



# Social recommendation via multi-view user preference learning



Hanqing Lu, Chaochao Chen, Ming Kong, Hanyi Zhang, Zhou Zhao\*

Zhejiang University, Hangzhou 310027, China

## ARTICLE INFO

### Article history:

Received 2 November 2015

Received in revised form

21 April 2016

Accepted 8 July 2016

Communicated by Guan Ziyu

Available online 21 July 2016

### Keywords:

Microblogging system

Social recommendation

Multi-view user preference learning

## ABSTRACT

Recommender system (RS) has become an active research area driven by the enormous industrial demands. Meanwhile, with the rapid development of microblogging system, various kinds of social data are available, which provide opportunities as well as challenges for traditional RSs. In this paper, we introduce the social recommendation (SR) problem utilizing microblogging data. We study this problem via multi-view user preference learning. Specifically, we first model user preference by learning a low-dimensional common representation of multi-view information including rating information, social relations, item side information, tagging information, and then recommend items based on the learnt user preference. We also develop an efficient alternating direction method of multipliers (ADMM) scheme to solve the proposed model. We empirically evaluate our approach using two real world datasets to demonstrate the significant improvement of our proposed approach against the state-of-the-art algorithms.

© 2016 Elsevier B.V. All rights reserved.

## 1. Introduction

With the rapid growth of available information on the Internet, users are getting into trouble with the information overload problem. Recommender systems (RS) [1] intend to help user select interesting information by mining user preference, which has been widely applied to many areas such as e-commerce, online reading, review websites, and so on. Meanwhile, microblogging system is witnessing a dramatic change in the recent years. Social information from microblogging system provides diverse sources for recommendation beyond explicit rating information, which provides opportunities as well as challenges for traditional RSs.

Social recommendation (SR) has attracted increasing attention in recent years. Under the broad definition, any recommender systems that target social media domains could be considered as SR [2]. Current SR could be divided into two categories, *content-based* SR [3–5] and *collaborative filtering (CF)*-based SR [6,7]. Content-based SR incorporates textual information to help the recommendation procedure, and most of them focus on recommending items with textual information. By contrast, CF-based SR integrates social information in traditional CF frameworks. Currently, most of the existing CF-based SRs add social regularization terms on rating regression objectives to improve recommendation performance [8]. Besides, there are some

hybrid models that combine content-based SR with CF-based SR [9,10]. For example, MR3 [10] intends to incorporate social relations and item reviews together with rating information to improve the recommendation performance.

In this paper, we introduce a new SR problem utilizing microblogging data. As is known to all, microblogging system such as Sina Weibo<sup>1</sup> contains a large number of personal information that can reflect user's preference. However, in the past, it is really hard to bridge the information from microblogging system and review site. As a consequence, few work has been done to incorporate microblogging data into recommender system. Fortunately, the rapid development of Internet provides the possibility of utilizing these information to enhance the SR performance. For example, we observe that many users on review site such as Douban<sup>2</sup> have a microblogging system accounts, which enable us to obtain their social relations and tagging information from microblogging system, and then integrate these information into our SR framework. We study this problem from the perspective of multi-view user preference learning, where *multi-view* means various description of a user's preference. We define the following two views in SR problem:  $\mathcal{V}1$ . **social view**, i.e., the microblogging system;  $\mathcal{V}2$ . **recommend view**, i.e., the review site. The multi-view learning process is to learn the common representation of the social view information and recommend view information. Previous works have shown that tagging information reflects user's

\* Corresponding author.

E-mail addresses: [lhq110@zju.edu.cn](mailto:lhq110@zju.edu.cn) (H. Lu), [zjucc@zju.edu.cn](mailto:zjucc@zju.edu.cn) (C. Chen), [zjukongming@zju.edu.cn](mailto:zjukongming@zju.edu.cn) (M. Kong), [3130102345@zju.edu.cn](mailto:3130102345@zju.edu.cn) (H. Zhang), [zhaozhou@zju.edu.cn](mailto:zhaozhou@zju.edu.cn) (Z. Zhao).

