Journal of Machine Learning Research 20 (2011) 1-24 Submitted 10/17; Published 2/19

Smooth neighborhood recommender systems

Ben Dai BENDAI2-C@MY.CITYU.EDU.HK
Junhui Wang J.H.WANG@CITYU.EDU.HK
School of Data Science
City University of Hong Kong
Kowloon Tong, 999077, Hong Kong

Xiaotong Shen XSHEN@UMN.EDU
School of Statistics
University of Minnesota
Minneapolis, MN 55455, USA

Annie Qu ANNIEQU@ILLINOIS.EDU
Department of Statistics
University of Illinois at Urbana Champaign
Champaign, IL 61820, USA

Editor: Gert Lanckriet

Abstract

Recommender systems predict users' preferences over a large number of items by pooling similar information from other users and/or items in the presence of sparse observations. One major challenge is how to utilize user-item specific covariates and networks describing user-item interactions in a high-dimensional situation, for accurate personalized prediction. In this article, we propose a smooth neighborhood recommender in the framework of the latent factor models. A similarity kernel is utilized to borrow neighborhood information from continuous covariates over a user-item specific network, such as a user's social network, where the grouping information defined by discrete covariates is also integrated through the network. Consequently, user-item specific information is built into the recommender to battle the 'cold-start" issue in the absence of observations in collaborative and content-based filtering. Moreover, we utilize a "divide-and-conquer" version of the alternating least squares algorithm to achieve scalable computation, and establish asymptotic results for the proposed method, demonstrating that it achieves superior prediction accuracy. Finally, we illustrate that the proposed method improves substantially over its competitors in simulated examples and real benchmark data–*Last.fm* music data.

Keywords: Blockwise coordinate decent, Cold-start, Kernel smoothing, Neighborhood, Personalized prediction, Singular value decomposition, Social networks.

1. Introduction

Recommender systems predict users' preferences over a large number of items by pooling similar information from other users or items when observations are sparse, which are particularly useful in personalized prediction. It has become an essential part of e-commerce, with applications in movie rentals (MovieLens; Miller et al. 2003), restaurant guides (Entree;

Burke 2002), book recommendations (Amazon; Linden et al. 2003), and personalized e-news (Daily learner; Billsus and Pazzani 2000).

Two main streams emerge for training recommender systems: collaborative filtering, which predicts users' behaviors based on similar users (Bell and Koren, 2007), and content-based filtering, which builds user and item profiles based on domain knowledge and recommends items with similar profiles (Lang, 1995; Melville et al., 2002). Specifically, collaborative filtering predicts unknown ratings by averaging over similar users' ratings with weights; such as the latent factor approach (Feuerverger et al., 2012), latent Dirichlet allocation (LDA; Blei et al. 2003), probabilistic latent semantic analysis (pLSA; Hofmann 2004), regularized singular value decomposition (regularized SVD; Paterek 2007), and restricted Boltzmann machines (RBM; Salakhutdinov et al. 2007). Among them, the regularized SVD approach has become popular due to its high predictive performance and scalability in real applications. In addition, Koren (2008) further generalizes SVD to model users' implicit feedbacks, and Forbes and Zhu (2011) incorporates content information in the regularized SVD approach through a regression-type of constraint. For content-based filtering, keywords analysis extracts features from items previously rated by a user to develop a profile of the user's interests, and recommendation is made by comparing the user profile and potential items (Lang, 1995). The naive Bayes (Billsus and Pazzani, 2000), decision tree (Pazzani et al., 1996) and kNN (Middleton et al., 2004) formulate this type of recommendation as a classification problem, where each item can be labeled by users as "like" or "dislike". Hybrid recommender systems (Burke, 2002) utilize geo-social correlations to accommodate new users and items through location-based recommendation systems; Bi et al. (2017) proposes a group-specific latent factor model by utilizing missingness-related characteristics to accommodate new users or items without any observed ratings.

To achieve better prediction accuracy, several main challenges remain in training recommender systems. First, smooth structure is contained in the user-item interactions, yet they are not fully utilized in model training. For instances, friends tend to share similar interests and preferences on various items, and relevant items tend to receive comparable ratings from users. Particularly, Figure 1 for the *Last.fm* data indicates that friends behave more similarly than randomly selected user pairs, which can be well characterized by a smoothing pattern of users' preference on items over the user-item specific network. However, such a user-item specific network usually involves a large number of parameters, which imposes great challenges in model building without a sufficient amount of observations. Therefore, how to incorporate the user-item specific network into a recommender system remains one key factor for improving prediction accuracy. Second, covariate information such as demographic and social network information of users, tags and content information of items, can be easily accessible (Nguyen and Zhu, 2013). In the existing literature, a number of methods have been proposed to utilize the side and relational information through Laplacian regularization (Gu et al., 2010), Gaussian generative models (Zhou et al., 2012), probability propagation (Yang et al., 2011), aggregating regression (Demir et al., 2017; Zhao et al., 2016; Zhao and Guo, 2017), Markov logic network (Salakhutdinov and Mnih, 2008; Richardson and Domingos, 2006), and the probabilistic soft logic (Kouki et al., 2015; Bach et al., 2017). Yet, most of the aforementioned methods are rather ad hoc and lack theoretical justification. Mainly, it remains largely unknown how the users' social networks and items' tagging information impact the accuracy of prediction. More importantly, a rating mechanism usually does not follow a parametric form in terms of covariates, which is commonly assumed in the literature due to the lack of sufficient observations for each user and item. How to employ a nonparametric approach to model the covariate-assisted recommender system in a high-dimensional setting continues to be an open question. Third, missing completely at random (MCAR) is often assumed by existing methods, leading to inaccurate prediction as the MCAR assumption is typically unrealistic for recommender systems. For example, users tend not to rate items that are of little interest to them, as illustrated in Figure 4. Fourth, most methods fail to recommend for new users or items without any observed ratings, which is referred as the "cold-start" problem. Thus, utilizing the covariate information to fully and efficiently solve the "cold-start" problem is attractive in devising recommender systems.

In this paper, we propose a novel approach based on the idea of a similarity-based neighborhood system pooling similar user-item pairs to improve prediction performance. Specifically, for each user-item pair, the proposed approach incorporates similar observed pairs through kernel weighting based on covariates as well as a user-item specific network. The weight function quantifies a smooth rating mechanism in terms of the closeness of continuous covariates within a neighborhood of the connected user-item pairs in the network. One novelty of the proposed approach is that it builds discrete covariates into a user-item specific network with the same discrete covariate values corresponding to connectivity. This enables us to handle high-dimensional covariates while not being burdened by the "curse of dimensionality." Unlike existing methods (Zhu et al., 2016; Bi et al., 2017), our approach is nonparametric, yet it goes beyond the traditional nonparametric framework which focuses primarily on continuous covariates. This provides a flexible framework to handle continuous covariates, discrete covariates and networks all together without specifying a functional relation. Moreover, the proposed approach also tackles the "cold-start" issue as it utilizes observed pairs in the neighborhood of any new user-item pairs in prediction.

The proposed approach takes full advantage of the smoothing pattern of the rating mechanism over covariates, while integrating user-item dependencies through user-item specific networks into a recommender system nonparametrically. This leads to a higher prediction accuracy for a recommender, as demonstrated in the numerical examples in Section 5. Significantly, our approach outperforms the state-of-the-art prediction performance for the *Last.fm* music benchmark dataset by nearly 20%. Additionally, we perform an error analysis, showing that the error rate of the proposed method is governed by the degree of smoothness of a neighborhood system with respect to continuous covariates, given that the grouping information is precisely defined by discrete covariates and networks. Most critically, as suggested by Theorem 1, the method performs well even in a high-dimensional situation in which the overall size of observed ratings is of the same magnitude as the number of unknown parameters.

The rest of the paper is organized as follows. Section 2 briefly introduces the regularized latent factor model, Section 3 presents the proposed recommender system and its implementation via the alternating least squares algorithm. Section 4 establishes the asymptotic results for the proposed method. Section 5 examines the numerical performance of the proposed method in simulation studies and a real application to the *Last.fm* dataset (`http://www.last.fm`). A brief summary is given in Section 6, and the Appendix contains the technical proofs.

2. Regularized latent factors

In this section, we provide the notations in recommender systems, and the framework of a regularized latent factor model. Consider a recommender system of n users' preference scores on m items, where r_{ui} denotes the preference score of user u on item i. Suppose a user-item specific covariate vector $\boldsymbol{x}_{ui} \in \mathcal{X} \subset \mathbb{R}^d$ is observed (e.g., a user's demographics and an item's content information). One key challenge for training a recommender system is that the preference matrix $R = (r_{ui}) \in \mathbb{R}^{n \times m}$ is only partially observed with a high missing percentage. Denote the index set of observed preference scores as Ω, then $|\Omega| \ll nm$.

A recommender system can be formulated in the framework of a latent factor model:

$$r_{ui} = \theta_{ui} + \epsilon_{ui} = \boldsymbol{p}_u^T \boldsymbol{q}_i + \varepsilon_{ui}, \quad 1 \leq u \leq n, 1 \leq i \leq m, \tag{1}$$

where $\theta_{ui} = E(r_{ui})$ is the expected preference score of a user-item pair (u, i), and ε_{ui} is independent from $\boldsymbol{x}_{ui}$ with mean zero and finite variance. The latent factor model assumes that θ_{ui} can be represented by user and item latent factors: $\theta_{ui} = \boldsymbol{p}_u^T \boldsymbol{q}_i$, where $\boldsymbol{p}_u$ and $\boldsymbol{q}_i$ are K-dimensional latent vectors representing user u's preference and item i's profile, respectively, and K is the number of latent factors for both users and items, which is also the rank in the latent factor model (Mukherjee et al., 2015).

To estimate these personalized parameters, a regularized SVD method (Paterek, 2007; Zhu et al., 2016) estimates $\boldsymbol{P} = (\boldsymbol{p}_1, \cdots, \boldsymbol{p}_n)^T$ and $\boldsymbol{Q} = (\boldsymbol{q}_1, \cdots, \boldsymbol{q}_m)^T$ by

$$\min_{\boldsymbol{P}, \boldsymbol{Q}} \frac{1}{|\Omega|} \sum_{(u,i) \in \Omega} (r_{ui} - \boldsymbol{p}_u^T \boldsymbol{q}_i)^2 + \lambda \Big\{ \sum_{u=1}^{n} J(\boldsymbol{p}_u) + \sum_{i=1}^{m} J(\boldsymbol{q}_i) \Big\}, \tag{2}$$

where λ is a nonnegative tuning parameter, and $J(\cdot)$ can be any penalty function such as the L_q-penalty with $q = 0, 1, 2$ (Zhu et al., 2016) or the alignment penalty (Nguyen and Zhu, 2013). Note that r_{ui} in (2) is often replaced by the residual $r_{ui} - \mu - \boldsymbol{x}_{ui}^T \boldsymbol{\beta}$, where $(\mu, \boldsymbol{\beta})$ is a vector of regression coefficients to be minimized in (2). Alternatively, $\boldsymbol{p}_u = \boldsymbol{s}_u - \boldsymbol{x}_u^T \boldsymbol{\alpha}$ and $\boldsymbol{q}_i = \boldsymbol{t}_i - \boldsymbol{x}_i^T \boldsymbol{\beta}$, where $(\boldsymbol{s}_u, \boldsymbol{t}_i)$ are latent factors, and $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ are regression coefficients to incorporate the covariate effects (Agarwal et al., 2011).

