JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2017, VOL. 112, NO. 519, 1344–1353, Theory and Methods
https://doi.org/10.1080/01621459.2016.1219261

A Group-Specific Recommender System

Xuan Bi[a], Annie Qu[b], Junhui Wang[c], and Xiaotong Shen[a,d]

[a]Department of Statistics, University of Illinois at Urbana-Champaign, Champaign, IL; [b]Yunnan University, China, and Department of Statistics, University of Illinois at Urbana-Champaign, Champaign, IL; [c]Department of Mathematics, City University of Hong Kong, Hong Kong; [d]School of Statistics, University of Minnesota, Minneapolis, MN

ABSTRACT

In recent years, there has been a growing demand to develop efficient recommender systems which track users' preferences and recommend potential items of interest to users. In this article, we propose a group-specific method to use dependency information from users and items which share similar characteristics under the singular value decomposition framework. The new approach is effective for the "cold-start" problem, where, in the testing set, majority responses are obtained from new users or for new items, and their preference information is not available from the training set. One advantage of the proposed model is that we are able to incorporate information from the missing mechanism and group-specific features through clustering based on the numbers of ratings from each user and other variables associated with missing patterns. In addition, since this type of data involves large-scale customer records, traditional algorithms are not computationally scalable. To implement the proposed method, we propose a new algorithm that embeds a back-fitting algorithm into alternating least squares, which avoids large matrices operation and big memory storage, and therefore makes it feasible to achieve scalable computing. Our simulation studies and MovieLens data analysis both indicate that the proposed group-specific method improves prediction accuracy significantly compared to existing competitive recommender system approaches. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received November 2015
Revised July 2016

KEYWORDS
Cold-start problem; Group-specific latent factors; Nonrandom missing observations; Personalized prediction

1. Introduction

Recommender systems have drawn great attention since they can be applied to many areas, such as movies reviews, restaurant and hotel selection, financial services, and even identifying gene therapies. Therefore, there is a great demand to develop efficient recommender systems which track users' preferences and recommend potential items of interest to users.

However, developing competitive recommender systems brings new challenges, as information from both users and items could grow exponentially, and the corresponding utility matrix representing users' preferences over items are sparse and high-dimensional. The standard methods and algorithms which are not scalable in practice may suffer from rapid deterioration on recommendation accuracy as the volume of data increases.

In addition, it is important to incorporate dynamic features of data instead of one-time usage only, as data could stream in over time and grow exponentially. For example, in the MovieLens 10M data, 96% of the most recent ratings are either from new users or on new items which did not exist before. This implies that the information collected at an early time may not be representative for future users and items. This phenomenon is also called the "cold-start" problem, where, in the testing set, majority responses are obtained from new users or for new items, and their preference information is not available from the training set. Another important feature of this type of data is that the missing mechanism is likely nonignorable missing, where the missing mechanism is associated with unobserved responses. For instance, items with fewer and lower rating scores are less likely to attract other users. Existing recommender systems typically assume missing completely at random, which may lead to estimation bias.

Content-based filtering and collaborative filtering are two of the most prevalent approaches for recommender systems. Content-based filtering methods (e.g., Lang 1995; Mooney and Roy 2000; Blanco-Fernandez et al. 2008) recommend items by comparing the content of the items with a user's profile, which has the advantage that new items can be recommended upon release. However, domain knowledge is often required to establish a transparent profile for each user (Lops et al. 2011), which entails pre-processing tasks to formulate information vectors for items (Pazzani and Billsus 2007). In addition, content-based filtering suffers from the "cold-start" problem as well when a new user is recruited (Adomavicius and Tuzhilin 2005).

For collaborative filtering, the key idea is to borrow information from similar users to predict their future actions. One significant advantage is that the domain knowledge for items is not required. Popular collaborative filtering approaches include, but are not limited to, singular value decomposition (SVD; Funk 2006; Mazumder et al. 2010), restricted Boltzman machines (RBM; Salakhutdinov et al. 2007), and the nearest neighbor methods (kNN; Bell and Koren 2007). It is well-known that an ensemble of these methods could further enhance prediction accuracy. (See Cacheda et al. 2011 and Feuerverger et al. 2012 for extensive reviews.)

CONTACT Annie Qu anniequ@illinois.edu Department of Statistics, University of Illinois at Urbana-Champaign, 725 South Wright Street, Champaign, IL 61820.
Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JASA.

However, most existing collaborative filtering approaches do not effectively solve the "cold-start" problem, although various attempts have been made. For example, Park et al. (2006) suggested adding artificial users or items with pre-defined characteristics, while Goldberg et al. (2001), Melville et al. (2002), and Nguyen et al. (2007) considered imputing "pseudo" ratings. Most recently, a hybrid system incorporating content-based auxiliary information has been proposed (e.g., Agarwal and Chen 2009; Nguyen and Zhu 2013; Zhu et al. 2016). Nevertheless, the "cold-start" problem imposes great challenges and has not been effectively solved.

In this article, we propose a group-specific singular value decomposition method that generalizes the SVD model by incorporating between-subject dependency and uses information of missingness. Specifically, we cluster users or items based on their missingness-related characteristics. We assume that individuals within the same cluster are correlated, while individuals from different clusters are independent. The cluster correlation is incorporated through mixed-effects modeling assuming that users or items from the same cluster share the same group effects, along with latent factors modeling using singular value decomposition.

The proposed method has two significant contributions. First, it solves the "cold-start" problem effectively through incorporating group effects. Most collaborative filtering methods rely on subject-specific parameters to predict users' and items' future ratings. However, for a new user or item, the training samples provide no information to estimate such parameters. In contrast, we are able to incorporate additional group information for new users and items to achieve higher prediction accuracy. Second, our clustering strategy takes nonignorable missingness into consideration. In the MovieLens data, we notice that individuals' rating behaviors are highly associated with their missing patterns: movies with higher average rating scores attract more viewers, while frequent viewers tend to be more critical and give low ratings. We cluster individuals into groups based on their nonrandom missingness, and this allows us to capture individuals' latent characteristics which are not used in other approaches.

To implement the proposed method, we propose a new algorithm that embeds a back-fitting algorithm into alternating least squares, which avoids large matrices operation and big memory storage, and makes it feasible to achieve scalable computing in practice. Notice that the proposed algorithm guarantees convergence to a stationary point which is a local minimum along each block direction, while our theoretical results provide general properties of the global minimum. In general, this distinction occurs in nonconvex optimization problems (e.g., Zhou et al. 2013; Zhu et al. 2016), since nonconvex optimization is NP-hard (Ge et al. 2015). Our numerical studies indicate that the proposed method is effective in terms of prediction accuracy. For example, for the MovieLens 1M and 10M data, the proposed method improves prediction accuracy significantly compared to existing competitive recommender system approaches (e.g., Agarwal and Chen 2009; Koren et al. 2009; Mazumder et al. 2010; Zhu et al. 2016).

This article is organized as follows. Section 2 provides the background of the singular value decomposition model and introduces the proposed method. Section 3 presents the proposed method, a new algorithm, and its implementation. Section 4 establishes the theoretical foundation of the proposed method. In Section 5, we illustrate the performance and robustness of the proposed method through simulation studies. MovieLens 1M and 10M data are analyzed in Section 6. Section 7 provides concluding remarks and discussion.

2. Background and Model Framework

2.1. Background

We provide the background of the singular value decomposition method (Funk 2006) as follows. Let $\mathbf{R} = (r_{ui})_{n\times m}$ be the utility matrix, where n is the number of users, m is the number of items, and each r_{ui} is an explicit rating from user u for item i ($u = 1, \ldots, n$, $i = 1, \ldots, m$). The SVD method decomposes the utility matrix $\mathbf{R}$ as

$$\mathbf{R} = \mathbf{P}\mathbf{Q}',$$

where $\mathbf{R}$ is assumed to be low-rank, $\mathbf{P} = (\mathbf{p}_1, \ldots, \mathbf{p}_n)'$ is an $n \times K$ user preference matrix, $\mathbf{Q} = (\mathbf{q}_1, \ldots, \mathbf{q}_m)'$ is an $m \times K$ item preference matrix, and K is the pre-specified upper bound of the number of latent factors, which corresponds to the rank of $\mathbf{R}$. Here, $\mathbf{q}_i$ and $\mathbf{p}_u$ are K-dimensional latent factors associated with item i and user u, respectively, which explain variability in $\mathbf{R}$.

The predicted value of r_{ui} given by the SVD method is: $\hat{r}_{ui} = \hat{\mathbf{p}}_u'\hat{\mathbf{q}}_i$, where $\hat{\mathbf{q}}_i$ and $\hat{\mathbf{p}}_u$ are estimated iteratively by

$$\hat{\mathbf{q}}_i = \underset{q_i}{\operatorname{argmin}} \sum_{u\in U_i} (r_{ui} - \mathbf{p}_u'\mathbf{q}_i)^2 + \lambda\|\mathbf{q}_i\|_2^2,$$

and

$$\hat{\mathbf{p}}_u = \underset{p_u}{\operatorname{argmin}} \sum_{i\in I_u} (r_{ui} - \mathbf{p}_u'\mathbf{q}_i)^2 + \lambda\|\mathbf{p}_u\|_2^2.$$

Here, U_i denotes the set of all users who rate item i, and I_u is the set of all items rated by user u. Different penalty functions can be applied. For example, Zhu et al. (2016) suggested L_0 and L_1 penalties to achieve sparsity of $\mathbf{P}$ and $\mathbf{Q}$. In addition, some SVD methods (e.g., Koren 2010; Mazumder et al. 2010; Nguyen and Zhu 2013) are implemented on residuals after a baseline fit, such as linear regression or ANOVA, rather than the raw ratings r_{ui} directly.