<sup>1</sup> <http://www.sinaweibo.com>

<sup>2</sup> <http://www.douban.com>

preference in social view [3,11,12]. However, the free-form of tags results in tag redundancy and ambiguity, and thus the tagging matrix can be considered as a low-rank matrix. Therefore, user preference in social view can be regarded as a low-dimensional representation of tagging information. Similarly, user preference in recommendation view can be regarded as a low-dimensional representation of rating information. There is strong evidence proving that information from both two views have a strong correlation [13,12,3]. For example, a user with a personal tag 'Geek' may be interested in technology. As a result, he may like movie 'Interstellar'. Therefore, we assume these two views share the same low-dimensional representation, i.e. user preference. We model user preference by learning a common low-dimensional representation of information from both views. We employ a multi-view learning style method [14–16] in our SR framework to bridge tagging information and rating information. As shown in Fig. 1, the user preference matrix is learnt by factorizing the rating matrix from recommend view and the tag matrix from social view simultaneously. Particularly, we utilize item side information from recommend view to accelerate the learning process. That is, we extract item textual description, and then represent it using bag-of-words model [17,18], which can be regarded as side information of user preference. Then we perform the matrix completion with side information (MCWSI) [19,20,8] to complete the rating matrix, which can significantly reduce the computing complexity of low-rank matrix completion.

Furthermore, social relations from social view between two users provide a strong evidence for them to have similar preference, which can be used as another regularization term named graph regularizer [8,7]. We employ it in our framework to user preference modeling process.

It is worthwhile to highlight the main contributions of this paper as follows:

- We introduce a new social recommendation problem by utilizing microblogging data.
- We propose a novel SR model termed multi-view user preference

learning (MVUPL), which utilizes rating information, item side information, user social relations, and tagging information.

- We present an efficient alternating direction method of multipliers (ADMMs) scheme to solve the introduced model.
- We evaluate the proposed model extensively on two real-world datasets, showing that the proposed model outperforms several state-of-the-art SR approaches.

The remainder of this paper is organized as follows. Section 2 gives a brief review on related work. Section 3 describes the learning process of the new social recommendation problem, and then introduces the MVUPL model. The solution of our proposed model is also given in Section 3. Section 4 presents experiments and results. Finally, conclusions are given in Section 5.

## 2. Related work

In this section, we present the most related literature to our work, including matrix completion, matrix factorization, multi-view learning, and recommender system.

### 2.1. Matrix completion and matrix factorization

Matrix factorization (MF) [21,22] is a classical kind of approach for recommender systems. The basic idea of MF is to factorize the rating matrix into two low rank matrices, i.e., user preference matrix and item feature matrix respectively. Then the two matrices to approximate are multiplied to the original rating matrix. Matrix completion (MC) is the method of filling in unknown entries in an uncompleted matrix [23]. MC has been widely applied to recommender system because the rating matrix is often incomplete. Most MC algorithms assume that the incomplete matrix is low rank [24,25]. The rank minimization problem is NP-hard, thus, it cannot be applied to practical applications. However,