There are a number of major challenges for the regularized SVD model in (2). First, a user-item specific network is often available to capture user-item dependence but is ignored in (2). Here networks consists of information from existing users' social network, or item network, or the network constructed from available discrete covariates of the user-item pair (u, i). It can provide additional information regarding the preference similarity between connected pairs. This is the case for the *Last.fm* data, where a user specific network impacts users' preference on items, indicated by Figure 1. Second, a linear model in (2) may be inadequate to incorporate user and item covariates, especially when the linearity assumption is violated. Third, the objective function (2) assumes implicitly that missing occurs completely at random so that the first loss function in (2) can approximate the loss function for the complete data. When the missing is informative, missing characteristics such as the missing pattern for each user can be utilized for subgrouping (Bi et al., 2017). Fourth, the regularized SVD in (2) suffers from the "cold-start" problem for new users or items in the absence of observed ratings. For instance, if a user u or an item i is completely missing from Ω, the corresponding $\boldsymbol{p}_u$ or $\boldsymbol{q}_i$ is estimated incorrectly as 0 due to the penalty $J(\cdot)$ in (2).

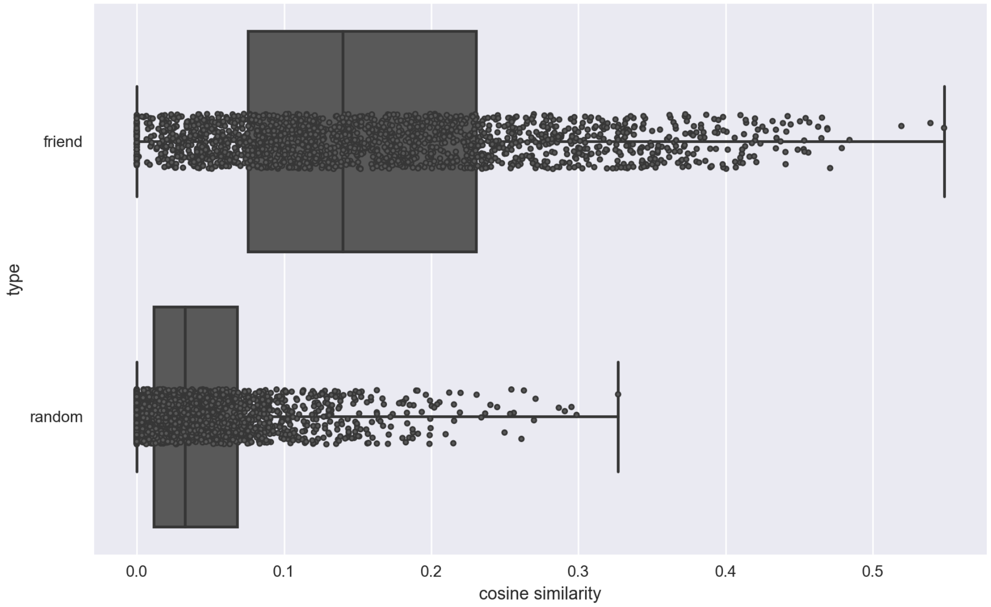


Figure 1: Comparison of the cosine similarity of logarithm of users' listening counts vector between friends and randomly selected users in the *Last.fm* dataset.

3. Methods

This section introduces a smooth recommender system, which incorporates user and/or item specific covariates and the user-item specific network information to improve prediction accuracy. The proposed method utilizes informative observations for each user-item pair for personalized prediction and resolving the "cold-start" problem.

3.1. Proposed method

One key strategy of the proposed method is to pool information across user-item pairs to improve prediction accuracy through increasing effective sample size. In contrast to grouping approaches based on user-item-specific information (Bi et al., 2017; Masthoff, 2011), the proposed recommender system integrates similar observed pairs $(u', i') \in \Omega$ for each (u, i) through a weight function, regardless of whether (u, i) is observed or not. This allows for nonlinear or nonparametric modeling of the relation between user-item preferences and latent factors, which is more flexible than the linear modeling in (2).

The proposed weight function is constructed based on the closeness between continuous covariates in addition to a user-item specific network. To precisely describe the network structure, we define an indicator $S^{ui}_{u'i'} = S^{u}_{u'} S^{i}_{i'} = 1$ if there exist edges connecting u and u' in a user-specific social network and i and i' in an item-specific network, and $S^{ui}_{u'i'} = 0$ otherwise. Moreover, we integrate the above user-item specific network with existing user social network, item network, as well as information from discrete covariates. For example, in the *Last.fm* dataset, r_{ui} is the listening count of a user-artist pair (u, i), and $\boldsymbol{x}_{ui}$ is the pattern of listening counts for user u and artist i. The available network information is given as $S^{ui}_{u'i'} = S^u_{u'} S^i_{i'}$, where $S^u_{u'}$ represents a user's friendship network, and $S^i_{i'}$ is constructed from artists' tag information with $S^i_{i'} = 1$ indicating artists i and i' share the same tags, and $S^i_{i'} = 0$ otherwise.

Using local likelihood via smoothing weights (Tibshirani and Hastie, 1987; Fan and Gijbels, 1996), we propose the following cost function for a smooth neighborhood recommender system:

$$\mathcal{L}(\boldsymbol{P}, \boldsymbol{Q}) = \frac{1}{nm} \sum_{u=1}^{n} \sum_{i=1}^{m} \Big(\sum_{(u',i') \in \Omega} \omega_{ui,u'i'} (r_{u'i'} - \boldsymbol{p}_u^T \boldsymbol{q}_i)^2 \Big) + \lambda_1 \sum_{u=1}^{n} J(\boldsymbol{p}_u) + \lambda_2 \sum_{i=1}^{m} J(\boldsymbol{q}_i), \quad (3)$$

where $J(\cdot)$ is a general penalty, and λ_1 and λ_2 are two nonnegative tuning parameters. It is also assumed that $\sum_{(u',i') \in \Omega} \omega_{ui,u'i'} = 1$. Lemma 1 below gives an equivalent form of (3).

Lemma 1 *The cost function $\mathcal{L}(\boldsymbol{P}, \boldsymbol{Q})$ in (3) is equivalent to*

$$\frac{1}{nm} \sum_{u=1}^{n} \sum_{i=1}^{m} \Big(\sum_{(u',i') \in \Omega} \omega_{ui,u'i'} r_{u'i'} - \boldsymbol{p}_u^T \boldsymbol{q}_i \Big)^2 + \lambda_1 \sum_{u=1}^{n} J(\boldsymbol{p}_u) + \lambda_2 \sum_{i=1}^{m} J(\boldsymbol{q}_i). \quad (4)$$

The choice of $\omega_{ui,u'i'}$ is critical to properly measure the similarity between (u, i) and (u', i'). We introduce a weight function to define the smooth neighborhood of (u, i):

$$\omega_{ui,u'i'} = \frac{\mathcal{K}_h(\boldsymbol{x}_{ui}, \boldsymbol{x}_{u'i'}) S^{ui}_{u'i'}}{\sum_{(u',i') \in \Omega} \mathcal{K}_h(\boldsymbol{x}_{ui}, \boldsymbol{x}_{u'i'}) S^{ui}_{u'i'}}, \quad (5)$$

where $\omega_{ui,u'i'}$ involves only observed pairs (u', i')'s with $S^{ui}_{u'i'} = 1$. In (5), if $\boldsymbol{x}_{ui}$ or $S^{ui}_{u'i'}$ is absent, $\mathcal{K}_h(\boldsymbol{x}_{ui}, \boldsymbol{x}_{u'i'})$ or $S^{ui}_{u'i'}$ can be set as 1 correspondingly. The kernel function is set as $\mathcal{K}_h(\boldsymbol{x}_{ui}, \boldsymbol{x}_{u'i'}) = \mathcal{K}(h^{-1} \|\boldsymbol{x}_{ui} - \boldsymbol{x}_{u'i'}\|_2)$ measuring the closeness between $\boldsymbol{x}_{ui}$ and $\boldsymbol{x}_{u'i'}$, where the choice of the L_2-norm reduces the dimension of the covariate space and other choices of distance may be also considered. Here $h > 0$ is the window size and $\mathcal{K}(\cdot)$ is a kernel whose degree of smoothness reflects prior knowledge about how the true preference varies in terms of $\|\boldsymbol{x}_{ui} - \boldsymbol{x}_{u'i'}\|_2$. Note that this is different from standard kernel smoothing (Fan and Gijbels, 1996; Delaigle and Hall, 2010), in that the smooth neighborhood is constructed based on continuous and discrete covariates as well as user-item specific networks, whereas standard kernel smoothing focuses primarily on smoothing over continuous covariates.

The proposed framework has the following advantages. First, the user-item specific covariates and network structures are integrated in constructing the neighborhood for (u, i) pairs. Thus the effective sample size for (u, i) increases when pooling information from its neighborhood through $\sum_{(u',i') \in \Omega} \omega_{ui,u'i'} r_{u'i'}$ in (4). Second, it solves the "cold-start" problem and yields more accurate estimators of $\boldsymbol{p}_u$ and $\boldsymbol{q}_i$ for all u's and i's by leveraging dependencies among users and items, expressed in terms of user-specific social networks and items' tagging information. This is evident from (4), since even for an unobserved (u, i), $\sum_{(u',i') \in \Omega} \omega_{ui,u'i'} r_{u'i'}$ is a weighted average of preference scores over its neighborhood. On this ground, a recommendation can be made by estimating $\boldsymbol{p}_u$ and $\boldsymbol{q}_i$ for any new user-item pair (u, i). Third, the non-ignorable missing can be addressed through a covariate-adjusted neighborhood associated with missingness and users' preferences. For instance, each user's and item's missing percentages and the percentiles of their observed ratings can be modeled nonparametrically as covariates in defining $\omega_{ui,u'i'}$.

3.2. Scalable computation

To solve large-scale optimization in (3), we employ a "divide-and-conquer" type of alternating least squares (ALS) algorithm, with a principle of solving many small penalized regression problems iteratively. This permits parallel and efficient computation. The ALS method has been extensively investigated in the literature (Carroll and Chang, 1970; Friedman and Stuetzle, 1981), and the divide-and-conquer strategy is also employed in (Zhu et al., 2016) for the parallelization of (2).

The computational strategy of ALS is to break large-scale optimization into multiple small subproblems by alternatively fixing either $\boldsymbol{p}_u$ or $\boldsymbol{q}_i$, where each subproblem is a simple penalized least squares regression and can be solved analytically with $J(\cdot) = \|\cdot\|_2^2$. Note that this strategy is applicable as long as $J(\cdot)$ is separable for $\boldsymbol{p}_u$ and $\boldsymbol{q}_i$.