The SVD method can be carried out through several algorithms, for example, the alternating least square (ALS; Carroll and Chang 1970; Harshman 1970; Koren et al. 2009), gradient descent approaches (Wu 2007), and one-feature-at-a-time ALS (Funk 2006).

2.2. Model Framework

The general framework of the proposed method is constructed as follows. Suppose $\mathbf{x}_{ui}$ is a covariate vector corresponding to the user u and item i. In the rest of this article, we consider $r_{ui} - \mathbf{x}_{ui}'\hat{\boldsymbol{\beta}}$ as the new response, where $\hat{\boldsymbol{\beta}}$ is the linear regression coefficient of $\mathbf{x}_{ui}$ to fit r_{ui}. To simplify our notation, we still use r_{ui} to denote the residual here. In case covariate information is not available, we apply the ANOVA-type model where the grand mean, the user main effects, and the item main effects are subtracted and replace r_{ui} by its residual.

Let $\theta_{ui} = E(r_{ui})$. We generalize the SVD model and formulate each θ_{ui} as

$$\theta_{ui} = (\mathbf{p}_u + \mathbf{s}_{v_u})'(\mathbf{q}_i + \mathbf{t}_{j_i}), \tag{1}$$

where $\mathbf{s}_{v_u}$ and $\mathbf{t}_{j_i}$ are K-dimensional group effects that are identical across members from the same cluster. We denote users from the v-th cluster as $V_v = \{u : v_u = v\}$ $(v = 1, \ldots, N)$, and items from the jth cluster as $J_j = \{i : j_i = j\}$ $(j = 1, \ldots, M)$, where $\sum_{v=1}^{N} |V_v| = n$ and $\sum_{j=1}^{M} |J_j| = m$, $|\cdot|$ is the cardinality of a set, and N and M are the total number of clusters for users and items, respectively. Details about selecting N and M are provided in Section 3.3.

In matrix form, we use $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_N)'$ and $\mathbf{T} = (\mathbf{t}_1, \ldots, \mathbf{t}_M)'$ to denote the user and item group-effect matrices, respectively. However, the dimensions of matrix $\mathbf{S}$ and $\mathbf{T}$ are $N \times K$ and $M \times K$, which are not compatible with the dimensions of $\mathbf{P}$ and $\mathbf{Q}$, respectively. Therefore, alternatively we define $\mathbf{S}_c = (\mathbf{s}_1\mathbf{1}'_{|V_1|}, \ldots, \mathbf{s}_N\mathbf{1}'_{|V_N|})'$ and $\mathbf{T}_c = (\mathbf{t}_1\mathbf{1}'_{|J_1|}, \ldots, \mathbf{t}_M\mathbf{1}'_{|J_M|})'$, corresponding to group-effects from users and items, where $\mathbf{1}_k$ is a k-dimensional vector of 1's, and the subscript "c" in $\mathbf{S}_c$ and $\mathbf{T}_c$ denotes the "complete" forms of matrices. Let $\mathbf{\Theta} = (\theta_{ui})_{n\times m}$, then we have

$$\mathbf{\Theta} = (\mathbf{P} + \mathbf{S}_c)(\mathbf{Q} + \mathbf{T}_c)',$$

and if there are no group effects, $\mathbf{\Theta}$ degenerates to $\mathbf{\Theta} = \mathbf{P}\mathbf{Q}'$, which is the same as the SVD model.

Here, the users or items can be formed as clusters based on their similar characteristics. For example, we can use missingness-related information such as the number of ratings from each user and each item. Users or items within the same cluster are correlated with each other through the group effects $\mathbf{s}_{v_u}$ or $\mathbf{t}_{j_i}$, while observations from different clusters are assumed to be independent. In Sections 3, 4, and 5.1, we assume N and M are known, and that members in each cluster are correctly labeled.

Remark 1. For easy operation, one could use users' and items' covariate information for clustering. In fact, (1) is still a generalization of the SVD method even if $N = M = 1$, because $\mathbf{s}'_{v_u}\mathbf{t}_{j_i}$, $\mathbf{p}'_u\mathbf{t}_{j_i}$, $\mathbf{s}'_{v_u}\mathbf{q}_i$ correspond to the grand mean, the user main effects and the item main effects, analogous to the ANOVA-type of SVD model. Note that covariate information might not be collected from new users and new items. However, missingness-related information is typically available for clustering, and therefore $\mathbf{s}_{v_u}$ and $\mathbf{t}_{j_i}$ can be used for new users and new items. This is crucial to solve the "cold-start" problem.

3. The General Method

3.1. Parameter Estimation

In this subsection, we illustrate how to obtain estimations of model parameters through training data. In addition, we develop a new algorithm that embeds back-fitting (Breiman and Friedman 1985) into alternating least squares. This enables us to circumvent large-scale matrix operations through a two-step iteration, and hence significantly improve computational speed and scalability.

Let $\boldsymbol{\gamma}$ be a vectorization of $(\mathbf{P}, \mathbf{Q}, \mathbf{S}, \mathbf{T})$, Ω be a set of user-item pairs associated with observed ratings, and $R^o = \{r_{ui} : (u, i) \in \Omega\}$ be a set of observed ratings. We define the loss function as

$$\mathcal{L}(\boldsymbol{\gamma}|R^o) = \sum_{(u,i)\in\Omega} (r_{ui} - \theta_{ui})^2 + \lambda \left(\sum_{u=1}^{n} \|\mathbf{p}_u\|_2^2 + \sum_{v=1}^{N} \|\mathbf{s}_v\|_2^2 + \sum_{i=1}^{m} \|\mathbf{q}_i\|_2^2 + \sum_{j=1}^{M} \|\mathbf{t}_j\|_2^2 \right), \tag{2}$$

where θ_{ui} is given by (1) and λ is a tuning parameter. We can estimate $\boldsymbol{\gamma}$ via

$$\hat{\boldsymbol{\gamma}} = \underset{\gamma}{\operatorname{argmin}}\, \mathcal{L}(\boldsymbol{\gamma}|R^o).$$

Then, the predicted value of θ_{ui} can be obtained by $\hat{\theta}_{ui} = (\hat{\mathbf{p}}_u + \hat{\mathbf{s}}_{v_u})'(\hat{\mathbf{q}}_i + \hat{\mathbf{t}}_{j_i})$.

The estimation procedure consists of updating $(\hat{\mathbf{p}}_u + \hat{\mathbf{s}}_{v_u})$ and $(\hat{\mathbf{q}}_i + \hat{\mathbf{t}}_{j_i})$ iteratively. Following the strategy of the alternating least squares, the latent factors and the group effects associated with item cluster j are estimated by

$$(\{\hat{\mathbf{q}}_i\}_{i\in J_j}, \hat{\mathbf{t}}_j) = \arg\min_{\{q_i\}_{i\in J_j}, t_j} \sum_{i\in J_j} \sum_{u\in U_i} (r_{ui} - \theta_{ui})^2 + \lambda \left(\sum_{i\in J_j} \|\mathbf{q}_i\|_2^2 + \|\mathbf{t}_j\|_2^2 \right). \tag{3}$$

Similarly, we estimate latent factors and group effects associated with user cluster v:

$$(\{\hat{\mathbf{p}}_u\}_{u\in V_v}, \hat{\mathbf{s}}_v) = \arg\min_{\{p_u\}_{u\in V_v}, s_v} \sum_{u\in V_v} \sum_{i\in I_u} (r_{ui} - \theta_{ui})^2 + \lambda \left(\sum_{u\in V_v} \|\mathbf{p}_u\|_2^2 + \|\mathbf{s}_v\|_2^2 \right). \tag{4}$$

However, directly solving (3) and (4) by the alternating least square encounters large matrices. In the MovieLens 10M data, it could involve matrices with more than 100,000 rows. We develop a new algorithm which embeds back-fitting into alternating least squares, and minimize each of (3) and (4) iteratively. Specifically, for each item cluster J_j $(j = 1, \ldots, M)$, we fix $\mathbf{P}$ and $\mathbf{S}$, and minimize (3) through estimating $\hat{\mathbf{q}}_i$ and $\hat{\mathbf{t}}_j$ iteratively:

$$\hat{\mathbf{q}}_i = \underset{q_i}{\operatorname{argmin}} \sum_{u\in U_i} (r_{ui} - \theta_{ui})^2 + \lambda \|\mathbf{q}_i\|_2^2, \; i \in J_j, \tag{5}$$

$$\hat{\mathbf{t}}_j = \underset{t_j}{\operatorname{argmin}} \sum_{i\in J_j} \sum_{u\in U_i} (r_{ui} - \theta_{ui})^2 + \lambda \|\mathbf{t}_j\|_2^2. \tag{6}$$