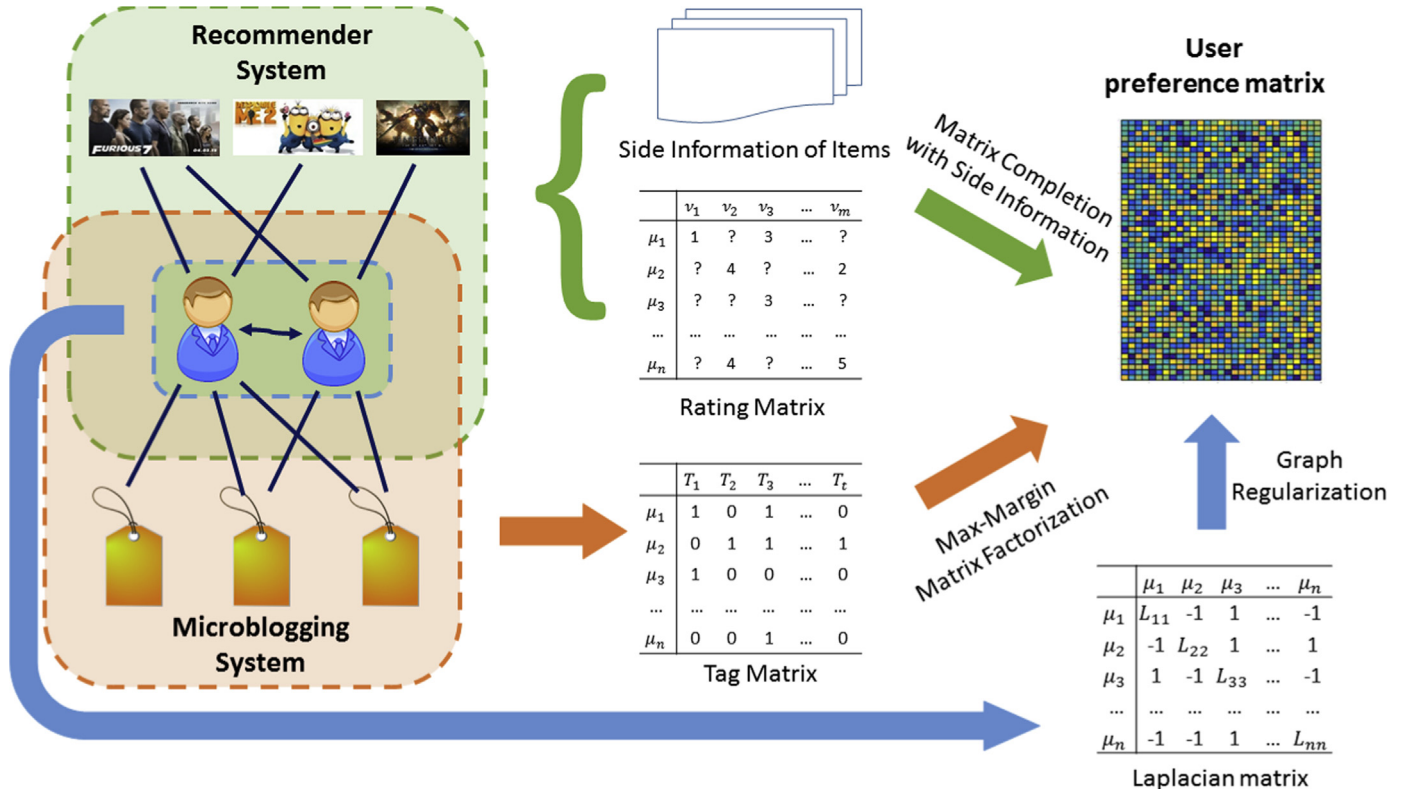


Fig. 1. MVUPL model.

theoretical studies [26] show that the rank function of matrices has a tightest convex lower bound named *nuclear norm*, i.e., the sum of singular values of the matrix. Therefore, a widely used method is to ensure the low rank constraints by minimizing the nuclear norm. Specifically, given a matrix  $Y \in \mathbb{R}^{n \times m}$  with low rank, the matrix completion problem is given as

$$\underset{M}{\operatorname{argmin}} \|M\|_* \quad \text{s. t. } Y_{ij} = M_{ij}, (i, j) \in \Omega, \quad (1)$$

where  $M \in \mathbb{R}^{n \times m}$  and  $\Omega$  is the set of the observed entries, and  $\|\cdot\|_*$  denotes the nuclear norm. Some recent research works have integrated the side information in MC to make it more efficient [19,20]. Specifically, given a rating matrix  $Y \in \mathbb{R}^{n \times m}$  and side information matrix  $X \in \mathbb{R}^{n \times k}$ , the optimization problem of matrix completion with side information is given as follows:

$$\underset{W}{\operatorname{argmin}} \|W\|_* \quad \text{s. t. } Y_{ij} = W_i X_j^T, (i, j) \in \Omega, \quad (2)$$

where  $W \in \mathbb{R}^{n \times k}$ . Unlike the standard algorithm for matrix completion that requires solving an optimization problem involved the rating matrix  $Y$  of  $n \times m$ , the optimization problem given in Eq. (2) only deals with the use preference matrix with  $n \times k$ . Therefore it can be solved more efficiently since  $k \ll m$  [19,8]. Our work differs from the aforementioned studies in that we utilize the microblogging data to develop a multi-view learning method.

## 2.2. Multi-view learning

Multi-view learning is an active research topic [14–16], where *multi-view* means various descriptions of a given sample. MCL [27] is a nonnegative matrix factorization (NMF)-based method which synthesize multi-view features to improve the data representation process. IrMVL [28] is a multi-view learning algorithm that seeks a low-dimensional common representation of all views. This algorithm based on the assumption that the common representation of all views is low-rank. Intensive experiments have been done to show that these approaches are effective. However, these multi-view learning approaches are unsuitable for our social recommendation framework and usually underperforms since they learn the latent factor of multi-view information by using the same technique as NMF or MF. It is often the case that information from different views are heterogeneous. As a result, it is more suitable to employ different factorization techniques in the framework in order to reach a better performance. For example, the latent factor of different tagging information should be learnt by using max-margin matrix factorization because tagging information only contains 0 and 1, while rating information should be learnt by using basic matrix factorization since it varies in a certain range, e.g., [0, 5]. Thus, we modify the traditional multi-view learning methods and employ it in our SR framework.