For illustration, consider $J(\cdot) = \|\cdot\|_2^2$. At iteration k, $\widehat{\boldsymbol{Q}}^{(k)}$ is fixed and the latent factor $\boldsymbol{p}_u$ is updated as $\hat{\boldsymbol{p}}_u^{(k+1)} = \text{argmin}_{\boldsymbol{p}_u} \sum_i \Big(\sum_{(u',i')\in\Omega} \omega_{ui,u'i'}(r_{u'i'} - \boldsymbol{p}_u^T \hat{\boldsymbol{q}}_i^{(k)})^2 \Big) + \lambda_1 \|\boldsymbol{p}_u\|_2^2$. Similarly, with fixed $\widehat{\boldsymbol{P}}^{(k+1)}$, $\boldsymbol{q}_i$ is updated as $\hat{\boldsymbol{q}}_i^{(k+1)} = \text{argmin}_{\boldsymbol{q}_i} \sum_u \Big(\sum_{(u',i')\in\Omega} \omega_{ui,u'i'}(r_{u'i'} - (\hat{\boldsymbol{p}}_u^{(k+1)})^T \boldsymbol{q}_i)^2 \Big) + \lambda_2 \|\boldsymbol{q}_i\|_2^2$. Then each subproblem is solved analytically,

$$\hat{\boldsymbol{p}}_u^{(k+1)} = \Big(\sum_i \hat{\boldsymbol{q}}_i^{(k)} (\hat{\boldsymbol{q}}_i^{(k)})^T + \lambda_1 \boldsymbol{I}_K \Big)^{-1} \Big(\sum_i \bar{r}_{ui}^{\Omega} \hat{\boldsymbol{q}}_i^{(k)} \Big), \tag{6}$$

$$\hat{\boldsymbol{q}}_i^{(k+1)} = \Big(\sum_u \hat{\boldsymbol{p}}_u^{(k+1)} (\hat{\boldsymbol{p}}_u^{(k+1)})^T + \lambda_2 \boldsymbol{I}_K \Big)^{-1} \Big(\sum_u \bar{r}_{ui}^{\Omega} \hat{\boldsymbol{p}}_u^{(k+1)} \Big), \tag{7}$$

where $\bar{r}_{ui}^{\Omega} = \sum_{(u',i')\in\Omega} \omega_{ui,u'i'} r_{u'i'}$ is a weighted rating for (u, i) over the neighborhood, and $\boldsymbol{I}_K$ is a $K \times K$ identity matrix. The iterative updating is continued until a termination criterion is reached. Once the solution $\{\hat{\boldsymbol{p}}_u, \hat{\boldsymbol{q}}_i\}_{1\le u\le n; 1\le i\le m}$ is obtained, the final predicted preference is $\hat{r}_{ui} = \hat{\boldsymbol{p}}_u^T \hat{\boldsymbol{q}}_i$.

It follows from Chen et al. (2012) that the algorithm converges to a stationary point $(\bar{\boldsymbol{P}}, \bar{\boldsymbol{Q}})$ of $\mathcal{L}(\boldsymbol{P}, \boldsymbol{Q})$ in (3), where $\bar{\boldsymbol{P}} = \text{argmin}_{\boldsymbol{P}} \mathcal{L}(\boldsymbol{P}, \bar{\boldsymbol{Q}})$, and $\bar{\boldsymbol{Q}} = \text{argmin}_{\boldsymbol{Q}} \mathcal{L}(\bar{\boldsymbol{P}}, \boldsymbol{Q})$. This is due to the nonconvex minimization in addition to having many missing observations. Moreover, each update of $\boldsymbol{p}_u$ and $\boldsymbol{q}_i$ in (6) and (7) can be computed in a parallel fashion. This can substantially speed up the computation, particularly when K is small but m and n are large.

The overall computational complexity of the algorithm is no greater than $O\big((nmK^2 + (n+m)K^3)I_{ALS}\big)$, where I_{ALS} is the number of iterations in the ALS algorithm. In our implementation, the algorithm is coded through PyMP, which is a Python version of OpenMP in C, and can handle a dataset with a size up to the order of 10^8 on a quad-core computer with one 3.40GHz CPU and 8G memory.

4. Theory

This section presents theoretical properties to quantify the asymptotic behavior of the proposed recommender system. Particularly, we show the convergence rate in prediction error of the proposed system.

Let the true and estimated parameters be $\theta^0_{ui} = {\boldsymbol{p}^0_u}^T \boldsymbol{q}^0_i$ and $\hat{\theta}_{ui} = \hat{\boldsymbol{p}}^T_u \hat{\boldsymbol{q}}_i$, where $\hat{\boldsymbol{p}}_u$ and $\hat{\boldsymbol{q}}_i$ are estimated latent factors; $u = 1, \cdots, n$; $i = 1, \cdots, m$. Note that the representation is unique with respect to θ_{ui}, although the decomposition of $\boldsymbol{p}_u$ and $\boldsymbol{q}_i$ may not be unique. Let $\hat{\boldsymbol{\theta}} = (\hat{\theta}_{ui}) \in \mathbb{R}^{n\times m}$ and $\boldsymbol{\theta}^0 = (\theta^0_{ui}) \in \mathbb{R}^{n\times m}$, the prediction accuracy of $\hat{\boldsymbol{\theta}}$ is defined by the root mean square error:

$$\mathrm{RMSE}(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^0) = \Big(\frac{1}{nm}\sum_{u=1}^{n}\sum_{i=1}^{m}\big(\hat{\theta}_{ui} - \theta^0_{ui}\big)^2\Big)^{1/2}.$$

We require the following technical assumptions.

Assumption A. There exist constants $c_1 > 0$ and $\alpha > 0$ such that for any (u, i) and (u', i') , $\big|\theta^0_{ui} - \theta^0_{u'i'}\big| \le c_1\sqrt{K}\max\{\|\boldsymbol{x}_{ui} - \boldsymbol{x}_{u'i'}\|_2^\alpha, I(S^{ui}_{u'i'} = 0)\}$, where the corresponding expression in the maximum operator is set as 0 if $\boldsymbol{x}_{ui}$ or $S^{ui}_{u'i'}$ is absent.

Assumption A defines the smoothness of θ^0_{ui} in terms of the continuous covariate $\boldsymbol{x}_{ui}$ in the presence of connected user-item pairs in the network. As a special case, if $\boldsymbol{x}_{ui}$ is absent, Assumption A degenerates to $\theta^0_{ui} = \theta^0_{u'i'}$ for pairs with $S^{ui}_{u'i'} = 1$. This assumption is mild when all covariates are available, and is relatively more restrictive when $\boldsymbol{x}_{ui}$ is absent as it pushes the model to be a parametric one with only a finite number of unknown θ's. It is necessary to capture the smooth property of θ over the network structure, and similar assumptions are widely used in nonparametric regression (Vieu, 1991; Wassermann, 2006) and kernel density estimation (Stone, 1984; Marron and Padgett, 1987).

Assumption B. The continuous covariate $\boldsymbol{x}$ has a bounded support $\mathcal{X}$, and the error term ε_{ui} has a sub-Gaussian distribution with variance σ^2.

Assumption B is a regularity condition for the underlying probability distribution, and widely used in literature (Ma and Huang, 2017; Lin et al., 2017). Further, assume that $\{\boldsymbol{x}_{ui}, \Delta_{ui}\}_{1\le u\le n, 1\le i\le m}$ are independent and identically distributed, but the distribution of Δ_{ui} may depend on $\boldsymbol{x}_{ui}$. We denote the parameter space $\boldsymbol{\Gamma}(L)$ to be $\{\boldsymbol{\theta} : \|\boldsymbol{\theta}\|_\infty \le L\}$, where $\|\cdot\|_\infty$ is the uniform norm, and L is chosen so that $\boldsymbol{\theta}^0 \in \boldsymbol{\Gamma}(L)$. To accurately utilize the network information, we set $\omega_{ui,u'i'} = 0$ if $S^{ui}_{u'i'} = 0$. Theorem 1 establishes an upper bound for the estimation error of the proposed recommender system, where the convergence rate is determined by the size of the preference matrix nm, the number of parameters K, the size of the observed ratings $|\Omega|$, the tuning parameters λ_1 and λ_2, and the window size h.

Theorem 1 *Suppose that Assumptions A and B are satisfied. For some constant $c_2 > 0$, let $\kappa_1 = \max_{u,i}\sum_{(u',i')\in\Omega}\omega_{ui,u'i'}\|\boldsymbol{x}_{ui} - \boldsymbol{x}_{u'i'}\|_2^\alpha$, and $\kappa_2 = \max_{u,i}\sum_{(u',i')\in\Omega}\omega^2_{ui,u'i'}$, then*

$$P\big(RMSE(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^0) \ge \eta\big) \le \exp\big\{-\frac{c_2\eta^2}{\kappa_2} + \log(nm)\big\},$$

provided that $\eta \ge \max\big\{\sqrt{K}\kappa_1, \sqrt{\kappa_2}\big\}\log(nm)$ and $\lambda_1\sum_{u=1}^{n}J(\boldsymbol{p}^0_u) + \lambda_2\sum_{i=1}^{m}J(\boldsymbol{q}^0_i) \le \eta^2/4$. The convergence rate then becomes $RMSE(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^0) = O_p(\max\big\{\sqrt{K}\kappa_1, \kappa_2^{1/2}\big\}\log(nm))$, where κ_1 and κ_2 tend to zero and can be computed for some specific weights.

Theorem 1 provides a general upper bound of RMSE$(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^0)$, which may vary by the choice of weights $\omega_{ui,u'i'}$ and the two quantities κ_1 and κ_2. In the following, an explicit convergence rate is given for $\omega_{ui,u'i'}$ defined in (5) under some additional assumptions.

Let $\Delta_{ui} \in \{0, 1\}$ be a binary variable, with $\Delta_{ui} = 1$ indicating that r_{ui} is observed and 0 otherwise. Assume that $\{\boldsymbol{x}_{ui}, \Delta_{ui}\}_{1 \leq u \leq n, 1 \leq i \leq m}$ are independent and identically distributed, but the distribution of Δ_{ui} may depend on $\boldsymbol{x}_{ui}$.

Assumption C. For any (u, i) and (u', i'), $P(S^{ui}_{u'i'} = 1 | \Delta_{u'i'} = 1)$ is bounded away from zero, and the conditional density $f_{U^{ui}_{u'i'} | S^{ui}_{u'i'}=1, \Delta_{u'i'}=1}$ is continuous and bounded away from zero, where $U^{ui}_{u'i'} = \|\boldsymbol{x}_{ui} - \boldsymbol{x}_{u'i'}\|_2$.