For each user cluster V_v $(v = 1, \ldots, N)$, we fix $\mathbf{Q}$ and $\mathbf{T}$ and minimize (4) through estimating $\hat{\mathbf{p}}_u$ and $\hat{\mathbf{s}}_v$ iteratively:

$$\hat{\mathbf{p}}_u = \underset{p_u}{\operatorname{argmin}} \sum_{i\in I_u} (r_{ui} - \theta_{ui})^2 + \lambda \|\mathbf{p}_u\|_2^2, \; u \in V_v, \tag{7}$$

$$\hat{\mathbf{s}}_v = \underset{s_v}{\operatorname{argmin}} \sum_{u\in V_v} \sum_{i\in I_u} (r_{ui} - \theta_{ui})^2 + \lambda \|\mathbf{s}_v\|_2^2. \tag{8}$$

The above backfitting is an iterative algorithm for additive models. In contrast, the alternating least squares is an iterative algorithm for multiplicative models. Although both backfitting and alternating least squares are blockwise coordinate descent methods in general, their convergence properties are different under our framework. This is because the objective function for the backfitting algorithm is convex, while the objective function for the alternating least squares is nonconvex. Ansley and Kohn (1994) showed that for penalized least-square problems, the backfitting algorithm converges to the unique optimum solution from any initial values, while the alternating least-squares algorithm for two blocks only converges to a stationary point (Chen et al. 2012).

In addition, the proposed algorithm is also different from the blockwise coordinate descent algorithm which estimates each of $(\mathbf{P}, \mathbf{Q}, \mathbf{S}, \mathbf{T})$ sequentially and iteratively while keeping the other terms as constants. The convergence property of the proposed algorithm is illustrated in Section 3.2. Note that the blockwise coordinate descent algorithm does not have such a property.

3.2. Algorithm

In this section, we provide the detailed algorithm as follows.

Algorithm 1. Parallel Computing for the Proposed Method

1. *(Initialization)* Set $l = 1$. Set initial values for $(\mathbf{P}^{(0)}, \mathbf{Q}^{(0)}, \mathbf{S}^{(0)}, \mathbf{T}^{(0)})$ and the tuning parameter λ.
2. *(Item Effects)* Estimate $\mathbf{Q}^{(l)}$ and $\mathbf{T}^{(l)}$ iteratively.
   (i) Set $\mathbf{Q}^{(l)} \leftarrow \mathbf{Q}^{(l-1)}$, and set $\mathbf{T}^{(l)} \leftarrow \mathbf{T}^{(l-1)}$.
   (ii) For each item $i = 1, \ldots, m$, calculate $\mathbf{q}_i^{(l)_{\text{new}}}$ using (5).
   (iii) For each item cluster J_j, $j = 1, \ldots, M$, calculate $\mathbf{t}_j^{(l)_{\text{new}}}$ based on (6).
   (iv) Stop iteration if $\frac{1}{mK}\|\mathbf{Q}^{(l)_{\text{new}}} - \mathbf{Q}^{(l)}\|_F^2 + \frac{1}{MK}\|\mathbf{T}^{(l)_{\text{new}}} - \mathbf{T}^{(l)}\|_F^2 < 10^{-5}$, otherwise assign $\mathbf{Q}^{(l)} \leftarrow \mathbf{Q}^{(l)_{\text{new}}}$ and $\mathbf{T}^{(l)} \leftarrow \mathbf{T}^{(l)_{\text{new}}}$, and go to Step 2(ii).
3. *(User Effects)* Estimate $\mathbf{P}^{(l)}$ and $\mathbf{S}^{(l)}$ iteratively.
   (i) Set $\mathbf{P}^{(l)} \leftarrow \mathbf{P}^{(l-1)}$, and set $\mathbf{S}^{(l)} \leftarrow \mathbf{S}^{(l-1)}$.
   (ii) For each user $u = 1, \ldots, n$, calculate $\mathbf{p}_u^{(l)_{\text{new}}}$ using (7).
   (iii) For each user cluster V_v, $v = 1, \ldots, N$, calculate $\mathbf{s}_v^{(l)_{\text{new}}}$ based on (8).
   (iv) Stop iteration if $\frac{1}{nK}\|\mathbf{P}^{(l)_{\text{new}}} - \mathbf{P}^{(l)}\|_F^2 + \frac{1}{NK}\|\mathbf{S}^{(l)_{\text{new}}} - \mathbf{S}^{(l)}\|_F^2 < 10^{-5}$, otherwise assign $\mathbf{P}^{(l)} \leftarrow \mathbf{P}^{(l)_{\text{new}}}$ and $\mathbf{S}^{(l)} \leftarrow \mathbf{S}^{(l)_{\text{new}}}$, and go to Step 3(ii).
4. *(Stopping criterion)* Stop if $\frac{1}{nK}\|\mathbf{P}^{(l)} + \mathbf{S}_c^{(l)} - \mathbf{P}^{(l-1)} - \mathbf{S}_c^{(l-1)}\|_F^2 + \frac{1}{mK}\|\mathbf{Q}^{(l)} + \mathbf{T}_c^{(l)} - \mathbf{Q}^{(l-1)} - \mathbf{T}_c^{(l-1)}\|_F^2 < 10^{-3}$, otherwise set $l \leftarrow l + 1$ and go to Step 2.

Note that the alternating least square is performed by conducting Steps 2 and 3 iteratively, while the back-fitting algorithm is carried out within Steps 2 and 3. The parallel computing can be implemented in Steps 2(ii), (iii) and 3(ii), (iii).

Algorithm 1 does not require large computational and storage cost. We denote I_{B1}, I_{B2} and I_{ALS} as the numbers of iterations for back-fitting in Steps 2 and 3, and the ALS, respectively, and C_{Ridge} as the computational complexity of solving the ridge regression with K variables and $\max\{|V_1|, \ldots, |V_N|, |J_1|, \ldots, |J_M|\}$ observations. Then, the computational complexity of Algorithm 1 is no greater than $\{(m+M)I_{B1} + (n+N)I_{B2}\}C_{\text{Ridge}}I_{\text{ALS}}$. Since both ridge regression and Lasso have the same computational complexity as ordinary least squares (Efron et al. 2004), the computational cost of the proposed method is indeed no greater than that of Zhu et al. (2016). For the storage cost, Algorithm 1 requires storages of only item-specific or user-specific information to solve (5) or (7), and the sizes of items and users information not exceeding $\max\{|J_1|, \ldots, |J_M|\}$ and $\max\{|V_1|, \ldots, |V_N|\}$ to solve (6) or (8), respectively.

We also establish the convergence property of Algorithm 1 as follows. Let $\boldsymbol{\gamma}^* = \text{vec}(\mathbf{P}^*, \mathbf{Q}^*, \mathbf{S}^*, \mathbf{T}^*)$ be a stationary point of $\mathcal{L}(\boldsymbol{\gamma}|R^o)$ corresponding to two blocks. That is,

$$\text{vec}(\mathbf{P}^*, \mathbf{S}^*) = \underset{P,S}{\text{argmin}}\, \mathcal{L}\left(\text{vec}(\mathbf{P}, \mathbf{Q}^*, \mathbf{S}, \mathbf{T}^*)|R^o\right),$$

and

$$\text{vec}(\mathbf{Q}^*, \mathbf{T}^*) = \underset{Q,T}{\text{argmin}}\, \mathcal{L}\left(\text{vec}(\mathbf{P}^*, \mathbf{Q}, \mathbf{S}^*, \mathbf{T})|R^o\right).$$

The following lemma shows the convergence of Algorithm 1 to a stationary point, which is a local minimum along each block direction. One way to approximate the global minimum is to adopt the branch-and-bound technique, and search all possible local minima (Liu et al. 2005). However, this technique could be computationally intensive.

Lemma 1. The estimate $\hat{\boldsymbol{\gamma}} = \text{vec}(\hat{\mathbf{P}}, \hat{\mathbf{Q}}, \hat{\mathbf{S}}, \hat{\mathbf{T}})$ from Algorithm 1 is a stationary point of the loss function $\mathcal{L}(\boldsymbol{\gamma}|R^o)$ in (2).

3.3. Implementation

In this subsection, we address some implementation issues for the proposed method. To select tuning parameter λ, we search from grid points which minimizes the root mean square error (RMSE) on the validation set. The RMSE on a given set Ω_0 is defined as $\{\frac{1}{|\Omega_0|}\sum_{(u,i)\in\Omega_0}(r_{ui} - \hat{\theta}_{ui})^2\}^{1/2}$. In selection of the number of latent factors K, we choose K such that it is sufficiently large and leads to stable estimations. In general, K needs to be larger than the rank of the utility matrix $\mathbf{R}$, but not so large as to intensify the computation. Regarding the selection of the number of clusters N and M, Corollary 2 of Section 4 provides the lower bound in the order of $O(N)$ and $O(M)$. Note that too small N and M may not have the power to distinguish between the proposed method and the SVD method. In practice, if clustering is based on categorical variables, then we can apply the existing categories, and N and M are known. However, if clustering is based on a continuous variable, we can apply the quantiles of the continuous variable to determine N and M and categorize users and items evenly. We then select the number of clusters through a grid search, similar to the selection of λ and K. See Wang (2010) for a consistent selection of the number of clusters in more general settings.