## 2.3. Recommender system

Collaborative filtering (CF) has become an indispensable component in traditional recommender systems. The intuition of CF is that users who have similar tastes may show similar preferences on the same item [29,30]. However, performance of tradition recommender systems may be affected by the data sparsity problem, i.e., only few user-item behavior relations are known [31]. In order to solve the data sparsity problem, many methods adopt auxiliary data to boost the performance of recommendation [32,33]. It is also worth noting that there are several research works that exploit the same information as our proposed method, i.e., rating information, tagging information and content information. For example, MIOE [34] represents users, tags, and documents in the same semantic space in order to conduct document recommendation. Besides, GRoMO [35] generalizes the manifold ranking technique to heterogeneous manifolds and

addresses personalized tag recommendation by modeling the tagging data as a heterogeneous graph. Our approach distinguishes from these methods in that they adopt tagging information from social tagging system while we employ tagging information from microblogging system, and these two systems have different kinds of tagging information. Tags on microblogging system are personal tags that describe users themselves, while tags on social tagging system are used by user to describe items. Therefore, using personal tags is relatively a more reasonable solution for modeling user preference. For example, A user with a personal tag ‘Geek’ may indicate that this user is interested in technology. As a result, he may like the movie ‘Interstellar’. On the contrary, a user adds a tag ‘Romantic’ to the movie ‘Titanic’ does not necessarily mean that he has a special affinity for romantic movies. He adds ‘Romantic’ to the ‘Titanic’ merely because he takes ‘Titanic’ as a romantic movie.

## 3. Social recommendation via multi-view user preference learning

In this section, we first introduce some notions for social recommendation utilizing microblogging data, which are the observed rating matrix  $Y$ , the side item matrix  $X$ , the tagging matrix  $T$ , the Laplacian matrix  $D$ . Then we formally present the social recommendation problem via multi-view user preference learning and provide the optimization algorithm to solve the problem. Finally, we conduct a time complexity analysis for the proposed algorithm.

### 3.1. Notations

Suppose there are  $n$  users  $U = \{\mu_1, \dots, \mu_n\}$ ,  $m$  items  $V = \{\nu_1, \dots, \nu_m\}$ ,  $v$  tags  $N = \{\zeta_1, \dots, \zeta_v\}$ . We denote the observed rating matrix by  $Y \in \mathbb{R}^{n \times m}$ . We denote the tag matrix by  $T \in \mathbb{R}^{n \times v}$  where  $T_{ij} = 1$  means that user  $\mu_i$  owns tag  $\zeta_j$ . We use  $A \in \mathbb{R}^{n \times n}$  to denote the relation matrix of user  $U$ , where  $A_{ij} = 1$  if and only if user  $u_i$  and  $u_j$  has a social relation. Let  $K$  be the diagonal matrix with  $K_{ii} = \sum_{j=1}^n A_{ij}$ , and  $D \in \mathbb{R}^{n \times n} = K - A$  be the Laplacian matrix. We consider that the textual description of item is the side information for user preference, denoting by  $X \in \mathbb{R}^{m \times k}$ . We extract the textual description of item, and then represent it by using *bag-of-words* model [17,18].

### 3.2. The problem

We now describe the learning problem for social recommendation. Given side information of items  $X \in \mathbb{R}^{m \times k}$ , the incomplete rating matrix  $Y \in \mathbb{R}^{n \times m}$ , the tagging matrix  $T \in \mathbb{R}^{n \times v}$ , and the Laplacian matrix  $D \in \mathbb{R}^{n \times n}$ , the learning process is to learn a user preference matrix  $W \in \mathbb{R}^{n \times k}$  and use it to complete the rating matrix  $Y \in \mathbb{R}^{n \times m}$ . For example, the prediction for entry  $Y_{ij}$  based on user preference  $W$  is given by  $\hat{Y}_{ij} = W_i X_j^T$ , where  $W_i \in \mathbb{R}^{1 \times k}$  is the preference vector of  $u_i$  and  $X_j \in \mathbb{R}^{k \times 1}$  is the feature vector of  $\zeta_j$ .