Assumption C is necessary for tackling the "cold-start" problem, and similar assumptions are also widely used in the local smoothing technique (Chen et al., 2014; Scott, 2015). It ensures that for any pair (u, i), the probability of $\Delta_{ui} = 1$ may depend on covariates $\boldsymbol{x}_{ui}$ and $S^{ui}_{u'i'}$, and that the corresponding neighboring pairs are observed with positive probability. In fact, it suffices to assume that $P(S^{ui}_{u'i'} = 1 | \Delta_{u'i'} = 1)$ is bounded away from zero with certain order, and similar results can be obtained with more involved derivation.

Assumption D. There exists a constant c_2 such that the nonnegative kernel $\mathcal{K}(\cdot)$ satisfies

$$\max\Big\{ \int_0^\infty \mathcal{K}^2(u) du, \int_0^\infty \mathcal{K}(u) u^\alpha du \Big\} \leq c_3.$$

Assumption D is a standard assumption for smoothing kernels. Notably, the choice of kernel should match up the smoothness at an order α of $\boldsymbol{\theta}^0$. Alternatively, to effectively employ the smoothing pattern of θ, the decay rate of the chosen kernel may be larger than α, to filter out more distant user-item pairs and preserve more reliable local neighborhood structure. For example, the Gaussian kernel has an exponential decay rate, and always satisfies the inequality conditions in Assumption D.

Corollary 1 *Suppose that Assumptions A-D are satisfied. For $\omega_{ui,u'i'}$ defined in* (5)*, then the convergence rate of RMSE$(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^0)$ becomes RMSE$(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^0) = O_p\big(|\Omega|^{-\frac{\alpha}{2\alpha+1}} K^{\frac{1}{2(2\alpha+1)}} \log(nm)\big)$.*

Note that this convergence rate is intriguing compared with some existing results. Particularly, when $K = O(1)$, we have RMSE$(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^0) = O_P\big(|\Omega|^{-\frac{\alpha}{2\alpha+1}} \log(nm)\big)$. For $\alpha > 1/2$, since $|\Omega| \leq nm < (n+m)^2$, it leads to a tighter bound than $O_P\Big(\big(\frac{n+m}{|\Omega|} \log(\frac{\sqrt{nm}}{|\Omega|})\big)^{\frac{1}{2}}\Big)$ and $O_P\Big(\big(\frac{n+m}{|\Omega|} \log(m) \log(\frac{|\Omega|}{n+m})\big)^{\frac{1}{2}}\Big)$ established in Bi et al. (2017) and Srebro et al. (2005), respectively. In addition, Theorem 1 still guarantees the convergence of RMSE$(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^0)$, if K goes to infinity at a rate slower than $|\Omega|^{2\alpha}(\log(nm))^{-2(2\alpha+1)}$. In practice, the size of Ω is often much less than nm, and only proportional to $(n+m)K$. For example, in the MovieLens 1M dataset with $K = 10$, $nm = 0.2 \times 10^8, |\Omega| = 10^6$, and $(n+m)K = 10^6$; and in the *Last.fm* dataset, $nm = 0.3 \times 10^8, |\Omega| = 10^6$ and $(n+m)K = 0.2 \times 10^6$. In such cases with $K = O(1)$, the theoretical results in Bi et al. (2017) and Srebro et al. (2005) fail to give a reasonable convergence rate. However, Theorem 1 still yields that RMSE$(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^0) = O_P\big(|\Omega|^{-\frac{\alpha}{2\alpha+1}} \log(nm)\big)$. Interestingly, if the continuous covariates are absent, then Assumption A is satisfied with $\alpha = \infty$, and there is only a finite number of θ's to be estimated for different discrete covariates. In such cases, Theorem 1 implies that the recommender system can be estimated with a rate $\eta \sim O_P\big(|\Omega|^{-1/2} \log(nm)\big)$.

5. Numerical results

This section examines the performance of the proposed method, denoted as sSVD, in simulations and with a real *Last.fm* dataset. We compare sSVD to several strong competitors, including restricted Boltzmann machines (RBM; Salakhutdinov et al. 2007) or a continuous version of restricted Boltzmann machines (CRBM; Chen and Murray 2003); an iterative soft-threshold matrix completion method (SoftImpute; Hastie et al. 2015); the regularized SVD (rSVD; Paterek 2007); the self-recovered side regression (SSR; Zhao et al. (2016)) and a group-specific SVD approach (gSVD; Bi et al. 2017). Note that the Python codes for RBM and CRBM are publicly available (`https://github.com/yusugomori/DeepLearning/tree/master/python`), the code for SoftImpute is available in the Python package "fancyimpute", the Matlab code for SSR is provided by Zhao et al. (2016), and the R code for gSVD is provided by Bi et al. (2017). Note that RBM and CRBM are essentially the same but differ in implementation only for discrete and continuous responses. Although RBM, SoftImpute and rSVD are not designed to incorporate covariates, we include them in comparison to illustrate the importance of utilizing covariates for personalized prediction.

For tuning parameter selection, we set the learning rate, the momentum rate and the number of hidden units for RBM and CRBM as 0.005, 0.9, and 100, respectively. For rSVD, gSVD and sSVD, we set the tuning parameter K to be the true one, and the optimal λ is determined by a grid search over $\{10^{(\nu-31)/10}; \nu = 1, \cdots, 61\}$. For the proposed sSVD, a Gaussian kernel is used with the window size h being the median distance among all user-item pairs. The predictive performance of all methods is measured by the root mean squared error (RMSE).

5.1. Simulated examples

This subsection investigates the "cold-start" problem and the utility of a user-item specific network on prediction performance. We generate the simulation setting as follows. The dimensions of a rating matrix $\{r_{ui}\}_{1\leq u\leq n; 1\leq i\leq m}$ are $n = 501$ and $m = 201$. Let $\boldsymbol{p}_u = (1 + 0.002u)\mathbf{1}_K + N(\mathbf{0}_K, \xi\boldsymbol{I}_K)$ and $\boldsymbol{q}_i = (1 + 0.0075i)\mathbf{1}_K + N(\mathbf{0}_K, \xi\boldsymbol{I}_K)$ for $u = 1, \cdots, n$ and $i = 1, \cdots, m$, where $\mathbf{1}_K$ and $\mathbf{0}_K$ being K-dimensional vectors of ones and zeros, respectively. Here we sample observed ratings at random, where π is the missing rate, and the total number of observed ratings is $|\Omega| = (1 - \pi)nm$. For each user-item pair $(u, i) \in \Omega$, we let u and i be uniformly sampled from $1, \cdots, n$ and $1, \cdots, m$, respectively, and r_{ui} be generated from a truncated normal distribution on $[1, 5]$ with mean $\boldsymbol{p}_u^T\boldsymbol{q}_i$ and standard deviation 0.5. Further, r_{ui} is rounded to the closest integer in $\{1, \cdots, 5\}$ to mimic discrete ratings in practice.

To generate a similarity network for each user-item (u, i) pair, we connect (u, i) with its nearest first-order neighbor indices, say $u - 1$ and $u + 1$ for u and $i - 1$ and $i + 1$ for i. This mimics a user's friendship network. In addition, we introduce a "cold-start" rate ρ to examine the prediction accuracy of the proposed method on new users and items. Specifically, we randomly select ρ-proportion of users and items, and retain all of their corresponding ratings in a testing set to resemble the "cold-start" phenomenon. The remaining ratings are randomly split into training, tuning and testing sets with 60%, 15%, and 25% of the observations, respectively. In simulations, we choose $\xi = 0.1$, $K = 3$ or 6, the missing rate $\pi = .8, .9, .95$, and the "cold-start" rate $\rho = 0, 0.1, 0.2$. For the proposed sSVD, the weight function is set as $\omega_{ui,u'i'} = S^{ui}_{u'i'}$, which is constructed based on the adjacency of each user-item pairs. For gSVD, we use the ranking percentages of each user and item as covariates according to Bi et al. (2017), but the user-item specific network information is not incorporated.

Table 1: *Averaged RMSEs of various methods and their estimated standard deviations in parentheses on the simulated examples over 50 simulations. Here RBM, SoftImput, rSVD, gSVD, sSVD denote: restricted Boltzmann machines (Salakhutdinov et al., 2007), SoftImput method (Hastie et al., 2015), regularized SVD method (Paterek, 2007), group-specific SVD method (Bi et al., 2017) and the proposed method, respectively. The best performer in each setting is bold-faced.*

	RBM	SoftImpute	SSR	rSVD	gSVD	sSVD
Use of covariate	No	No	Yes	No	Yes	Yes
K = 3, π = 0.8						
$\rho = 0.0$	0.842(0.001)	1.082(.003)	2.471(.002)	0.323(.000)	**0.302(.000)**	0.364(.001)
$\rho = 0.1$	0.911(0.003)	2.100(.007)	2.570(.004)	1.112(.009)	0.472(.003)	**0.371(.001)**
$\rho = 0.2$	0.943(0.002)	2.404(.007)	2.650(.014)	1.331(.007)	0.578(.009)	**0.375(.001)**
K = 3, π = 0.9						
$\rho = 0.0$	0.848(0.002)	2.146(0.003)	2.617(.002)	0.498(0.002)	**0.351(0.002)**	0.368(0.001)
$\rho = 0.1$	0.915(0.002)	2.486(0.006)	2.674(.003)	1.170(0.010)	0.525(0.007)	**0.378(0.001)**
$\rho = 0.2$	0.948(0.003)	2.261(0.005)	2.697(.005)	1.418(0.007)	0.622(0.006)	**0.388(0.001)**
K = 3, π = 0.95						
$\rho = 0.0$	0.861(0.002)	2.554(0.005)	2.693(.004)	1.002(0.004)	0.442(0.002)	**0.379(0.001)**
$\rho = 0.1$	0.923(0.003)	2.668(0.006)	2.716(.005)	1.445(0.009)	0.707(0.017)	**0.407(0.002)**
$\rho = 0.2$	0.956(0.003)	2.722(0.005)	2.733(.004)	1.636(0.007)	0.849(0.014)	**0.440(0.002)**
K = 6, π = 0.8						
$\rho = 0.0$	0.843(0.001)	1.086(0.003)	2.472(.002)	0.340(0.001)	**0.302(0.001)**	0.367(0.001)
$\rho = 0.1$	0.914(0.003)	2.113(0.006)	2.584(.004)	0.823(0.008)	0.472(0.005)	**0.374(0.001)**
$\rho = 0.2$	0.943(0.003)	2.414(0.006)	2.634(.004)	0.984(0.007)	0.568(0.005)	**0.382(0.001)**
K = 6, π = 0.9						
$\rho = 0.0$	0.851(0.002)	2.156(0.004)	2.616(.003)	0.451(0.001)	**0.350(0.001)**	0.373(0.001)
$\rho = 0.1$	0.908(0.002)	2.492(0.005)	2.667(.005)	0.872(0.007)	0.529(0.005)	**0.387(0.001)**
$\rho = 0.2$	0.946(0.002)	2.625(0.004)	2.697(.003)	1.030(0.006)	0.619(0.006)	**0.403(0.001)**
K = 6, π = 0.95						
$\rho = 0.0$	0.857(0.003)	2.558(0.005)	2.697(.003)	0.713(0.003)	0.442(0.002)	**0.396(0.002)**
$\rho = 0.1$	0.925(0.002)	2.679(0.006)	2.726(.005)	1.015(0.009)	0.696(0.019)	**0.422(0.002)**
$\rho = 0.2$	0.955(0.003)	2.717(0.005)	2.733(.004)	1.154(0.006)	0.866(0.012)	**0.460(0.002)**

Table 1 indicates that the proposed sSVD achieves the highest predictive accuracy compared against its competitors in 14 out of 18 different settings, whereas its accuracy is close to the best performer gSVD in the remaining four settings. Note that these four settings correspond to no "cold-start" user-item pairs in a linear situation. This simulation result is anticipated as only sSVD is capable of utilizing user-specific social networks. Interestingly, gSVD, without using the user-specific networks, performs well as it is able to capture user-user dependencies with respect to ranking due to strong dependency from the neighboring user-specific friendship networks. By comparison, SSR has a comparable performance as SoftImput, which is much worse than rSVD, gSVD, and sSVD, partly because it fails to account for non-ignorable missing.