In particular, for our numerical studies, we split our dataset into 60% training, 15% validation, and 25% testing sets based on the time of ratings (timestamps; Zhu et al. 2016). That is, we use historical data to predict future data. If time information is not

available, we use a random split to determine training, validation, and testing sets instead.

4. Theory

In this section, we provide the theoretical foundation of the proposed method in a general setting. That is, we allow r_{ui} to follow a general class of distributions. In particular, we derive an upper bound for the prediction error in probability, and show that existing approaches without using group effects lead to a larger value of the loss function, and therefore are less efficient compared to the proposed method. Furthermore, we establish a lower bound of the number of clusters which guarantees that the group effects can be detected effectively.

Suppose the expected value of each rating is formulated via a known mean function μ. That is,

$$E(r_{ui}) = \mu(\theta_{ui}),$$

and θ_{ui} is defined as in (1). For example, if r_{ui} is a continuous variable, then $\mu(\theta_{ui}) = \theta_{ui}$; and if r_{ui}'s are binary, then $\mu(\theta_{ui}) = \frac{\exp(\theta_{ui})}{1+\exp(\theta_{ui})}$.

We let $f_{ui} = f(r_{ui}|\theta_{ui})$ be the probability density function of r_{ui}. Since each r_{ui} is associated with $\boldsymbol{\gamma}$ only through θ_{ui}, we denote $f_{ui}(r, \boldsymbol{\gamma}) = f(r_{ui}|\theta_{ui})$. We define the likelihood-based loss function as:

$$\mathcal{L}(\boldsymbol{\gamma}|R^o) = -\sum_{(u,i)\in\Omega} \log f_{ui} + \lambda_{|\Omega|} D(\boldsymbol{\gamma}),$$

where $\lambda_{|\Omega|}$ is the penalization coefficient, $|\Omega|$ is the total number of observed ratings, and $D(\cdot)$ is a nonnegative penalty function of $\boldsymbol{\gamma}$. For example, we have $D(\boldsymbol{\gamma}) = \|\boldsymbol{\gamma}\|_2^2$ for the L_2-penalty.

Since, in practice, the ratings are typically nonnegative finite values, it is sensible to assume $\|\boldsymbol{\gamma}\|_\infty \le L$, where L is a positive constant. We define the parameter vector space as

$$\mathcal{S}(k) = \{\boldsymbol{\gamma} : \|\boldsymbol{\gamma}\|_\infty \le L, D(\boldsymbol{\gamma}) \le k^2\}.$$

Notice that the dimension of $\boldsymbol{\gamma}$ is $\dim(\boldsymbol{\gamma}) = (n+m+N+M)K$ which goes to infinity as either n or m increases. Therefore, we assume $k \sim O(\sqrt{(n+m+N+M)K})$. Similarly, we define the parameter space for each θ_{ui}: $\mathcal{S}_\Theta(k) = \{\theta : \|\boldsymbol{\gamma}\|_\infty \le L, D(\boldsymbol{\gamma}) \le k^2\}$.

Assumption 1. For some constant $\bar{G} \ge 0$, and $\theta_{ui}, \tilde{\theta}_{ui} \in \mathcal{S}_\Theta(k)$,

$$\left| f^{1/2}(r_{ui}|\theta_{ui}) - f^{1/2}(r_{ui}|\tilde{\theta}_{ui}) \right| \le G(r_{ui}) \|\theta_{ui} - \tilde{\theta}_{ui}\|_2,$$

where $EG^2(r_{ui}) \le \bar{G}^2$ for $u = 1, \ldots, n$, $i = 1, \ldots, m$.

In the following, we discuss the convergence properties based on the probability density function f_{ui} rather than the parameter itself. This is because with the knowledge of $f_{ui}(r, \boldsymbol{\gamma})$, we can estimate the predicted rating $\mathrm{E}(r_{ui})$, the corresponding variance $\mathrm{var}(r_{ui})$, and other distributional features of r_{ui}. Specifically, we employ the Hellinger metric to measure the distance between two density functions, since it is the most convenient metric associated with density functions and is always well-defined (Van de Geer 1993). The Hellinger metric $h_\Theta(\cdot,\cdot)$ on $\mathcal{S}_\Theta(k)$ is defined as

$$h_\Theta(\theta_{ui}, \tilde{\theta}_{ui}) = \left[\int \{ f^{1/2}(r_{ui}|\theta_{ui}) - f^{1/2}(r_{ui}|\tilde{\theta}_{ui}) \}^2 d\nu(r_{ui}) \right]^{1/2},$$

where $\nu(\cdot)$ is a probability measure. Based on Assumption 1, $h_\Theta(\theta_{ui}, \tilde{\theta}_{ui})$ is bounded by $\|\theta_{ui} - \tilde{\theta}_{ui}\|_2$.

We now define the Hellinger metric $h_S(\cdot,\cdot)$ on $\mathcal{S}(k)$. For $\boldsymbol{\gamma}, \tilde{\boldsymbol{\gamma}} \in \mathcal{S}(k)$, let

$$h_S(\boldsymbol{\gamma}, \tilde{\boldsymbol{\gamma}}) = \left\{ \frac{1}{nm} \sum_{i=1}^{m} \sum_{u=1}^{n} h_\Theta^2(\theta_{ui}, \tilde{\theta}_{ui}) \right\}^{1/2}.$$

It is straightforward to show that h_S is still a metric. In the rest of this article, we suppress the subscript and use $h(\cdot,\cdot)$ to denote the Hellinger metric on $\mathcal{S}(k)$. In the following, we show that $h(\boldsymbol{\gamma}, \tilde{\boldsymbol{\gamma}})$ can be bounded by $\|\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}}\|_2$.

Lemma 2. Under Assumption 1, there exists a constant $d_0 \ge 0$, such that for $\boldsymbol{\gamma}, \tilde{\boldsymbol{\gamma}} \in \mathcal{S}(k)$,

$$h(\boldsymbol{\gamma}, \tilde{\boldsymbol{\gamma}}) \le d_0 \sqrt{\frac{n+m}{nm}} \|\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}}\|_2.$$

Suppose $\hat{\boldsymbol{\gamma}} = \mathrm{argmin}_{\gamma\in\mathcal{S}(k)} \mathcal{L}(\boldsymbol{\gamma}|R^o)$ is a penalized maximum likelihood estimator of $\boldsymbol{\gamma}$. Theorem 1 indicates that $\hat{\boldsymbol{\gamma}}$ converges to $\boldsymbol{\gamma}$ exponentially in probability, with a convergence rate of $\epsilon_{|\Omega|}$.

Theorem 1. Under Assumption 1 and suppose $\lambda_{|\Omega|} < \frac{1}{2k}\epsilon_{|\Omega|}^2$, the best possible convergence rate of $\hat{\boldsymbol{\gamma}}$ is

$$\epsilon_{|\Omega|} \sim \frac{\sqrt{(n+m)K}}{|\Omega|^{1/2}} \left\{ \log\left(\frac{|\Omega|}{\sqrt{nmK}} \right) \right\}^{1/2},$$

and there exists a constant $c > 0$, such that

$$P\left(h(\hat{\boldsymbol{\gamma}}, \boldsymbol{\gamma}) \ge \epsilon_{|\Omega|} \right) \le 7 \exp\left(-c|\Omega|\epsilon_{|\Omega|}^2 \right).$$

Remark 2. Theorem 1 can be generalized to achieve the convergence property measured by the L_2 distance as a special case of Corollary 2 in Shen (1998). However, the convergence under the L_2 distance is more restrictive than the convergence under the Hellinger distance. In addition, the Hellinger distance has the following advantages. First, the convergence rate of $\hat{\boldsymbol{\gamma}}$ depends only on the size of the parameter space $\mathcal{S}(k)$ and the penalization coefficient $\lambda_{|\Omega|}$ (Shen 1998). In contrast, the convergence rate based on the L_2 distance depends on additional local and global behavior of $\mathrm{var}\{\mathcal{L}(\hat{\boldsymbol{\gamma}}|R^o) - \mathcal{L}(\boldsymbol{\gamma}|R^o)\}$. Second, the exponential bound under the Hellinger distance does not rely on the existence of the moment generating function of $G(\cdot)$, which is needed for the exponential bound under the L_2 distance.

Remark 3. Theorem 1 is quite general in terms of the rates of n and m. If we assume $O(n) = O(m) = O(n+m)$ such as in the MovieLens data, then $\epsilon_{|\Omega|}$ converges faster than $\epsilon_{|\Omega|}^{SAJ}$, where $\epsilon_{|\Omega|}^{SAJ} \sim \frac{\sqrt{(n+m)K}}{|\Omega|^{1/2}} \{\log(\frac{|\Omega|}{(n+m)k})\}^{1/2} \{\log(\frac{m}{k})\}^{1/2}$ is the convergence rate provided by the collaborative prediction method with binary ratings (Srebro et al. 2005). The exact rate comparison is not available here.