We start from learning user preference in recommend view, where we only consider the side information of items  $X$  and the incomplete rating matrix  $Y$ . Note that, the preference of different users may be correlated. For example, two users with similar background are likely to be interested in similar items, so it is natural to assume that the user preference matrix  $W$  is low rank. Consequently, we cast the social recommendation problem into the optimization problem of matrix completion with side information [8,19].

$$\underset{W}{\operatorname{argmin}} \|W\|_* \quad \text{s. t. } P \otimes Y = P \otimes W X^T, \quad (3)$$

where  $\otimes$  represents the Hadamard element-wise product and  $P \in \mathbb{R}^{n \times m}$  denotes the indicator matrix with ones for the observed rating, and zeros for the missing values. By requiring  $P \otimes Y = P \otimes WX^T$ , we expect that the prediction for all the observed ratings are accurate. However, we notice that the values in the rating matrix  $Y$  are noisy because the rating of user is often inaccurate. Thus, the hard constraint in Eq. (3) is not robust due to the noise in  $Y$ . To overcome this weakness, we introduce the following optimization problem:

$$\operatorname{argmin}_W \|P \otimes (Y - WX^T)\|_F^2 + \tau \|W\|_* \quad (4)$$

The regularization coefficient  $\tau$  is employed to balance the weight between the data penalty term  $\|P \otimes (Y - WX^T)\|_F^2$  and the regularization term  $\|W\|_*$ , which is usually set empirically.

Next, we take microblogging data into account. User preference in social view can be regarded as a low-dimensional representation of tagging information, and there is a strong evidence which shows that information from both views have a strong correlation [13]. Therefore, we assume that these two views share the same low-dimensional representation, i.e. the user preference matrix  $W$ . As a result, the multi-view learning process equals to seek a low-dimensional common representation, i.e.,  $W$ , of the two matrix  $Y$  and  $T$ . Following this thought, we can decompose the tagging matrix  $T$  into two matrices  $W$  and  $B$ , where  $B \in \mathbb{R}^{v \times k}$  represents the latent factor of tagging information. We also observe that the tagging matrix  $T$  is a binary matrix. In order to make our model more robust, we employ the max-margin matrix factorization (MMMF) on the tagging matrix  $T$ , and cast the social recommendation problem into an optimization problem as follows:

$$\operatorname{argmin}_{W,B} \|P \otimes (Y - WX^T)\|_F^2 + \alpha H(W, B) + \beta \|B\|_F^2 + \tau \|W\|_*, \quad (5)$$

where  $H(W, B) = \sum_{i=1}^n \sum_{j=1}^v \max(0, 1 - T_{ij} W_i B_j^T)$ ,  $B_j \in \mathbb{R}^{v \times 1}$  represents the  $j$ -th column of  $B$ , and  $\alpha$  balances the weight between tagging information and rating information. The social relation between two users provides a strong evidence for them to have common bias, which can be taken as another regularization term named graph regularizer, as discussed in [8]. Here we use it to regularize the user preference matrix, and this term can be achieved by minimizing:

$$\begin{aligned} \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n A_{ij} \|W_i - W_j\|_F^2 &= \operatorname{tr}(W^T K W) - \operatorname{tr}(W^T A W) \\ &= \operatorname{tr}(W^T D W), \end{aligned} \quad (6)$$

where  $\operatorname{tr}(\cdot)$  denotes the trace of matrix. Based on Eqs. (5) and (6), we have the following objective function for our proposed model:

$$\begin{aligned} \operatorname{argmin}_{W,B} \|P \otimes (Y - WX^T)\|_F^2 + \alpha H(W, B) \\ + C \operatorname{tr}(W^T D W) + \beta \|B\|_F^2 + \tau \|W\|_*, \end{aligned} \quad (7)$$

where  $\alpha, C, \beta, \tau$  are hyperparameters.

### 3.3. The optimization

A lot of optimization methods such as R-GLM [36] require the loss function to be differentiability. However, both hinge loss and the nuclear norm in our model are not smooth, and thus we come to derive a learning algorithm based on ADMM [37,38]. The problem in Eq. (7) can be equivalently written as follows:

$$\operatorname{argmin}_{W,B,S} F(W, B) + G(S), \text{ s. t. } S - W = 0, \quad (8)$$

where

$$\begin{aligned} F(W, B) &= \|P \otimes (Y - WX^T)\|_F^2 + \alpha H(W, B) \\ &\quad + C \operatorname{tr}(W^T D W) + \beta \|B\|_F^2, \end{aligned}$$

and  $G(S) = \tau \|S\|_*$ .