Compared with the methods without using covariates such as rSVD and SoftImpute, sSVD yields superior performance in most scenarios, and the percentages of improvement can be as high as 83.1%. In contrast to RBM, SoftImpute and SSR, the percentages of improvement from the sSVD range from 51.8% to 83.2%. In addition, sSVD is robust against high missing rates and high "cold-start" rates as measured by the missing rate π and the cold-start rate ρ. As π and ρ increase, the performance of sSVD is stable, whereas the performances of competitors deteriorate rapidly. In other words, the improvement of sSVD over its competitors is more significant when π and ρ are large. For example, the largest amount of improvement of sSVD over the second best performer gSVD is 46.9% when $K = 6$, $\pi = .95$, and $\rho = .2$.

Moreover, we examine the robust aspect of the proposed sSVD with respect to the degree of smoothness of ratings in a neighborhood and the degree of sparseness of a user-specific network. Specifically, given $K = 6$ and $\pi = 0.95$, we contaminate the neighborhood structure by increasing the noise level $\xi = 0.1, 0.2, 0.4, 0.6$, and allow the sparsity ratio $\pi_{network} = 0.0, 0.2, 0.5, 1.0$ for user-item specific networks, where the sparseness ratio is the probability that the edges connecting each pair of users or items are removed from the original network. Thus as $\pi_{network}$ increases, the resultant network becomes sparser, and the proposed sSVD degenerates to rSVD when $\pi_{network} = 1.0$. As suggested by Table 2, the performance of sSVD deteriorates as the contamination level $\pi_{network}$ escalates, yet still outperforming rSVD when $\xi = 0.1, 0.2$. On the other hand, rSVD outperforms the proposed sSVD when $\xi \geq 0.4$, which is anticipated because the proposed sSVD relies on the smoothness structure of the ratings. Moreover, it is evident that the proposed sSVD is less affected by the "cold-start" users and items when $\pi_{network}$ decreases.

Table 2: *Averaged RMSEs and estimated standard deviations over 50 simulations for the proposed method when the error standard deviation is 0.1, 0.2, 0.4, or 0.6, and the "cold-start" rate ρ is 0.0, 0.1, or 0.2, and the network structure missing rate $\pi_{network}$ is 0.0, 0.2, 0.5, or 1.0.*

	$\xi = 0.1$	$\xi = 0.2$	$\xi = 0.4$	$\xi = 0.6$
$\pi_{network} = 0.0$				
$\rho = 0.0$	0.396(0.002)	0.589(0.003)	0.984(0.005)	1.547(0.010)
$\rho = 0.1$	0.422(0.002)	0.665(0.004)	1.032(0.014)	1.596(0.015)
$\rho = 0.2$	0.460(0.002)	0.716(0.005)	1.094(0.009)	1.683(0.016)
$\pi_{network} = 0.2$				
$\rho = 0.0$	0.512(0.002)	0.608(0.003)	0.987(0.007)	1.518(0.010)
$\rho = 0.1$	0.599(0.004)	0.694(0.005)	1.062(0.010)	1.622(0.016)
$\rho = 0.2$	0.648(0.006)	0.746(0.004)	1.129(0.009)	1.660(0.013)
$\pi_{network} = 0.5$				
$\rho = 0.0$	0.569(0.003)	0.648(0.004)	1.000(0.009)	1.516(0.014)
$\rho = 0.1$	0.660(0.004)	0.743(0.007)	1.092(0.008)	1.629(0.011)
$\rho = 0.2$	0.711(0.007)	0.796(0.006)	1.127(0.010)	1.670(0.012)
$\pi_{network} = 1.0$				
$\rho = 0.0$	0.713(0.003)	0.745(0.006)	0.952(0.009)	1.397(0.008)
$\rho = 0.1$	1.015(0.009)	1.046(0.011)	1.289(0.014)	1.686(0.015)
$\rho = 0.2$	1.154(0.006)	1.193(0.009)	1.415(0.013)	1.790(0.020)

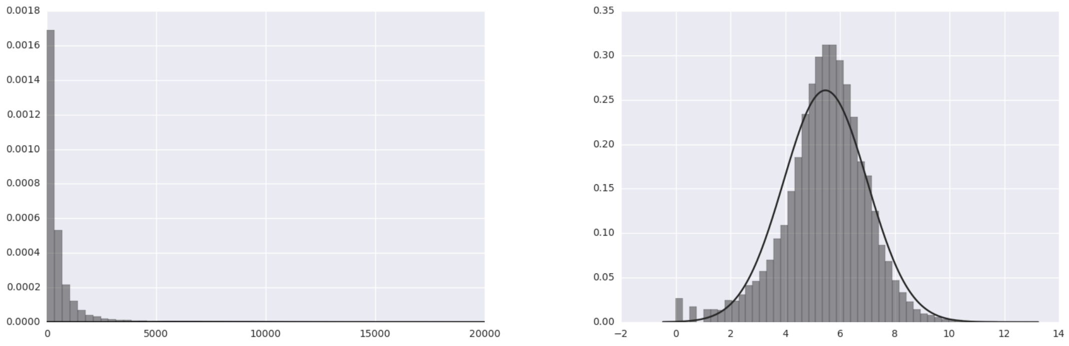
In summary, the simulation studies illustrate that integration of user-item specific networks into latent factor modeling can enhance the accuracy of prediction and tackle the "cold-start" issue. In this regard, sSVD is a more effective recommender system than the existing methods in the literature.

5.2. Music data from *Last.fm*

In this section, we analyze a online music dataset from the *Last.fm* (`http://www.last.fm`), which was released in the second International Workshop HetRec 2011 (`http://ir.ii.uam.es/hetrec2011`). This dataset contains user-artist listening counts, social networks among 1,892 users, and tagging information for 17,632 artists. The user-artist listening counts contain 92,834 tuples defined by `[user, artist, listeningCount]`, where each user listens to an average of about 49 artists, and each artist has been listened to by about 5 users on average. Note that the observation rate in this dataset is below 0.28%, leading to a severe "cold-start" problem.

For this application, we focus on utilizing user-item specific covariates and networks to solve the "cold-start" issue. Specifically, in the *Last.fm* dataset, a user-specific social network is available based on 12,717 bi-directional user friendships with an average of about 13 friends per user. Moreover, there are 186,479 tags given by users to artists, with an average of about 99 tags per user, and about 15 tags per artist. A typical tag of an artist gives a short description of the artist, such as "rock", "electronic", "jazz" and "80s". The artist-specific tags are converted to a l-length 0/1 covariate vector to indicate whether music tags are assigned to an artist by users, where l is the total number of different tags.

Figure 2 illustrates the listening pattern, showing that a log-transformation of the listening counts appears to be normally distributed. In addition, Figure 3 indicates that only a small portion of artists are popular and frequently listened to by many users. However, the majority of the artists are categorized in more specialized genres, which make them less popular over all, but still having their own followers. This is illustrated by the horizontal and diagonal stripes in Figure 3 respectively.



(a) Original listening counts

(b) log-transformed listening counts

Figure 2: The original and log-transformed listening counts in the *Last.fm* dataset.

Figure 4 shows that popular artists tend to be listened to more by each follower; that is, the number of users listening to each artist is positively related with its averaged listening counts, indicating that missing is non-ignorable and that the missing pattern shall be incorporated in the recommender system.

For comparison, we investigate the performance of five methods: CRBM, SoftImpute, SSR, rSVD, and gSVD, where the residuals of these methods are obtained after adjusting for

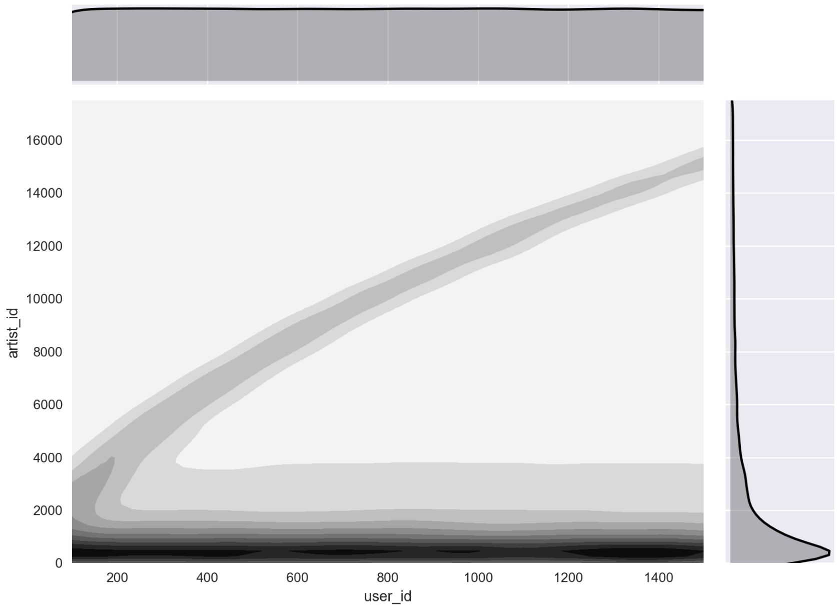


Figure 3: Two-dimensional histogram plot of the joint distribution of the observed user-artist pairs from the *Last.fm* dataset. The grey level represents the concentration, with darker color indicating more densely populated. Together with the joint distribution, the marginal distributions for user and artist are displayed along the horizontal and vertical axes.