Remark 4. The definition of $\mathcal{S}(k)$ is for the purpose of achieving the best possible convergence rate. Specifically, let $\mathcal{S} \in \mathbb{R}^{(n+m+N+M)K}$ be the true underlying parameter space. Since $\mathcal{S}$

is in an infinite-dimensional space when n or m goes to infinity, $\hat{\boldsymbol{\gamma}}$ obtained by optimizing over $\mathcal{S}$ may not achieve the best possible convergence rate (Shen and Wong 1994). Instead, we adopt the idea of *sieve* MLE (Grenander 1981), and approximate $\mathcal{S}$ by $\mathcal{S}(k)$ which grows as the sample size increases. This ensures that the penalized MLE $\hat{\boldsymbol{\gamma}}$ on $\mathcal{S}(k)$ is capable of achieving the best possible convergence rate (Shen 1998).

Remark 5. If we impose $\|\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}\|_2 \le d_{n,m}$ with radius $d_{n,m} = \sqrt{\frac{2nm}{d_0^2(n+m)}}\epsilon_{|\Omega|}$, then the entropy of $\mathcal{S}(k)$ under Assumption 1 also satisfies the condition of local entropy (Wong and Shen 1995). That is,

$$\mathcal{S}(k) = \mathcal{S}(k) \cap \left\{ \frac{1}{nm} \sum_{i=1}^{m} \sum_{u=1}^{n} \| f^{1/2}(r_{ui}, \boldsymbol{\gamma}) - f^{1/2}(r_{ui}, \hat{\boldsymbol{\gamma}}) \|_2^2 \le 2s^2 \right\}, \text{ for all } s \ge \epsilon_{|\Omega|}.$$

Consequently, the convergence rate of $\epsilon_{|\Omega|}$ is $\log(|\Omega|)$ times faster than the convergence rate calculated by using global entropy.

We now assume that the density function f_{ui} is a member of the exponential family in its canonical form. That is,

$$f(r_{ui}|\theta_{ui}) = H(r_{ui}) \exp\{\theta_{ui} T(r_{ui}) - A(\theta_{ui})\}.$$

In fact, the following results still hold if f is in the overdispersed exponential family.

Suppose $\boldsymbol{\gamma} \in \mathcal{S}(k)$ and $\theta_{ui} \in \mathcal{S}_\Theta(k)$ are the true parameters. Then, Theorem 2 indicates that if misspecified $\tilde{\theta}_{ui}$'s are not close to θ_{ui}'s, then the loss function of the corresponding $\tilde{\boldsymbol{\gamma}}$ cannot be closer to the loss function of $\boldsymbol{\gamma}$ than a given threshold in probability.

Theorem 2. Under Assumption 1 and $\lambda_{|\Omega|} < \frac{1}{2k}\epsilon_{|\Omega|}^2$, there exist $c_i > 0$, $i = 1, 2$, such that for $\epsilon_{|\Omega|} > 0$, there exists $\delta_{|\Omega|} > 0$, and $\min_{1\le u\le n, 1\le i\le m} |\tilde{\theta}_{ui} - \theta_{ui}| > \delta_{|\Omega|}$ implies that

$$P^*\left(\frac{1}{|\Omega|}\{\mathcal{L}(\tilde{\boldsymbol{\gamma}}|R^o) - \mathcal{L}(\boldsymbol{\gamma}|R^o)\} > c_1\epsilon_{|\Omega|}^2\right) \ge 1 - 7\exp(-c_2|\Omega|\epsilon_{|\Omega|}^2),$$

where P^* denotes the outer measure (Pollard 2012).

Remark 6. Theorems 1 and 2 still hold if the loss function $\mathcal{L}(\cdot|\cdot)$ is not likelihood-based, but is a general criterion function. For such $\mathcal{L}(\cdot|\cdot)$, we can replace $h(\cdot,\cdot)$ by $\rho(\cdot,\cdot) = K^{1/2}(\cdot,\cdot)$ as the new measure of convergence, where $K(\boldsymbol{\gamma}, \tilde{\boldsymbol{\gamma}}) = E\{\mathcal{L}(\boldsymbol{\gamma}|\mathbf{R}) - \mathcal{L}(\tilde{\boldsymbol{\gamma}}|\mathbf{R})\}$. Note that $K(\cdot,\cdot)$ is the Kullback–Leiber information if $\mathcal{L}(\cdot|\cdot)$ is a log-likelihood, which dominates the Hellinger distance $h(\cdot,\cdot)$, and hence the convergence is stronger under $K(\cdot,\cdot)$. See Shen (1998) for more details about regularity conditions for a more general criterion function.

Suppose $\boldsymbol{\gamma}_0 \in \mathcal{S}(k)$ is a vectorization of $(\mathbf{P}, \mathbf{Q}, \mathbf{0}, \mathbf{0})$, which corresponds to models with no group effects. The following Corollary 1 shows that if the true group effects are not close to 0, then existing methods ignoring group effects such as the SVD model ($\theta_{ui}^0 = \mathbf{p}_u'\mathbf{q}_i$) lead to a larger loss in probability than the proposed method.

Corollary 1. Under Assumption 1 and $\lambda_{|\Omega|} < \frac{1}{2k}\epsilon_{|\Omega|}^2$, there exists $c_i > 0$, $i = 1, 2$, and a constant $\phi \in (0, 1]$, such that for $\frac{1}{\sqrt{\phi}}\epsilon_{|\Omega|} > 0$, there exists $\delta_{|\Omega|} > 0$. Assume that at least (ϕnm) pairs of (u, i) satisfy $|\theta_{ui}^0 - \theta_{ui}| > \delta_{|\Omega|}$. Then

$$P^*\left(\frac{1}{|\Omega|}\{\mathcal{L}(\boldsymbol{\gamma}_0|R^o) - \mathcal{L}(\boldsymbol{\gamma}|R^o)\} > c_1\epsilon_{|\Omega|}^2\right) \ge 1 - 7\exp\left(-c_2|\Omega|\epsilon_{|\Omega|}^2\right).$$

The following corollary provides the minimal rate of N and M, in terms of n, m, K, and $|\Omega|$. This implies that the number of clusters should be sufficiently large so that the group effects can be detected.

Corollary 2. Under assumptions in Theorem 1, the rate of N and M satisfies

$$O(N + M) \ge \frac{nm}{|\Omega|} \log\left(\frac{|\Omega|}{\sqrt{nmK}}\right).$$

If we further assume that the number of ratings is proportional to the size of the utility matrix, that is, $O(|\Omega|) = O(nm)$, then $O(N + M) \ge \log(\frac{|\Omega|^{1/2}}{K^{1/2}})$. The lower bound of $O(N + M)$ is useful in determining the minimal number of clusters. For example, for the MovieLens 10M data where $|\Omega| = 10{,}000{,}000$, we have the lower bound $\log(\frac{|\Omega|^{1/2}}{K^{1/2}}) \approx 7$ if $K \le 10$.

5. Simulation Studies

In this section, we provide simulation studies to investigate the numerical performance of the proposed method in finite samples. Specifically, we compare the proposed method with four matrix factorization methods in Section 5.1 under a dynamic setting where new users and new items appear at later times. In Section 5.2, we test the robustness of the proposed model under various degrees of cluster misspecification.

5.1. Comparison Under the "Cold-Start" Problem

In this simulation studies, we simulate the "cold-start" problem, where new users' and new items' information is not available in the training set. In addition, we simulate that users' behavior is affected by other users' behavior, and therefore the missingness is not missing completely at random. Here, users and items from the same group are generated to be dependent from each other.

We set $n = 650$ and $m = 660$ and generate $\mathbf{p}_u, \mathbf{q}_i \overset{iid}{\sim} \mathrm{N}(0, \mathbf{I}_K)$ for $u = 1, \ldots, n$, $i = 1, \ldots, m$, where $\mathbf{I}_K$ is a K-dimensional identity matrix with $K = 3$ or 6. To simulate group effects, we let $\mathbf{s}_v = (-3.5 + 0.5v)\mathbf{1}_K$, $v = 1, \ldots, N$, and $\mathbf{t}_j = (-3.6 + 0.6j)\mathbf{1}_K$, $j = 1, \ldots, M$, where $N = 13$, $M = 11$. We set cluster size $|V_1| = \cdots = |V_N| = 50$, and $|J_1| = \cdots = |J_M| = 60$. Without loss of generality, we assume that covariate information is not available for this simulation.

In contrast to other simulation studies, we do not generate the entire utility matrix $\mathbf{R}$. Instead, we mimic the real data case, where only a small percentage of ratings is collected. We choose the total number of ratings to be $|\Omega| = (1 - \bar{\pi})nm$, where $\bar{\pi} = 0.7, 0.8, 0.9$, or 0.95 is the missing rate. The following procedure is used to generate these ratings.