We solve Eq. (8) by using augmented Lagrangian function

$$L(W, S, \Lambda, B) = F(W, B) + G(S) + \operatorname{tr}(\Lambda^T (S - W)) + \frac{\rho}{2} \|S - W\|_F^2, \quad (9)$$

where  $\rho > 0$  is a hyperparameter.

The ADMM learning procedure for our MVUPL is in Algorithm 1. The key steps for our algorithm are updating  $S, W, B$ , and  $\Lambda$ . Nuclear norm and hinge loss function are not smooth, which make it difficult to calculate the update values. But they do have sub-gradient, and thus we can solve it by the subgradient method, which has been discussed a lot in previous paper [37]. The update value can be computed based on Theorems 1 and 2.

#### Algorithm 1. ADMM for MVUPL

**Input:** users' tag matrix  $T$ , Laplacian matrix  $D$ , rating matrix  $Y$ , item feature matrix  $X$ ;

**Output:** the users' preference matrix  $W$ ;

$S^{(0)} \in \mathbb{R}^{n \times k}$ ,  $\Lambda^{(0)} \in \mathbb{R}^{n \times k}$ ,  $\rho > 0$

**for**  $i=1$  to Max iter **do**

Update  $(W^{(k+1)}, B^{(k+1)})$  by Eq. (15)

Update  $S^{(k+1)}$  by Eq. (12)

$\Lambda^{(k+1)} = \Lambda^{(k)} - \rho(W^{(k)} - S^{(k)})$

**end for**

**Return** user preference  $W$

We first introduce a useful tool: the singular value shrinkage operator to estimate the matrix  $S^{(k+1)}$  in MVUPL.

#### 3.3.1. Definition 1

*Singular value thresholding operator:* Consider the singular value decomposition (SVD) of a matrix  $A \in \mathbb{R}^{n \times m}$  of rank  $r$ ,

$$A = U_A \Sigma_A V_A^T = \operatorname{diag}(\sigma_1(A), \dots, \sigma_r(A)),$$

where  $\sigma_i(A)$  is the  $i$ -th biggest singular value of  $A$ . For any  $\tau > 0$ , we define the singular value thresholding (SVT) operator [23,39] as below:

$$\mathcal{D}_\tau(A) = U_A S_\tau(\Sigma_A) V_A^T,$$

where  $S_\tau(\Sigma_A) = \operatorname{diag}([(\sigma_1(A) - \tau)_+], \dots, [(\sigma_r(A) - \tau)_+])$  and  $[z]_+ = \max(z, 0)$ .

Using the SVT operator above, we have the following useful theorems for the composite objective function:

**Theorem 1.** [39] For positive numbers  $\tau$ ,  $S \in \mathbb{R}^{n \times k}$ , and  $W \in \mathbb{R}^{n \times k}$ , we have

$$\mathcal{D}_\tau(W) = \operatorname{argmin}_S \frac{1}{2} \|S - W\|_F^2 + \tau \|S\|_*. \quad (10)$$

**Update  $S^{(k+1)}$ :** Ignoring constant terms, the minimization with respect to  $S^{(k+1)}$  is given by

$$S^{(k+1)} = \operatorname{argmin}_S \frac{1}{2} \left\| \frac{A}{\rho} + S^{(k)} - W^{(k)} \right\|_F^2 + \frac{\tau}{\rho} \|S^{(k)}\|_*. \quad (11)$$

Therefore, we can obtain the closed form solution of  $S^{(k+1)}$  by Theorem 1 as follows:

$$S^{(k+1)} = \mathcal{D}_{\frac{\tau}{\rho}} \left( W^{(k)} - \frac{\Lambda^{(k)}}{\rho} \right). \quad (12)$$



We then introduce the following theorem to estimate the user preference matrix  $W$  and latent factor of tagging information  $B$ .

**Lemma 1.** The subgradient of  $H(W, B)$  is

$$\frac{\partial H(W, B)}{\partial W} = Q, \quad \frac{\partial H(W, B)}{\partial B} = Z, \quad (13)$$

where

$$Q_{pq} = - \sum_{j=1}^m T_{pj} B_{jq} R_{pj}, \quad Z_{pq} = - \sum_{j=1}^m T_{ip} W_{iq} R_{ip},$$

$$p \in (1, 2, \dots, n), \quad q \in (1, 2, \dots, k),$$

and

$$R_{ij} = \begin{cases} 1 & T_{ij} Y_{ij} < 1 \\ 0 & \text{otherwise,} \end{cases}$$