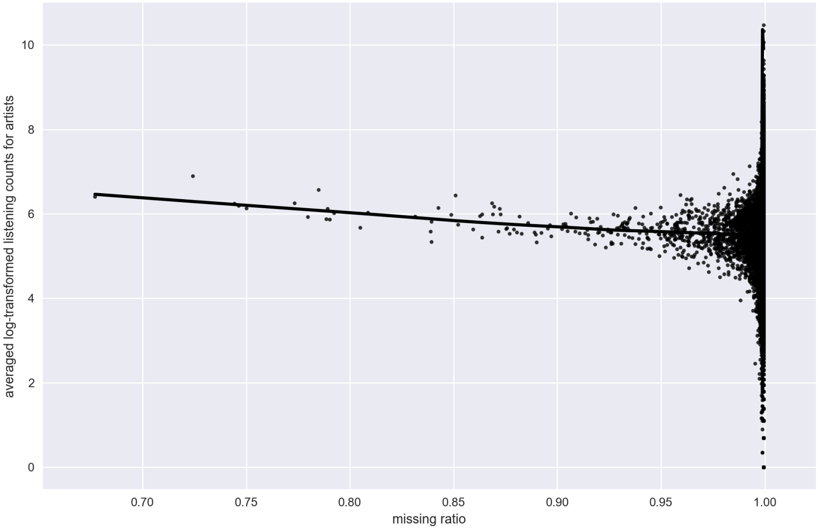


Figure 4: Missing pattern analysis for artists in the *Last.fm* dataset. It is clear that artists attracting fewer users tend to have smaller averaged listening counts.

covariate effects from user-specific social networks and artist-specific tags. Furthermore, to incorporate the non-ignorable missing pattern, the missing percentage is used to generate 12 homogeneous subgroups of users and 10 homogeneous subgroups of artists for gSVD, and covariates $(\boldsymbol{x}_u, \boldsymbol{x}_i)$ for sSVD are obtained from 5%, 25%, 50%, 75%, and 95% quantiles of the log observed listening counts in the training set for each user and artist. Also, we slightly modify the weight $\omega_{ui,u'i'}$ defined in (5) using thresholding. Specifically, $S^{ui}_{u'i'} = S^u_{u'} S^i_{i'}$, where $S^u_{u'} = 0.8$ if user u and u' are adjacent in the user social network, and $S^u_{u'} = 0.5$ otherwise; and $S^i_{i'}$ is computed based on the cosine similarity between the tag information of artists i and i'. For new users or artists, only user social network and artist tags information are available, therefore the weight function for covariate is automatically set as one. Furthermore, we apply a Gaussian kernel with window size h as the median distance among all user-artist pairs. For each user-artist pair (u, i), we compute the weights as in (5), and truncate them to keep only the five most similar user-artist pairs for each (u, i) to facilitate computation.

For evaluation, we apply 5-fold cross-validation over a random partition of the original dataset, and calculate the RMSE as in Koyejo and Ghosh (2011). For SVD-based methods, we set $K = 5$, and select the optimal λ from $\{1, \cdots, 25\}$ through the 5-fold cross-validation. For CRBM, the parameters are set as suggested in Nguyen and Lauw (2016); and for SSR, the parameters are determined through cross-validation as suggested in Zhao et al. (2016).

Table 3: *RMSEs of various methods and their estimated standard deviations in parentheses for observed, cold-start, and entire pairs on the Last.fm dataset. Here RBM, SoftImput, SSR, rSVD, gSVD, sSVD denote: restricted Boltzmann machines (Salakhutdinov et al., 2007), SoftImput method (Hastie et al., 2015), Self-recovered side regression (Zhao et al., 2016), regularized SVD method (Paterek, 2007), group-specific SVD method (Bi et al., 2017) and the proposed method, respectively. The second column indicates what type of covariates are used by each method, where N, T and M denote user-specific social networks, artist tags and non-ignorable missing pattern, respectively. The best performance in each setting is bold-faced.*

	Covariate	Observed pair	“Cold-start” pair	Entire pair
Regression	N, T	1.507(.005)	1.832(.023)	1.552(.005)
CRBM	N, T	1.507(.005)	1.834(.023)	1.552(.004)
SoftImpute	N, T	1.308(.005)	1.832(.023)	1.386(.004)
SSR	N, T	1.436(.006)	1.841(.035)	1.493(.009)
rSVD	N, T	1.124(.006)	1.832(.023)	1.237(.004)
gSVD	M, T	0.997(.013)	1.202(.012)	1.026(.011)
sSVD	N, M, T	**0.880(.004)**	**0.706(.003)**	**0.860(.003)**

Table 3 shows that sSVD significantly outperforms its competitors with a RMSE of 0.860 for the entire dataset, whereas gSVD is the second best performer with a RMSE of 1.020. As a reference, these two RMSEs are both smaller than 1.071 for the weighted interaction method under the same setting, which is reported as the best performer in analyzing the *Last.fm* dataset (Koyejo and Ghosh, 2011). The amounts of improvement of sSVD over gSVD, rSVD, SoftImput, SSR and CRBM are 15.6%, 24.6%, 38.0%, 42.4%, and 44.6%, respectively. For the "cold-start" pairs, sSVD yields a more than 40% improvement over the second-best competitor gSVD. Note that SoftImput and rSVD do not perform well in handling the "cold-start" problem, as their penalization leads to the same performance as the regression approach.

6. Summary

This article proposes a smooth collaborative recommender system which integrates the network structure of user-item pairs to improve prediction accuracy. The proposed method provides a flexible framework to exploit the covariate information, such as user demographics, item contents, and social network information for users and/or items. The network structure allows us to increase the effective sample size for higher prediction accuracy, in addition to producing a more accurate recommender system for "cold-start" pairs. In addition, we implement a "divide-and-conquer" type of alternating least square algorithm. We also establish the asymptotic properties of the proposed method, which provide the theoretical foundation of its superior performance over other state-of-the-art methods. Although the proposed method is formulated based on the latent factor model, the framework can be extended to other models, such as Koren (2008) and Bi et al. (2017).

Acknowledgments

We would like to acknowledge support for this project from the National Science Foundation grants DMS-1415500, DMS-1712564, DMS-1721216, DMS-1613190, DMS-1415308, DMS-1821198, the National Institute of Health grants 1R01GM126002, R01HL105397, and Hong Kong Research Grant Council grants GRF-11302615, GRF-11303918 and GRF-11331016. The authors are grateful to reviewers and the Action Editor for their insightful comments and suggestions which have improved the manuscript significantly.

Appendix: technical proofs

Proof of Lemma 1. By the definition of $(\hat{\boldsymbol{P}}, \hat{\boldsymbol{Q}})$ as a minimizer of (3), we have

$$
\begin{aligned}
(\hat{\boldsymbol{P}}, \hat{\boldsymbol{Q}}) &= \operatorname*{argmin}_{\boldsymbol{P},\boldsymbol{Q}} \ \frac{1}{nm}\sum_{u=1}^{n}\sum_{i=1}^{m}\Big(\sum_{(u',i')\in\Omega} \omega_{ui,u'i'}(r_{u'i'} - \boldsymbol{p}_u^T\boldsymbol{q}_i)^2\Big) + \lambda_1\sum_{u=1}^{n} J(\boldsymbol{p}_u) + \lambda_2\sum_{i=1}^{m} J(\boldsymbol{q}_i) \\
&= \operatorname*{argmin}_{\boldsymbol{P},\boldsymbol{Q}} \ \frac{1}{nm}\sum_{u=1}^{n}\sum_{i=1}^{m}\Big(\sum_{(u',i')\in\Omega} \omega_{ui,u'i'} r_{u'i'}^2 - \sum_{(u',i')\in\Omega} 2\omega_{ui,u'i'} r_{u'i'}\boldsymbol{p}_u^T\boldsymbol{q}_i \\
&\qquad + \sum_{(u',i')\in\Omega} \omega_{ui,u'i'}(\boldsymbol{p}_u^T\boldsymbol{q}_i)^2\Big) + \lambda_1\sum_{u=1}^{n} J(\boldsymbol{p}_u) + \lambda_2\sum_{i=1}^{m} J(\boldsymbol{q}_i) \\
&= \operatorname*{argmin}_{\boldsymbol{P},\boldsymbol{Q}} \ \frac{1}{nm}\sum_{u=1}^{n}\sum_{i=1}^{m}\Big(\sum_{(u',i')\in\Omega} \omega_{ui,u'i'} r_{u'i'}^2 - \sum_{(u',i')\in\Omega} 2\omega_{ui,u'i'} r_{u'i'}\boldsymbol{p}_u^T\boldsymbol{q}_i + (\boldsymbol{p}_u^T\boldsymbol{q}_i)^2 \\
&\qquad - \sum_{(u',i')\in\Omega} \omega_{ui,u'i'} r_{u'i'}^2 + \big(\sum_{(u',i')\in\Omega} \omega_{ui,u'i'} r_{u'i'}\big)^2\Big) + \lambda_1\sum_{u=1}^{n} J(\boldsymbol{p}_u) + \lambda_2\sum_{i=1}^{m} J(\boldsymbol{q}_i) \\
&= \operatorname*{argmin}_{\boldsymbol{P},\boldsymbol{Q}} \ \frac{1}{nm}\sum_{u=1}^{n}\sum_{i=1}^{m}\Big(\big(\sum_{(u',i')\in\Omega} \omega_{ui,u'i'} r_{u'i'}\big)^2 - \sum_{(u',i')\in\Omega} 2\omega_{ui,u'i'} r_{u'i'}\boldsymbol{p}_u^T\boldsymbol{q}_i + (\boldsymbol{p}_u^T\boldsymbol{q}_i)^2\Big) \\
&\qquad + \lambda_1\sum_{u=1}^{n} J(\boldsymbol{p}_u) + \lambda_2\sum_{i=1}^{m} J(\boldsymbol{q}_i) \\
&= \operatorname*{argmin}_{\boldsymbol{P},\boldsymbol{Q}} \ \frac{1}{nm}\sum_{u=1}^{n}\sum_{i=1}^{m}\Big(\sum_{(u',i')\in\Omega} \omega_{ui,u'i'} r_{u'i'} - \boldsymbol{p}_u^T\boldsymbol{q}_i\Big)^2 + \lambda_1\sum_{u=1}^{n} J(\boldsymbol{p}_u) + \lambda_2\sum_{i=1}^{m} J(\boldsymbol{q}_i),
\end{aligned}
$$

where the third equality follows that $\sum_{(u',i')\in\Omega}\omega_{ui,u'i'} = 1$ and the fact that adding constants to the cost function does not impact minimization. This completes the proof.

Proof of Theorem 1. Our treatment for bounding $P\big(\mathrm{RMSE}(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) \geq \eta\big)$ is to bound empirical processes induced by $\mathrm{RMSE}(\cdot,\cdot)$ by a chaining argument as in (Wong and Shen, 1995; Shen et al., 2003; Liu and Shen, 2006).

Let $\boldsymbol{\Gamma}(L,\eta) = \{\boldsymbol{\theta} \in \boldsymbol{\Gamma}(L) : \mathrm{RMSE}(\boldsymbol{\theta}, \boldsymbol{\theta}^0) \geq \eta\}$ be a parameter subset of the parameter space $\boldsymbol{\Gamma}(L)$. Let $\lambda J(\boldsymbol{\theta}) = \lambda_1\sum_{u=1}^{n} J(\boldsymbol{p}_u) + \lambda_2\sum_{i=1}^{m} J(\boldsymbol{q}_i)$ be the regularizer.