We first select the lth user–item pair (u_l, i_l), where $l = 1, \ldots, |\Omega|$ indicates the sequence of ratings from the earliest to the latest. If item i_l's current average rating is greater than 0.5, then for user u_l, we assign a rating $r_{u_l i_l}$ with probability 0.85; otherwise we assign $r_{u_l i_l}$ with probability 0.2. The rating $r_{u_l i_l}$ is generated by $(\mathbf{p}_{u_l} + \mathbf{s}_{v_{u_l}})'(\mathbf{q}_{i_l} + \mathbf{t}_{j_{i_l}})/3 + \varepsilon$, where $\varepsilon \overset{\text{iid}}{\sim} \mathrm{N}(0, 1)$. That is, we simulate a setting where users tend to rate highly-rated items. Here, u_l and i_l are sampled from $1, \ldots, n$ and $1, \ldots, m$ independently, but with weights proportional to the density of normal distributions $\mathrm{N}(nl/|\Omega|, (0.2n)^2)$ and $\mathrm{N}(ml/|\Omega|, (0.2m)^2)$, respectively. That is, ratings appearing at a later time are more likely corresponding to newer users or to newer items. If we fail to assign $r_{u_l i_l}$ a value, we re-draw (u_l, i_l) and restart this procedure. The selection of $r_{u_l i_l}$ is based on observed information, so the missing mechanism is missing at random (Rubin 1976).

We compare the performance of the proposed method with four competitive matrix factorization models, namely, the regularized singular value decomposition method solved by the alternating least-square algorithm (RSVD; Funk 2006; Koren et al. 2009), a regression-based latent factor model (Agarwal and Chen 2009), a nuclear-norm matrix completion method (Soft-Impute; Mazumder et al. 2010), and a latent factor model with sparsity pursuit (Zhu et al. 2016). The proposed algorithm is implemented with Matlab, and the RSVD is solved in R. For the last three methods, we apply the codes in *https://github.com/beechung/latent-factor-models*, the R package "softImpute," and that of Zhu et al. (2016), respectively.

For the proposed method, we apply the loss function (2). The tuning parameter λ for the proposed method and the RSVD is selected from grid points ranging from 1 to 29 to minimize the RMSEs on the validation set. For Agarwal and Chen (2009), we use the default of 10 iterations, while for Mazumder et al. (2010), the default $\lambda = 0$ is chosen to achieve convergence for the local minimum; and for Zhu et al. (2016), the tuning parameter selection is integrated in their programming coding. We generate simulation settings when the number of latent factors $K = 3$ and 6, and the missing rate $\bar{\pi} = 0.7, 0.8, 0.9, 0.95$. The means and standard errors of RMSEs on the testing set are reported in Table 1. The simulation results are based on 500 replications.

Table 1 indicates that the proposed method performs the best across all settings. Overall, the proposed method is relatively robust against different missing rates or different numbers of latent factors, and has the smallest standard error in most settings. In the most extreme case with $K = 6$ and $\bar{\pi} = 0.95$, the proposed method is still more than 100% better than the best of the four existing methods in terms of the RMSEs. The RSVD method performs well when both $\bar{\pi}$ and K are small, but performs poorly when either $\bar{\pi}$ or K increases. By contrast, Agarwal and Chen (2009), Mazumder et al. (2010), and Zhu et al. (2016) are able to provide small standard errors when $K = 6$ and $\bar{\pi} = 0.95$, but have large RMSEs across all settings. Mazumder et al. (2010) occasionally provided outlying results due to a convergence problem when $\bar{\pi}$ is 0.9 or 0.95. We remove these extreme results in our simulations.

5.2. Robustness Against Cluster Misspecification

In this simulation study, we test the robustness of the proposed method when the clusters are misspecified.

We follow the same data-generating process as in the previous study, but allow the cluster assignment to be misspecified. Specifically, we misassign users and items to adjacent clusters with 10%, 30%, and 50% chance. Here, adjacent clusters are defined as the clusters with the closest group effects. This definition of adjacent clusters reflects the real-data situation. For example, a horror movie might be misclassified as a thriller movie, but less likely a romantic movie.

The simulation results based on 500 replications are summarized in Table 2. In general, the proposed method is robust against the misspecification of clusters. In comparison with the previous results from Table 1, the proposed method performs better than the other four methods in all settings even when 50% of the cluster members are misclassified. On the other hand, the misspecification rate affects the performance of the proposed method to different degrees for various settings of $\bar{\pi}$ and K. For example, the proposed method below the 50% misspecification rate is 2.7% worse than the proposed method when there is no misspecification, in terms of the RMSE under $K = 3$ and $\bar{\pi} = 0.7$; and becomes 18.8% worse than the one with no misspecification under $K = 6$ and $\bar{\pi} = 0.95$.

6. MovieLens Data

We apply the proposed method to MovieLens 1M and 10M data. The two datasets are collected by GroupLens Research and are available at *http://grouplens.org/datasets/movielens*. The MovieLens 1M data contain 1,000,209 ratings of 3883 movies by 6040 users, and rating scores range from 1 to 5. In addition, the 1M dataset provides demographic information for the users (age, gender, occupation, zipcode), and genres and release dates of the movies. In the MovieLens 10M data, we have 10,000,054 ratings collected from 71,567 users over 10,681 items, and 99% of

Table 1. RMSE (standard error) of the proposed method compared with four existing methods, with the missing rate $\bar{\pi} = 70\%, 80\%, 90\%,$ and 95%, and the number of latent factors $K = 3$ or 6, where RSVD, AC, MHT, and ZSY stand for regularized singular value decomposition, the regression-based latent factor model (Agarwal and Chen 2009), Soft-Impute (Mazumder et al. 2010), and the latent factor model with sparsity pursuit (Zhu et al. 2016), respectively.

No. of latent factors	Missing rate	The proposed method	RSVD	AC	MHT	ZSY
$K = 3$	70%	1.232 (0.029)	1.823 (0.324)	4.218 (0.089)	3.591 (0.178)	2.384 (0.077)
	80%	1.329 (0.042)	2.574 (0.506)	4.190 (0.091)	4.064 (0.140)	2.574 (0.085)
	90%	1.521 (0.070)	4.002 (0.689)	4.109 (0.095)	4.581 (0.116)	2.982 (0.095)
	95%	1.800 (0.103)	4.526 (0.172)	4.087 (0.096)	4.774 (0.123)	3.288 (0.100)
$K = 6$	70%	1.461 (0.035)	3.728 (0.188)	7.164 (0.132)	7.126 (0.294)	5.844 (0.656)
	80%	1.634 (0.058)	4.926 (0.274)	6.962 (0.134)	8.038 (0.267)	5.885 (0.145)
	90%	2.032 (0.136)	7.048 (0.270)	6.805 (0.136)	8.931 (0.172)	6.019 (0.420)
	95%	2.839 (0.388)	8.316 (0.270)	6.846 (0.149)	9.142 (0.176)	6.207 (0.151)

Table 2. RMSE (standard error) of the proposed method when the missing rate is 70%, 80%, 90%, or 95%, and the number of latent factors $K = 3$ or 6, under 0%, 10%, 30%, and 50% cluster misspecification rate.

No. of latent factors	Missing rate	Misspecification rate 0%	10%	30%	50%
$K = 3$	70%	1.232 (0.029)	1.237 (0.029)	1.250 (0.032)	1.265 (0.038)
	80%	1.329 (0.042)	1.340 (0.052)	1.359 (0.051)	1.380 (0.049)
	90%	1.521 (0.070)	1.544 (0.180)	1.591 (0.162)	1.626 (0.255)
	95%	1.800 (0.103)	1.810 (0.116)	1.869 (0.102)	1.920 (0.093)
$K = 6$	70%	1.461 (0.035)	1.502 (0.049)	1.560 (0.048)	1.623 (0.059)
	80%	1.634 (0.058)	1.698 (0.070)	1.815 (0.074)	1.911 (0.092)
	90%	2.032 (0.136)	2.229 (0.198)	2.428 (0.146)	2.648 (0.150)
	95%	2.839 (0.388)	3.041 (0.302)	3.245 (0.238)	3.373 (0.178)

the movie ratings are actually missing. Rating scores range from 0.5, 1, . . . , 5, but no user information is available.

Figure 1 illustrates the missing pattern of MovieLens 1M data. Both graphs indicate that the missing mechanism is possibly missing not at random. In the left figure, the right-skewed distribution from users indicates that only a few users rated a large number of movies. While the median number of ratings is 96, the maximum can reach up to 2314. The right figure shows that popular movies attract more viewers. That is, the number of ratings for each movie is positively associated with its average rating score, indicating nonignorable missingness.

For the proposed method, we take advantage of missingness information from each user and item for clustering. We observe that users who give a large number of ratings tend to assign low rating scores; therefore, we classify users based on the quantiles of the number of their ratings. For items, we notice that old movies being rated are usually classical and have high average rating scores. Therefore, the items are clustered based on their release dates. We use $N = 12$ and $M = 10$ as the number of clusters for users and items in both datasets. The means of ratings from different clusters are significantly different based on their pairwise two-sample t-tests. In addition, we also try a large range of N's and M's, but they do not affect the results very much.