**Theorem 2.** Define

$$F_1(W, S, \Lambda, B) = F(W, B) - \text{tr}(\Lambda^T W) + \frac{\rho}{2} \|S - W\|_F^2$$

By using Lemma 1 we can calculate the gradient of  $F_1(W, B)$ , which is

$$\begin{aligned} \frac{\partial F_1(W, B)}{\partial W} &= 2(P \otimes Y - WX^T)X + \alpha Q + 2CDW - \Lambda + \rho(W - S) \\ \frac{\partial F_1(W, B)}{\partial B} &= \alpha Z + 2\beta B. \end{aligned} \quad (14)$$

Since  $F_1(W, B)$  is convex, implementing the gradient descent upon  $F_1(W, B)$ , we will finally find:

$$(W_*, B_*) = \underset{W, B}{\text{argmin}} F_1(W, B). \quad (15)$$

**Update ( $W_{(k+1)}, B_{(k+1)}$ ).** Since  $\partial L(W, S, \Lambda, B) = \partial F_1(W, B)$  with respect to  $W$  and  $B$ , the update for  $(W_{(k+1)}, B_{(k+1)})$  in ADMM for MVUPL is given as follows:

$$(W^{(k+1)}, B^{(k+1)}) = \underset{W, B}{\text{argmin}} F_1(W, S^{(k)}, \Lambda^{(k)}, B). \quad (16)$$

Since we can obtain  $\underset{W, B}{\text{argmin}} F_1(W, B)$  from Theorem 2, we then write the update rule for  $W$  and  $B$  as follows:

$$(W^{(k+1)}, B^{(k+1)}) = (W_*, B_*). \quad (17)$$

### 3.4. Time complexity

According to Algorithm 1, the time complexity of MVUPL can be divide into two parts. Firstly, updating  $(W^{(k+1)}, B^{(k+1)})$  can be done in  $O(n \times k \times m + 2n \times m \times k + 2n \times k \times m + n \times n \times k)$ . Secondly, updating  $S^{(k+1)}$  need to compute the SVD of a  $n \times k$  matrix. According to [40], its time complexity is  $O(n^2 \times k + k^3)$ . Thus, the over all time complexity is  $O(5n \times m \times k + n \times n \times k + n^2 \times k + k^3)$ . Considering that  $k \ll m, n$ , the complexity can be reduced to quadric time  $O(n \times (n + m))$ .

## 4. Experimental results

In this section, we use several datasets to evaluate the performance of our proposed model. The experiments are conducted by using Matlab, and tested on machines with Linux OS, Intel(R) Xeon (R) Quad CPU 2.10 GHz, and 32 GB RAM.

### 4.1. Experimental setup

The dataset is crawled from Douban and Sina Weibo. Douban is a famous review site in China, which contains both movie reviews and music reviews. We finally crawled 5000 users who have 193,171

ratings on 5000 music on Douban as the music dataset, and 2000 users who have 56,648 ratings on 2000 movies on Douban as the movie dataset. Then, we crawled the comments and brief introduction of these music and movies, and represent it using bag-of-word model [41,17]. We use it as the side information. Sina Weibo is a micro-blogging system in China. We observed that many users in Douban have Sina Weibo account. We first extracted the Sina Weibo account of the Douban users, and then crawled their following relationship and tagging information. In music dataset, there are 51,048 tagging behavior contributed by 5000 users with 6177 distinct tags. In movie dataset, there are 15,267 tagging behavior contributed by 2000 users with 2272 distinct tags.

We randomly select  $x\%$  as the training set and report the prediction performance on the remaining  $1 - x\%$  testing set where  $x$  varied in  $\{10, 30, 50, 70, 90\}$ . We use following metrics to evaluate the rating prediction performances of different models:

Root mean square error (RMSE) and mean average error (MAE):

$$RMSE = \sqrt{\sum_{u_i, v_j \in S} (Y_{ij} - \hat{Y}_{ij})^2 / |S|} \quad (18)$$

$$MAE = \sum_{u_i, v_j \in S} (Y_{ij} - \hat{Y}_{ij}) / |S| \quad (19)$$

where  $Y_{ij}$  and  $\hat{Y}_{ij}$  represent the true and the predicted rating value.  $S$  and  $|S|$  is the test set and its cardinality. RMSE and MAE indicate the different between true rating and predicted rating, and smaller RMSE and MAE means a better prediction performance.