Note that $\hat{\boldsymbol{\theta}}$ is a minimizer of $\mathcal{L}(\boldsymbol{P},\boldsymbol{Q})$ in $\boldsymbol{\Gamma}(L)$, we have that $P\big(\mathrm{RMSE}(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^0)) \geq \eta\big) \leq P^*\Big(\sup_{\boldsymbol{\theta}\in\boldsymbol{\Gamma}(L,\eta)}(\mathcal{L}(\boldsymbol{P}^0,\boldsymbol{Q}^0) - \mathcal{L}(\boldsymbol{P},\boldsymbol{Q})) \geq 0\Big)$, where P^* is the outer probability (Billingsley, 2013). Using the expression of $\mathcal{L}(\boldsymbol{P},\boldsymbol{Q})$, we obtain an upper bound of the latter as follows.

$$
\begin{aligned}
&P^*\Big(\sup_{\boldsymbol{\theta}\in\boldsymbol{\Gamma}(L,\eta)} (nm)^{-1}\sum_{u,i}\sum_{(u',i')\in\Omega} \omega_{ui,u'i'}(\theta_{ui} - \theta_{ui}^0)(2r_{u'i'} - \theta_{ui}^0 - \theta_{ui}) + \lambda J(\boldsymbol{\theta}^0) - \lambda J(\boldsymbol{\theta}) \geq 0\Big) \\
&\leq P^*\Big(\sup_{\boldsymbol{\theta}\in\boldsymbol{\Gamma}(L,\eta)} (nm)^{-1}\sum_{u,i}\sum_{(u',i')\in\Omega} \omega_{ui,u'i'}(\theta_{ui} - \theta_{ui}^0)(2r_{u'i'} - \theta_{ui}^0 - \theta_{ui}) \geq -\lambda J(\boldsymbol{\theta}^0)\Big),
\end{aligned}
$$

where the fact that $\lambda J(\boldsymbol{\theta}) \geq 0$ has been used. Now, let $A_j = \big\{\boldsymbol{\theta} \in \boldsymbol{\Gamma}(L) : 2^{j-1}\eta \leq \mathrm{RMSE}(\boldsymbol{\theta}, \boldsymbol{\theta}^0) \leq 2^j\eta\big\}$; $j = 1, \cdots, \infty$ be a partition in that $\boldsymbol{\Gamma}(L,\eta) = \bigcup_{j=1}^{\infty} A_j$. Combining the above inequalities, we have that $P\Big(\text{RMSE}(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^0) \geq \eta\Big)$ is bounded by

$$\sum_{j=1}^{\infty} P^*\Big(\sup_{A_j} (nm)^{-1} \sum_{u,i} \sum_{(u',i')\in\Omega} \omega_{ui,u'i'}(\theta_{ui} - \theta^0_{ui})(2r_{u'i'} - \theta^0_{ui} - \theta_{ui}) \geq -\lambda J(\boldsymbol{\theta}^0)\Big) \equiv \sum_{j=1}^{\infty} I_j.$$

Next, we bound each I_j separately. Since $\sum_{(u',i')\in\Omega} \omega_{ui,u'i'} = 1$,

$$\begin{aligned}
& (nm)^{-1} \sum_{u,i} \sum_{(u',i')\in\Omega} \omega_{ui,u'i'}(\theta_{ui} - \theta^0_{ui})(2r_{u'i'} - \theta^0_{ui} - \theta_{ui}) \\
= \; & (nm)^{-1} \sum_{u,i} \sum_{(u',i')\in\Omega} 2\omega_{ui,u'i'}(\theta_{ui} - \theta^0_{ui})(\theta^0_{u'i'} - \theta^0_{ui}) \\
& + (nm)^{-1} \sum_{u,i} \sum_{(u',i')\in\Omega} 2\omega_{ui,u'i'}(\theta_{ui} - \theta^0_{ui})\epsilon_{u'i'} - (nm)^{-1} \sum_{u,i} (\theta_{ui} - \theta^0_{ui})^2,
\end{aligned}$$

which, by Assumption A and the fact that $\omega_{ui,u'i'} = 0$ when $S^{ui}_{u'i'} = 0$, is upper-bounded by

$$\begin{aligned}
& c_1\sqrt{K}(nm)^{-1} \sum_{u,i} |\theta_{ui} - \theta^0_{ui}| \sum_{(u',i')\in\Omega} 2\omega_{ui,u'i'} \|\boldsymbol{x}_{u'i'} - \boldsymbol{x}_{ui}\|_2^{\alpha} \\
& \qquad + (nm)^{-1} \sum_{u,i} \sum_{(u',i')\in\Omega} 2\omega_{ui,u'i'}(\theta_{ui} - \theta^0_{ui})\epsilon_{u'i'} - (nm)^{-1} \sum_{u,i} (\theta_{ui} - \theta^0_{ui})^2.
\end{aligned}$$

Then for any $\boldsymbol{\theta} \in A_j$, $(2^j\eta)^2 \geq (nm)^{-1} \sum_{u,i} (\theta_{ui} - \theta^0_{ui})^2 \geq (2^{j-1}\eta)^2$, by the Cauchy-Schwarz inequality,

$$\begin{aligned}
& \sup_{A_j} c_1\sqrt{K}(nm)^{-1} \sum_{u,i} |\theta_{ui} - \theta^0_{ui}| \sum_{(u',i')\in\Omega} 2\omega_{ui,u'i'} \|\boldsymbol{x}_{u'i'} - \boldsymbol{x}_{ui}\|_2^{\alpha} \\
& \leq \sup_{A_j} c_1\sqrt{K}\Big((nm)^{-1} \sum_{u,i} (\theta_{ui} - \theta^0_{ui})^2\Big)^{1/2} \Big((nm)^{-1} \sum_{u,i} \big(\sum_{(u',i')\in\Omega} 2\omega_{ui,u'i'} \|\boldsymbol{x}_{u'i'} - \boldsymbol{x}_{ui}\|_2^{\alpha}\big)^2\Big)^{1/2} \\
& \leq 2c_1\sqrt{K}\kappa_1 2^j\eta.
\end{aligned}$$

Thus, $I_j \leq P^*\Big(\sup_{A_j} (nm)^{-1} \sum_{u,i} \sum_{(u',i')\in\Omega} 2\omega_{ui,u'i'}(\theta_{ui} - \theta^0_{ui})\epsilon_{u'i'} \geq (2^{j-1}\eta)^2 - 2c_1\sqrt{K}\kappa_1 2^j\eta - \lambda J(\boldsymbol{\theta}^0)\Big)$. Note that $\lambda J(\boldsymbol{\theta}^0) \leq \eta^2/4$ and $\text{RMSE}(\boldsymbol{\theta}, \boldsymbol{\theta}^0) = \Big(\sum_{u,i} (\theta_{ui} - \theta^0_{ui})^2\Big)^{1/2} \geq 2^{j-1}\eta$ for

$\boldsymbol{\theta} \in A_j$. For any $\eta \geq 2^4 c_1 \sqrt{K}\kappa_1$, $(2^{j-1}\eta)^2 - 2c_1\sqrt{K}\kappa_1 2^j \eta - \lambda J(\boldsymbol{\theta}^0) \geq 2^{2j-4}\eta^2$, implying that

$$\begin{aligned}
I_j &\leq P^*\Big(\sup_{A_j}(nm)^{-1}\sum_{u,i}(\theta_{ui}-\theta^0_{ui})\sum_{(u',i')\in\Omega}\omega_{ui,u'i'}\epsilon_{u'i'} \geq 2^{2j-4}\eta^2\Big)\\
&\leq P^*\Big(\sup_{A_j}(nm)^{-1}\Big(\sum_{u,i}(\theta_{ui}-\theta^0_{ui})^2\Big)^{1/2}\Big(\sum_{u,i}\big(\sum_{(u',i')\in\Omega}\omega_{ui,u'i'}\epsilon_{u'i'}\big)^2\Big)^{1/2} \geq 2^{2j-4}\eta^2\Big)\\
&\leq P^*\Big(\Big((nm)^{-1}\sum_{u,i}\big(\sum_{(u',i')\in\Omega}\omega_{ui,u'i'}\epsilon_{u'i'}\big)^2\Big)^{1/2} \geq 2^{j-4}\eta\Big)\\
&\leq P^*\Big(\max_{u,i}\big|\sum_{(u',i')\in\Omega}\omega_{ui,u'i'}\epsilon_{u'i'}\big| \geq 2^{j-4}\eta\Big)\\
&\leq \sum_{u,i} 2\exp\Big(-\frac{2^{(2j-8)}\eta^2}{2\sigma^2\sum_{(u',i')\in\Omega}\omega^2(\boldsymbol{x}_{ui},\boldsymbol{x}_{u'i'})}\Big) \leq 2nm\exp\Big(-\frac{2^{(2j-8)}\eta^2}{2\sigma^2\kappa_2}\Big),
\end{aligned}$$

where the second to the last inequalities follow from the Chernoff inequality of a weighted sub-Gaussian distribution (Chung and Lu, 2006) and Assumption B. Hence, there exist some positive constants a_2 and a_3, for $\eta \geq 2^4 c_1\sqrt{K}\kappa_1$ such that

$$\begin{aligned}
&P\big(\mathrm{RMSE}(\hat{\boldsymbol{\theta}},\boldsymbol{\theta}^0) \geq \eta\big) \leq 4nm\sum_{j=1}^{\infty}\exp\Big\{-\frac{2^{(2j-8)}\eta^2}{2\sigma^2\kappa_2}\Big\}\\
&\leq 4nm\exp\{-a_2\frac{\eta^2}{\sigma^2\kappa_2}\}/(1-\exp\{-a_2\frac{\eta^2}{\sigma^2\kappa_2}\}) \leq a_3\exp\Big\{-a_2\frac{\eta^2}{\sigma^2\kappa_2}+\log(nm)\Big\}.
\end{aligned}$$