The proposed method is compared with the four matrix factorization methods described in Section 5.1. Tuning parameters for each method are selected from grid points to minimize the RMSEs on the validation set. For the proposed method, we apply the loss function (2) and select $K = 2$ and $\lambda = 12$ for the 1M data, and $K = 6$ and $\lambda = 16$ for the 10M data. For Agarwal and Chen (2009), we select $K = 1$ for both the 1M and 10M data, which requires 25 and 10 iterations of the EM algorithm to guarantee convergence, respectively. For Mazumder et al. (2010), $K = 4$ and $K = 9$ are selected for the 1M and 10M data, and while using different λ's does not influence the RMSE very much, we apply $\lambda = 0$ to estimate the theoretical local minimum. For Zhu et al. (2016), the tuning and the selection of K are provided in their coding automatically, and the L_0-penalty function is applied. For the RSVD, $K = 4$ and $\lambda = 7.5$ are selected for the 1M data, and $K = 4$ and $\lambda = 6$ are selected for the 10M data. In addition, we also compare the proposed method with the "grand mean imputation" approach, which predicts each rating by the mean of the training set and the validation set, and the "linear regression" approach using ratings from the training and the validation sets against all available covariates from users and items.

Table 3 provides the prediction results on the testing set, which indicates that the proposed method outperforms the other methods quite significantly. For example, for the 1M data, the RMSE of the proposed method is 8.7% less than the RSVD, 19.5% less than Agarwal and Chen (2009), 10.3% less than Mazumder et al. (2010), 9.4% less than Zhu et al. (2016), and 13.2% and 11.6% less than grand mean imputation and

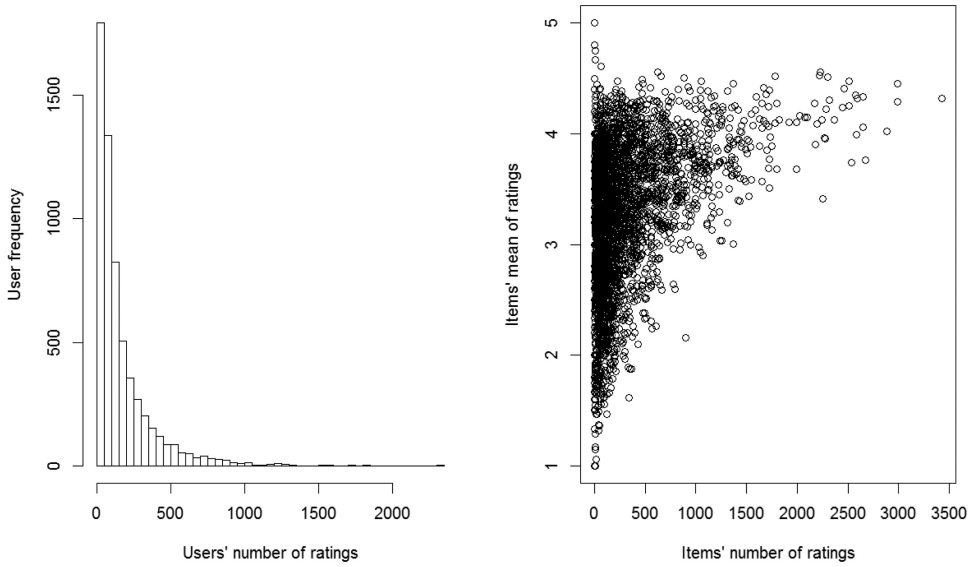

Figure 1. Missing pattern analysis for the MovieLens 1M data. Left: most users rated a small number of movies, while few users rated a large number of movies. Right: movies with a high average rating attract more users.

Table 3. RMSE of the proposed method compared with six existing methods for MovieLens 1M and 10M data, where RSVD, AC, MHT, and ZSY stand for regularized singular value decomposition, the regression-based latent factor model (Agarwal and Chen 2009), Soft-Impute (Mazumder et al. 2010), and the latent factor model with sparsity pursuit (Zhu et al. 2016), respectively.

	MovieLens 1M	MovieLens 10M
Grand Mean Imputation	1.1112	1.0185
Linear Regression	1.0905	1.0007
The Proposed Method	0.9635	0.9295
RSVD	1.0552	0.9966
AC	1.1974	0.9737
MHT	1.0737	1.0177
ZSY	1.0635	1.0108

linear regression, respectively. For the 10M data, the proposed method improves on grand mean imputation, linear regression, the RSVD, Agarwal and Chen (2009), Mazumder et al. (2010), and Zhu et al. (2016) by 8.7%, 7.1%, 6.7%, 4.5%, 8.7%, and 8.0% in terms of the RMSE, respectively. In addition, while some of the matrix factorization methods are worse than the linear regression method, the proposed method always beats the linear regression method.

The numerical studies are run on Dell C8220 computing sleds each with two 10-core Intel Xeon E5-2670V2 processors and 64GB RAM. The proposed method uses 0.8 minutes for 1M data ($K = 2$ and $\lambda = 12$), and 8.3 minutes for 10M data ($K = 6$ and $\lambda = 16$). The RSVD uses 6.4 minutes for 1M data ($K = 4$ and $\lambda = 7.5$), and 7.1 hours for 10M data ($K = 4$ and $\lambda = 6$). The Agarwal and Chen (2009) method requires 18.1 minutes for 1M data ($K = 1$ with 25 iterations), and 1.1 hours for 10M data ($K = 1$ with 10 iterations), while Mazumder et al. (2010) method uses 0.3 minutes for 1M data ($K = 4$), and 11.6 minutes for 10M data ($K = 9$), and Zhu et al. (2016) uses 1.1 minutes for 1M data, and 18.5 minutes for 10M data.

We also investigate the "cold-start" problem in the MovieLens 10M data, where 96% of the ratings in the testing set are either from new users or on new items which are not available in the training set. We name these ratings "new ratings," in contrast to the "old ratings" given by existing users to existing items. In Table 4, we compare the proposed method with the four competitive methods on the "old ratings," the "new ratings," and the entire testing set. On the one hand, the RSVD, Mazumder et al. (2010), Zhu et al. (2016) and the proposed method have similar RMSE for the "old ratings" set, indicating similar performances on prediction accuracy for existing users and items. On the other hand, the proposed method has the smallest RMSE compared to the other methods for the "new ratings" and the entire testing sets, indicating the superior performance of the proposed method for the "cold-start" problem.

Table 4. RMSE of the proposed method compared with four existing methods on the MovieLens 10M data to study the "cold-start" problem: "old ratings" and "new ratings" stand for ratings in the testing sets given by existing users to existing items, and by new users or to new items. Here, RSVD, AC, MHT, and ZSY stand for regularized singular value decomposition, the regression-based latent factor model (Agarwal and Chen 2009), Soft-Impute (Mazumder et al. 2010), and the latent factor model with sparsity pursuit (Zhu et al. 2016), respectively.

	The proposed method	RSVD	AC	MHT	ZSY
"Old ratings"	0.7971	0.8062	1.3324	0.8160	0.8018
"New ratings"	0.9348	1.0039	0.9553	1.0252	1.0189
The entire testing set	0.9295	0.9966	0.9737	1.0177	1.0108

7. Discussion

We propose a new recommender system which improves prediction accuracy through incorporating dependency among users and items, in addition to using information from the nonrandom missingness.

In most collaborative filtering methods, training data may not have sufficient information to estimate subject-specific parameters for new users and items. Therefore, only baseline models such as ANOVA or linear regression are applied. For example, for a new user u, $\hat{\mathbf{p}}_u = \mathbf{0}$, and a method without specifying the group effects has $\hat{\theta}_{ui} = \mathbf{x}'_{ui}\hat{\boldsymbol{\beta}}$. In contrast, the proposed method provides a prediction through $\hat{\theta}_{ui} = \mathbf{x}'_{ui}\hat{\boldsymbol{\beta}} + \hat{\mathbf{s}}'_{\upsilon_u}(\hat{\mathbf{q}}_i + \hat{\mathbf{t}}_{j_i})$. The interaction term $\hat{\mathbf{s}}'_{\upsilon_u}\hat{\mathbf{q}}_i$ provides the average rating of the υ_uth cluster on the ith item, which guarantees that $\hat{\theta}_{ui}$ is item-specific. The same property also holds for new items. The group effects $\mathbf{s}_{\upsilon_u}$ and $\mathbf{t}_{j_i}$ allow us to borrow information from existing users and items, and provide more accurate recommendations to new subjects.

The proposed model also takes the advantage of missingness information as users or items may have missing patterns associated with their rating behaviors. Therefore, we propose clustering users and items based on the numbers of their ratings or other variables associated with the missingness. Thus, the group effects $(\mathbf{s}_{\upsilon_u}, \mathbf{t}_{j_i})$ could provide unique latent information which are not available in $\mathbf{x}_{ui}$, $\mathbf{p}_u$ or $\mathbf{q}_i$. Note that if the group effects $(\mathbf{s}_{\upsilon_u}, \mathbf{t}_{j_i})$ are the only factors that are associated with the missing process, then the proposed method captures the entire missing-not-at-random mechanism. In other words, correctly estimating $(\mathbf{s}_{\upsilon_u}, \mathbf{t}_{j_i})$ enables us to achieve consistent and efficient estimation of θ_{ui}, regardless of the missing mechanism.

