Agarwal, D. and B.-C. Chen (2009a). Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, New York, NY, USA, pp. 19–28. ACM.

Agarwal, D. and B.-C. Chen (2009b). Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, New York, NY, USA, pp. 19–28. ACM.

Aggarwal, C. C. (2016). *Recommender systems.* Springer.

Agresti, A. and M. Kateri (2011). Categorical data analysis. In M. Lovric (Ed.), *International Encyclopedia of Statistical Science*, pp. 206–208. Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-04898-2_161.

Bell, R. M. and Y. Koren (2007). Lessons from the Netflix prize challenge. *SIGKDD Explor. Newsl. 9*(2), 75–79.

Bi, X., A. Qu, J. Wang, and X. Shen (2016). A group-specific recommender system. *Journal of the American Statistical Association*.

Blanco-Fernandez, Y., J. J. Pazos-Arias, A. Gil-Solla, M. Ramos-Cabrer, and M. Lopez-Nores (2008). Providing entertainment by content-based filtering and semantic reasoning in intelligent recommender systems. *IEEE Transactions on Consumer Electronics 54*(2).

Bobadilla, J., F. Ortega, A. Hernando, and J. Alcalá (2011). Improving collaborative filtering recommender system results and performance using genetic algorithms. *Knowledge-Based Systems 24*(8), 1310–1316.

Cacheda, F., V. Carneiro, D. Fernández, and V. Formoso (2011). Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Trans. Web 5*(1), 2:1–2:33.

Cron, A., L. Zhang, and D. Agarwal (2014). Collaborative filtering for massive multinomial data. *Journal of Applied Statistics 41*(4), 701–715.

Davidson, J., B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, and D. Sampath (2010). The YouTube video recommendation system. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, New York, NY, USA, pp. 293–296.

Ekstrand, M. D., J. T. Riedl, and J. A. Konstan (2011). Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction 4*(2), 81–173.

Ghosh, S., M. Mundhe, K. Hernandez, and S. Sen (1999). Voting for movies: the anatomy of a recommender System. In *Proceedings of the Third Annual Conference on Autonomous Agents*, AGENTS '99, New York, NY, USA, pp. 434–435.

Goldberg, K., T. Roeder, D. Gupta, and C. Perkins (2000). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval 4*(2), 133–151.

Gomez-Uribe, C. A. and N. Hunt (2015). The Netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Manage. Inf. Syst. 6*(4), 13:1–13:19.

Gunawardana, A. and G. Shani (2015). Evaluating recommender systems. In *Recommender Systems Handbook*, pp. 265–308. Springer.

Harper, F. M. and J. A. Konstan (2015). The MovieLens datasets: History and context. *ACM Trans. Interact. Intell. Syst. 5*(4), 19:1–19:19.

Iaquinta, L., M. De Gemmis, P. Lops, G. Semeraro, M. Filannino, and P. Molino (2008). Introducing serendipity in a content-based recommender system. In *Hybrid Intelligent Systems, 2008. HIS'08. Eighth International Conference on*, pp. 168–173. IEEE.

Kim, Y. S., B.-J. Yum, J. Song, and S. M. Kim (2005). Development of a recommender system based on navigational and behavioral patterns of customers in e-commerce sites. *Expert Systems with Applications 28*(2), 381–393.

Koenigstein, N., G. Dror, and Y. Koren (2011). Yahoo! Music recommendations: Modeling music ratings with temporal dynamics and item taxonomy. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, New York, NY, USA, pp. 165–172.

Koren, Y. (2010). Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Trans. Knowl. Discov. Data 4*(1), 1:1–1:24.

Koren, Y., R. Bell, and C. Volinsky (2009). Matrix factorization techniques for recommender systems. *Computer 42*(8), 30–37.

Levi, A., O. Mokryn, C. Diot, and N. Taft (2012). Finding a needle in a haystack of reviews: Cold start context-based hotel recommender system. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, New York, NY, USA, pp. 115–122.

Lops, P., M. De Gemmis, and G. Semeraro (2011). Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*, pp. 73–105. Springer.

Lu, J., D. Wu, M. Mao, W. Wang, and G. Zhang (2015). Recommender system application developments: A survey. *Decision Support Systems 74*(Supplement C), 12–32.

Mazumder, R., T. Hastie, and R. Tibshirani (2010). Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research 11*, 2287–2322.

Melville, P., R. J. Mooney, and R. Nagarajan (2002). Content-boosted collaborative filtering for improved recommendations. In *Eighteenth National Conference on Artificial Intelligence*, Menlo Park, CA, USA, pp. 187–192. American Association for Artificial Intelligence.

Miller, B. N., I. Albert, S. K. Lam, J. A. Konstan, and J. Riedl (2003). MovieLens unplugged: Experiences with an occasionally connected recommender system. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI '03, New York, NY, USA, pp. 263–266. ACM.

Mirbakhsh, N. and C. X. Ling (2015). Improving top-N recommendation for cold-start users via cross-domain information. *ACM Trans. Knowl. Discov. Data 9*(4), 33:1–33:19.

Mooney, R. J. and L. Roy (2000). Content-based book recommending using learning for text categorization. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, DL '00, New York, NY, USA, pp. 195–204.