The desired result then follows immediately. □

Proof of Corollary 1. It suffices to compute κ_1 and κ_2. First, we will show that for any $\boldsymbol{x}_{ui} \in \mathcal{X}$, when $|\Omega|$ is sufficiently large, there exist positive constants a_4–a_7 such that

$$\begin{aligned}
\sum_{(u',i')\in\Omega}\mathcal{K}_h(\|\boldsymbol{x}_{u'i'}-\boldsymbol{x}_{ui}\|_2)S^{ui}_{u'i'} &\geq \frac{|\Omega|}{2}E\big(\mathcal{K}_h(\|\boldsymbol{x}-\boldsymbol{x}_{ui}\|_2)S^{ui}\big|\Delta=1\big)\\
&= \frac{|\Omega|}{2}P(S^{ui}=1|\Delta=1)E\big(\mathcal{K}_h(\|\boldsymbol{x}-\boldsymbol{x}_{ui}\|_2)\big|S^{ui}=1,\Delta=1\big)\\
&\geq a_4|\Omega|E\big(\mathcal{K}_h(\|\boldsymbol{x}-\boldsymbol{x}_{ui}\|_2)\big|S^{ui}=1,\Delta=1\big)\\
&\geq a_5|\Omega|\int\mathcal{K}_h(u)f_{U^{ui}|S^{ui}=1,\Delta=1}(u)du \geq a_6|\Omega|h\int\mathcal{K}(u)du\\
&\geq a_7|\Omega|h,
\end{aligned}$$

where the $f_{U^{ui}|S^{ui}=1,\Delta=1}$ is the conditional density for U^{ui}, the first inequality follows from the law of large numbers, and the third inequality and the third to the last inequalities follow from Assumption C. Similarly, for some positive constants a_8 and a_9,

$$\begin{aligned}
\sum_{(u',i')\in\Omega}\mathcal{K}_h(\|\boldsymbol{x}_{ui}-\boldsymbol{x}_{u'i'}\|_2)\|\boldsymbol{x}_{ui}-\boldsymbol{x}_{u'i'}\|^{\alpha}S^{ui}_{u'i'} &\leq 2|\Omega|E\big(\mathcal{K}_h(U^{ui})(U^{ui})^{\alpha}|S^{ui}=1,\Delta=1\big)\\
&= 2|\Omega|\int\mathcal{K}_h(u)u^{\alpha}f_{U^{ui}|S^{ui}=1,\Delta=1}(u)du \leq a_8|\Omega|h^{\alpha+1},
\end{aligned}$$

where the last inequality follows from Assumptions C and D, and

$$\sum_{(u',i')\in\Omega} \mathcal{K}_h^2(\|\boldsymbol{x}_{ui} - \boldsymbol{x}_{u'i'}\|_2) S_{u'i'}^{ui} \leq 2|\Omega| E\big(\mathcal{K}_h^2(U^{ui}) | S^{ui} = 1, \Delta = 1\big) \leq a_9 |\Omega| h.$$

Combing the above inequalities yields that

$$\sum_{(u',i')\in\Omega} \omega(\boldsymbol{x}_0, \boldsymbol{x}_{u'i'}) \|\boldsymbol{x}_0, \boldsymbol{x}_{u'i'}\|_2^\alpha = \frac{\sum_{(u',i')\in\Omega} \mathcal{K}_h(\|\boldsymbol{x}_0 - \boldsymbol{x}_{u'i'}\|_2) \|\boldsymbol{x}_0 - \boldsymbol{x}_{u'i'}\|_2^\alpha S_{u'i'}^{ui}}{\sum_{(u',i')\in\Omega} \mathcal{K}_h(\|\boldsymbol{x} - \boldsymbol{x}_{u'i'}\|_2) S_{u'i'}^{ui}} \leq a_8 h^\alpha.$$

Furthermore,

$$\sum_{(u',i')\in\Omega} \omega^2(\boldsymbol{x}_0, \boldsymbol{x}_{u'i'}) \leq \frac{\sum_{(u',i')\in\Omega} \mathcal{K}_h^2(\|\boldsymbol{x}_0 - \boldsymbol{x}_{u'i'}\|_2) S_{u'i'}^{ui}}{\big(\sum_{(u',i')\in\Omega} \mathcal{K}_h(\|\boldsymbol{x} - \boldsymbol{x}_{u'i'}\|_2) S_{u'i'}^{ui}\big)^2} \leq \frac{a_9}{a_7^2 |\Omega| h}.$$

Consequently, $\kappa_1 = h^\alpha$ and $\kappa_2 = (|\Omega| h)^{-1}$, then the desired result follows immediately. □

References

Deepak Agarwal, Liang Zhang, and Rahul Mazumder. Modeling item-item similarities for personalized recommendations on Yahoo! front page. *Annals of Applied Statistics*, 5(3): 1839–1875, 2011.

Stephen H Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. Hinge-loss markov random fields and probabilistic soft logic. *Journal of Machine Learning Research*, 18 (109):1–67, 2017.

Robert M Bell and Yehuda Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *7th IEEE International Conference on Data Mining (ICDM 2007)*, pages 43–52. IEEE, 2007.

Xuan Bi, Annie Qu, Junhui Wang, and Xiaotong Shen. A group-specific recommender system. *Journal of the American Statistical Association*, 112(519):1344–1353, 2017.

Patrick Billingsley. *Convergence of probability measures.* John Wiley & Sons, 2013.

Daniel Billsus and Michael J Pazzani. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2-3):147–180, 2000.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.

Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35 (3):283–319, 1970.

Bilian Chen, Simai He, Zhening Li, and Shuzhong Zhang. Maximum block improvement and polynomial optimization. *SIAM Journal on Optimization*, 22(1):87–107, 2012.

Hsin Chen and Alan F Murray. Continuous restricted Boltzmann machine with an implementable training algorithm. *IEE Proceedings-Vision, Image and Signal Processing*, 150 (3):153–158, 2003.

Tianle Chen, Yuanjia Wang, Huaihou Chen, Karen Marder, and Donglin Zeng. Targeted local support vector machine for age-dependent classification. *Journal of the American Statistical Association*, 109(507):1174–1187, 2014.

Fan Chung and Linyuan Lu. Concentration inequalities and martingale inequalities: A survey. *Internet Mathematics*, 3(1):79–127, 2006.

Aurore Delaigle and Peter Hall. Defining probability density for a distribution of random functions. *Annals of Statistics*, 38(2):1171–1193, 2010.

Özgür Demir, Alexey Rodriguez Yakushev, Rany Keddo, and Ursula Kallio. Item-item music recommendations with side information. *CoRR*, abs/1706.00218, 2017. URL `http://arxiv.org/abs/1706.00218`.

Jianqing Fan and Irene Gijbels. *Local polynomial modelling and its applications: monographs on statistics and applied probability*, volume 66. CRC Press, 1996.

Andrey Feuerverger, Yu He, and Shashi Khatri. Statistical significance of the Netflix challenge. *Statistical Science*, 27(2):202–231, 2012.

Peter Forbes and Mu Zhu. Content-boosted matrix factorization for recommender systems: Experiments with recipe recommendation. In *Proceedings of the 5th ACM conference on Recommender Systems*, pages 261–264. ACM, 2011.

Jerome H Friedman and Werner Stuetzle. Projection pursuit regression. *Journal of the American Statistical Association*, 76(376):817–823, 1981.

Quanquan Gu, Jie Zhou, and Chris Ding. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 199–210. SIAM, 2010.

Trevor Hastie, Rahul Mazumder, Jason D Lee, and Reza Zadeh. Matrix completion and low-rank svd via fast alternating least squares. *Journal of Machine Learning Research*, 16(1):3367–3402, 2015.

Thomas Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 22(1):89–115, 2004.

Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 426–434. ACM, 2008.

Pigi Kouki, Shobeir Fakhraei, James Foulds, Magdalini Eirinaki, and Lise Getoor. Hyper: A flexible and extensible probabilistic framework for hybrid recommender systems. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 99–106. ACM, 2015.

Oluwasanmi Koyejo and Joydeep Ghosh. A kernel-based approach to exploiting interaction-networks in heterogeneous information sources for improved recommender systems. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, pages 9–16. ACM, 2011.

Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*, pages 331–339, 1995.

Huazhen Lin, Lixian Pan, Shaogao Lv, and Wenyang Zhang. Efficient estimation and computation for the generalised additive models with unknown link function. *Journal of Econometrics*, 2017.

Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

Yufeng Liu and Xiaotong Shen. Multicategory ψ-learning. *Journal of the American Statistical Association*, 101(474):500–509, 2006.

Shujie Ma and Jian Huang. A concave pairwise fusion approach to subgroup analysis. *Journal of the American Statistical Association*, 112(517):410–423, 2017.

JS Marron and WJ Padgett. Asymptotically optimal bandwidth selection for kernel density estimators from randomly right-censored samples. *Annals of Statistics*, pages 1520–1535, 1987.

Judith Masthoff. Group recommender systems: Combining individual models. In *Recommender systems handbook*, pages 677–702. Springer, 2011.

Prem Melville, Raymond J Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. *AAAI/IAAI*, 23:187–192, 2002.

Stuart E Middleton, Nigel R Shadbolt, and David C De Roure. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):54–88, 2004.

Bradley N Miller, Istvan Albert, Shyong K Lam, Joseph A Konstan, and John Riedl. Movielens unplugged: Experiences with an occasionally connected recommender system. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, pages 263–266. ACM, 2003.

A Mukherjee, K Chen, N Wang, and J Zhu. On the degrees of freedom of reduced-rank estimators in multivariate regression. *Biometrika*, 102(2):457–477, 2015.

Jennifer Nguyen and Mu Zhu. Content-boosted matrix factorization techniques for recommender systems. *Statistical Analysis and Data Mining*, 6(4):286–301, 2013.

Trong T Nguyen and Hady W Lauw. Representation learning for homophilic preferences. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 317–324. ACM, 2016.

Arkadiusz Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD Cup and Workshop*, pages 5–8, 2007.

Michael J Pazzani, Jack Muramatsu, Daniel Billsus, et al. Syskill & Webert: Identifying interesting web sites. In *AAAI/IAAI*, volume 1, pages 54–61, 1996.

Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62 (1-2):107–136, 2006.

Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine Learning*, pages 880–887. ACM, 2008.

Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine learning*, pages 791–798. ACM, 2007.

David W Scott. *Multivariate density estimation: theory, practice, and visualization.* John Wiley & Sons, 2015.

Xiaotong Shen, George C Tseng, Xuegong Zhang, and Wing Hung Wong. On ψ-learning. *Journal of the American Statistical Association*, 98(463):724–734, 2003.

Nathan Srebro, Noga Alon, and Tommi S Jaakkola. Generalization error bounds for collaborative prediction with low-rank matrices. In *Advances in Neural Information Processing Systems*, pages 1321–1328, 2005.

Charles J Stone. An asymptotically optimal window selection rule for kernel density estimates. *Annals of Statistics*, pages 1285–1297, 1984.

Robert Tibshirani and Trevor Hastie. Local likelihood estimation. *Journal of the American Statistical Association*, 82(398):559–567, 1987.

Philippe Vieu. Nonparametric regression: optimal local bandwidth choice. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 453–464, 1991.

Larry Wassermann. *All of nonparametric statistics.* New York: Springer, 2006.

Wing Hung Wong and Xiaotong Shen. Probability inequalities for likelihood ratios and convergence rates of sieve MLEs. *Annals of Statistics*, 23(2):339–362, 1995.

Shuang-Hong Yang, Bo Long, Alex Smola, Narayanan Sadagopan, Zhaohui Zheng, and Hongyuan Zha. Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the 20th International Conference on World Wide Web*, pages 537–546. ACM, 2011.

Feipeng Zhao and Yuhong Guo. Learning discriminative recommendation systems with side information. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3469–3475, 2017.

Feipeng Zhao, Min Xiao, and Yuhong Guo. Predictive collaborative filtering with side information. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 2385–2391, 2016.

Tinghui Zhou, Hanhuai Shan, Arindam Banerjee, and Guillermo Sapiro. Kernelized probabilistic matrix factorization: Exploiting graphs and side information. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 403–414. SIAM, 2012.

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