One possible future research direction is to bridge the gap between numerical implementation and theoretical justification, where the current numerical algorithm converges to a stationary point, while the theoretical properties are established for the global minimum. In general, this gap occurs in nonconvex optimization problems (e.g., Zhou et al. 2013; Zhu et al. 2016), since nonconvex optimization is NP-hard (Ge et al. 2015). In the global optimization literature, methods ensuring global solutions exist but may require extremely high computational cost; for instance, branch-and-bound methods (Land and Doig 1960; Liu et al. 2005). The most relevant approach is the outer approximation method proposed by Breiman and Cutler (1993), which guarantees a global minimizer but may suffer a slow convergence. This could make it infeasible to solve large-scale problems. In addition, further investigation is needed regarding the convergence behavior of parameters under the L_2 distance.

Supplementary Materials

The online supplement contains proofs for Lemmas 1 and 2, Theorems 1 and 2, and Corollarys 1 and 2.

Acknowledgments

The authors thank Yunzhang Zhu for providing the program code for Zhu et al. (2016)'s method, and the editor, associate editor, and two reviewers for insightful comments, and suggestions which improve the article significantly.

ORCID

Annie Qu http://orcid.org/0000-0002-8396-7828

References

Adomavicius, G. and Tuzhilin, A. (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, 734–749. [1344]

Agarwal, D. and Chen, B.-C. (2009), "Regression-Based Latent Factor Models," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 19–28. [1345,1350,1351,1352]

Ansley, C. F. and Kohn, R. (1994), "Convergence of the Backfitting Algorithm for Additive Models," *Journal of the Australian Mathematical Society*, Series A, 57, 316–329. [1347]

Bell, R. M. and Koren, Y. (2007), "Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights," *Proceedings of the 2007 7th IEEE International Conference on Data Mining*, Piscataway, NJ: IEEE, pp. 43–52. [1344]

Blanco-Fernandez, Y., Pazos-Arias, J. J., Gil-Solla, A., Ramos-Cabrer, M., and Lopez-Nores, M. (2008), "Providing Entertainment by Content-Based Filtering and Semantic Reasoning in Intelligent Recommender Systems," *IEEE Transactions on Consumer Electronics*, 54, 727–735. [1344]

Breiman, L. and Cutler, A. (1993), "A Deterministic Algorithm for Global Optimization," *Mathematical Programming*, 58, 179–199. [1352]

Breiman, L. and Friedman, J. H. (1985), "Estimating Optimal Transformations for Multiple Regression and Correlation," *Journal of the American Statistical Association*, 80, 580–598. [1346]

Cacheda, F., Carneiro, V., Fernández, D., and Formoso, V. (2011), "Comparison of Collaborative Filtering Algorithms: Limitations of Current Techniques and Proposals for Scalable, High-Performance Recommender Systems," *ACM Transactions on the Web (TWEB)*, 5, 2. [1344]

Carroll, J. D. and Chang, J.-J. (1970), "Analysis of Individual Differences in Multidimensional Scaling Via an N-way Generalization of 'Eckart-Young' Decomposition," *Psychometrika*, 35, 283–319. [1345]

Chen, B., He, S., Li, Z., and Zhang, S. (2012), "Maximum Block Improvement and Polynomial Optimization," *SIAM Journal on Optimization*, 22, 87–107. [1347]

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. (2004), "Least Angle Regression," *The Annals of Statistics*, 32, 407–499. [1347]

Feuerverger, A., He, Y., and Khatri, S. (2012), "Statistical Significance of the Netflix Challenge," *Statistical Science*, 27, 202–231. [1344]

Funk, S. (2006), "Netflix Update: Try this at Home," available at *http://sifter.org/ simon/journal/20061211.html*. [1344,1345,1350]

Ge, R., Huang, F., Jin, C., and Yuan, Y. (2015), "Escaping from Saddle Points–Online Stochastic Gradient for Tensor Decomposition," *Proceedings of The 28th Conference on Learning Theory*, pp. 797–842. [1345,1352]

Goldberg, K., Roeder, T., Gupta, D., and Perkins, C. (2001), "Eigentaste: A Constant Time Collaborative Filtering Algorithm," *Information Retrieval*, 4, 133–151. [1345]

Grenander, U. (1981), *Abstract Inference*. New York: Wiley. [1349]

Harshman, R. A. (1970), "Foundations of the Parafac Procedure: Models and Conditions for an 'Explanatory' Multi-modal Factor Analysis," *UCLA Working Papers in Phonetics*, 16, 1–84. [1345]

Koren, Y. (2010), "Factor in the Neighbors: Scalable and Accurate Collaborative Filtering," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 4, 1. [1345]

Koren, Y., Bell, R., and Volinsky, C. (2009), "Matrix Factorization Techniques for Recommender Systems," *Computer*, 42, 30–37. [1345,1350]

Land, A. H. and Doig, A. G. (1960), "An Automatic Method of Solving Discrete Programming Problems," *Econometrica*, 28, 497–520. [1352]

Lang, K. (1995), "Newsweeder: Learning to Filter Netnews," *Proceedings of the 12th International Conference on Machine Learning*, pp. 331–339. [1344]

Liu, S., Shen, X., and Wong, W. H. (2005), "Computational Developments of ψ-Learning," *Proceedings 5th SIAM International Conference on Data Mining, Newport Beach, CA*, 1–12. Philadelphia, PA: SIAM. [1347,1352]

Lops, P., De Gemmis, M., and Semeraro, G. (2011), "Content-Based Recommender Systems: State of the Art and Trends," *Recommender Systems Handbook*, New York: Springer, pp. 73–105. [1344]

Mazumder, R., Hastie, T., and Tibshirani, R. (2010), "Spectral Regularization Algorithms for Learning Large Incomplete Matrices," *The Journal of Machine Learning Research*, 11, 2287–2322. [1344,1345,1350,1351,1352]

Melville, P., Mooney, R. J., and Nagarajan, R. (2002), "Content-Boosted Collaborative Filtering for Improved Recommendations," *Proceedings of the 18th National Conference on Artificial Intelligence*, pp. 187–192. [1345]

Mooney, R. J. and Roy, L. (2000), "Content-Based Book Recommending Using Learning for Text Categorization," *Proceedings of the 5th ACM Conference on Digital Libraries*, San Antonio, TX: ACM, pp. 195–204. [1344]

Nguyen, A.-T., Denos, N., and Berrut, C. (2007), "Improving New User Recommendations with Rule-Based Induction on Cold User Data," in *Proceedings of the 2007 ACM Conference on Recommender Systems*, 121–128. ACM. [1345]

Nguyen, J., and Zhu, M. (2013), "Content-Boosted Matrix Factorization Techniques for Recommender Systems," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 6, 286–301. [1345]

Park, S.-T., Pennock, D., Madani, O., Good, N., and DeCoste, D. (2006), "Naïve Filterbots for Robust Cold-Start Recommendations," *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Antonio, TX: ACM, pp. 699–705. [1345]

Pazzani, M. J. and Billsus, D. (2007), "Content-Based Recommendation Systems," *The Adaptive Web*, New York: Springer, pp. 325–341. [1344]

Pollard, D. (2012), *Convergence of Stochastic Processes*, New York: Springer Science & Business Media. [1349]

Rubin, D. B. (1976), "Inference and Missing Data," *Biometrika*, 63, 581–592. [1350]

Salakhutdinov, R., Mnih, A., and Hinton, G. (2007), "Restricted Boltzmann Machines for Collaborative Filtering," *Proceedings of the 24th International Conference on Machine Learning*, San Antonio, TX: ACM, pp. 791–798. [1344]

Shen, X. (1998), "On the Method of Penalization," *Statistica Sinica*, 8, 337–357. [1348,1349]

Shen, X. and Wong, W. H. (1994), "Convergence Rate of Sieve Estimates," *Annals of Statistics*, 22, 580–615. [1349]

Srebro, N., Alon, N., and Jaakkola, T. S. (2005), "Generalization Error Bounds for Collaborative Prediction with Low-Rank Matrices," in *Advances in Neural Information Processing Systems*, 17, 5–27. [1348]

Van de Geer, S. (1993), "Hellinger-Consistency of Certain Nonparametric Maximum Likelihood Estimators," *The Annals of Statistics*, 21, 14–44. [1348]

Wang, J. (2010), "Consistent Selection of the Number of Clusters Via Crossvalidation," *Biometrika*, 97, 893–904. [1347]

Wong, W. H. and Shen, X. (1995), "Probability Inequalities for Likelihood Ratios and Convergence Rates of Sieve MLEs," *The Annals of Statistics*, 23, 339–362. [1349]

Wu, M. (2007), "Collaborative Filtering Via Ensembles of Matrix Factorizations," *Proceedings of KDD Cup and Workshop*. [1345]

Zhou, H., Li, L., and Zhu, H. (2013), "Tensor Regression with Applications in Neuroimaging Data Analysis," *Journal of the American Statistical Association*, 108, 540–552. [1345,1352]

Zhu, Y., Shen, X., and Ye, C. (2016), "Personalized Prediction and Sparsity Pursuit in Latent Factor Models," *Journal of the American Statistical Association*, 111, 241–252. [1345,1347,1350,1351,1352]