Nguyen, A.-T., N. Denos, and C. Berrut (2007). Improving new user recommendations with rule-based induction on cold user data. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, RecSys '07, New York, NY, USA, pp. 121–128.

Oghina, A., M. Breuss, M. Tsagkias, and M. d. Rijke (2012). Predicting IMDB movie ratings using social media. In *Advances in Information Retrieval*, Lecture Notes in Computer Science, pp. 503–507. Springer, Berlin, Heidelberg.

Park, S.-T., D. Pennock, O. Madani, N. Good, and D. DeCoste (2006). NaiVe filterbots for robust cold-start recommendations. In *Proceedings of the 12th ACM SIGKDD Interna-*

tional Conference on Knowledge Discovery and Data Mining, KDD '06, New York, NY, USA, pp. 699–705.

Resnick, P. and H. R. Varian (1997). Recommender systems. *Commun. ACM 40*(3), 56–58.

Ricci, F., L. Rokach, and B. Shapira (2011). Introduction to Recommender Systems Handbook. In *Recommender Systems Handbook*, pp. 1–35. Springer, Boston, MA.

Salakhutdinov, R., A. Mnih, and G. Hinton (2007). Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, New York, NY, USA, pp. 791–798.

Spall, J. C. (2005). *Introduction to stochastic search and optimization: Estimation, simulation, and control*, Volume 65. John Wiley & Sons.

Srebro, N., N. Alon, and T. S. Jaakkola (2005). Generalization error bounds for collaborative prediction with low-rank matrices. In L. K. Saul, Y. Weiss, and L. Bottou (Eds.), *Advances in Neural Information Processing Systems 17*, pp. 1321–1328. MIT Press.

Stern, D., R. Herbrich, and T. Graepel (2009). Matchbox: Large scale bayesian recommendations. In *Proceedings of the 18th International World Wide Web Conference.*

Zhu, Y., X. Shen, and C. Ye (2016). Personalized prediction and sparsity pursuit in latent factor models. *Journal of the American Statistical Association 111*(513), 241–252.

Table 1: Comparison of the root mean square error of the methods under different missing rates on the simulated binary rating data with 300 users and 300 items. RSVD, RBM, RB, SI, GS, TML, and TMLCI stand for the regularized singular value decomposition method (Koren et al., 2009), the restricted Boltzmann machine (Salakhutdinov et al., 2007), the regression-based latent factor model (Agarwal and Chen, 2009a), the soft-impute method (Mazumder et al., 2010), the group-specific recommender system (Bi et al., 2016), the proposed two-way multinomial logistic model, and the proposed two-way multinomial logistic model with covariate information, respectively.

Missing rate	RSVD	RBM	RB	SI	GS	TML	TMLCI
0.7	0.3784	0.3812	0.3742	0.3735	0.3624	0.3543	0.3484
0.8	0.3984	0.3904	0.3843	0.3793	0.3745	0.3635	0.3584
0.9	0.4064	0.3983	0.3997	0.3913	0.3874	0.3793	0.3643
0.95	0.4184	0.4043	0.4026	0.3984	0.3965	0.3913	0.3745

Table 2: Comparison of the methods under different sample sizes on the simulated binary rating data.

Missing rate	Data Size	RSVD	RBM	RB	SI	GS	TML	TMLCI
0.8	300 × 300	0.3984	0.3904	0.3843	0.3793	0.3745	0.3635	0.3584
	1500 × 1500	0.3774	0.3713	0.3647	0.3573	0.3532	0.3484	0.3454
0.95	300 × 300	0.4184	0.4043	0.4026	0.3984	0.3965	0.3913	0.3795
	1500 × 1500	0.3912	0.3832	0.3804	0.3764	0.3713	0.3682	0.3656

Table 3: Comparison of the methods under different missing rates on the simulated multi-category rating data with 1500 users and 1500 items.

Missing rate	RSVD	RBM	RB	SI	GS	TML	TMLCI
0.7	1.0523	1.0341	1.0764	1.0352	0.9724	0.9754	0.9514
0.8	1.0584	1.0404	1.0954	1.0567	0.9745	0.9721	0.9537
0.9	1.0745	1.0632	1.1177	1.0632	0.9798	0.9823	0.9601
0.95	1.1084	1.0903	1.1326	1.0844	0.9834	0.9878	0.9654

Table 4: Comparison of the methods in the cold-start problem on the simulated five-category rating data.

	RSVD	RBM	RB	SI	GS	TML	TMLCI
old	0.9761	0.9428	0.9817	0.9739	0.9221	0.9175	0.9013
new	1.2033	1.2204	1.2494	1.1834	1.0923	1.0799	1.0692
average	1.0956	1.0905	1.1236	1.0837	1.0108	1.0020	0.9888

Table 5: Comparison of the methods on MovieLens 1M and 10M data.

	RSVD	RBM	RB	SI	GS	TML	TMLCI
1M	1.0552	1.0128	1.1974	1.0737	0.9644	0.9692	0.9487
10M	0.9966	0.9772	0.9737	1.0177	0.9295	0.9243	0.9207