Mean average precision (MAP): Given a top- $N$  recommendation list  $C_{N, rec}$ , the precision is defined as:

$$Precision@N = \frac{|C_{N, rec} \cap C_{adopted}|}{N}, \quad (20)$$

where  $C_{adopted}$  are the items that a user adopted in the test data. Average precision (AP) is a ranked precision metric that gives larger credit to correctly recommended items in top ranks. AP@N is defined as the average of precisions computed at all positions with an adopted item, namely,

$$AP@N = \frac{\sum_{k=1}^N Precision@k \times rel(k)}{\min N, |C_{adopted}|}, \quad (21)$$

where Precision@ $k$  is the precision at cut-off  $k$  in the top- $N$  list  $C_{N, rec}$ , and  $rel(k)$  is an indicator function equals to 1 if the item at rank  $k$  is adopted, and 0 otherwise. Finally, Mean Average Precision (MAP@N) is defined as the mean of the AP scores for all users. In our paper, we mainly show the result of MAP@N with  $N = 5, 10$ , which is widely used in other research works [42]. In order to compute the MAP for the rating matrix  $Y$ , we set that one user adopt one item only if this user gives no less than 4 stars to this item. We get the Top- $N$  recommendation list for users by ranking the predicted ratings of item. This processing method is also widely used in previous work on recommendation [42].

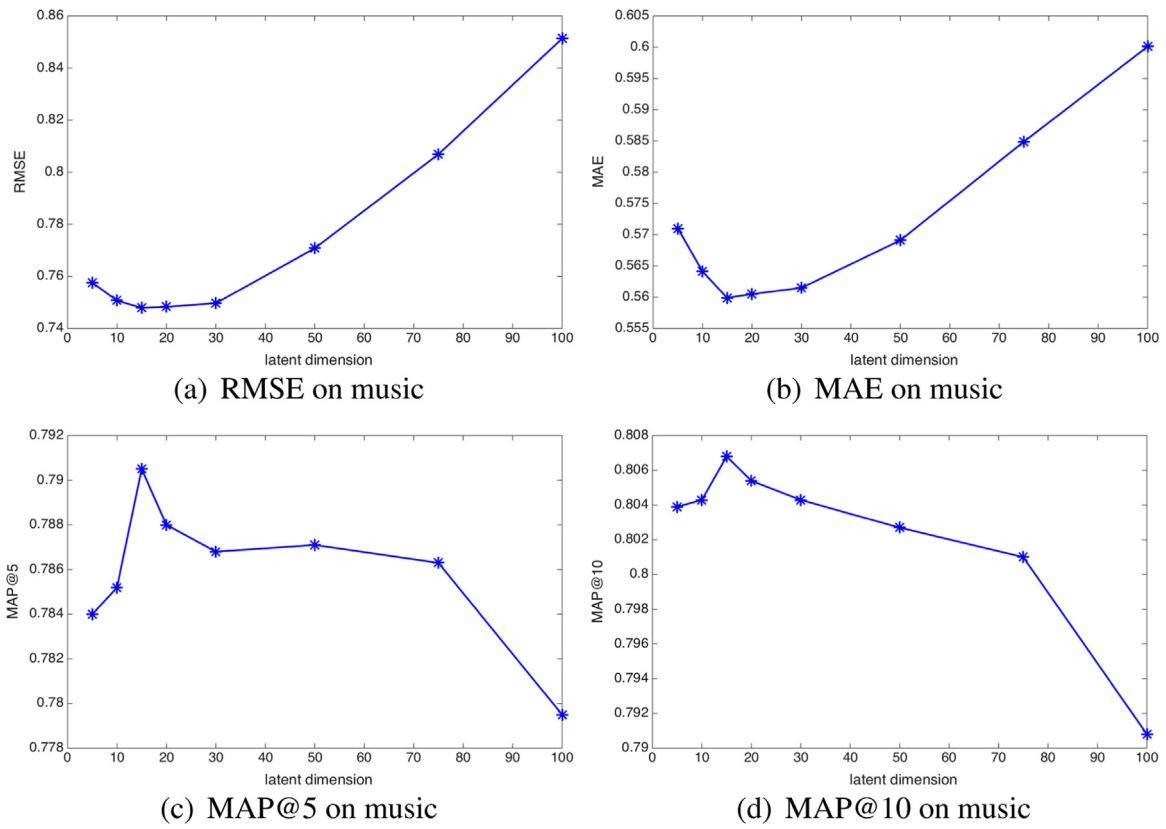
### 4.2. Comparing with different recommender systems

In this subsection, we compare the proposed model MVUPL with the following start-of-the-art baselines:

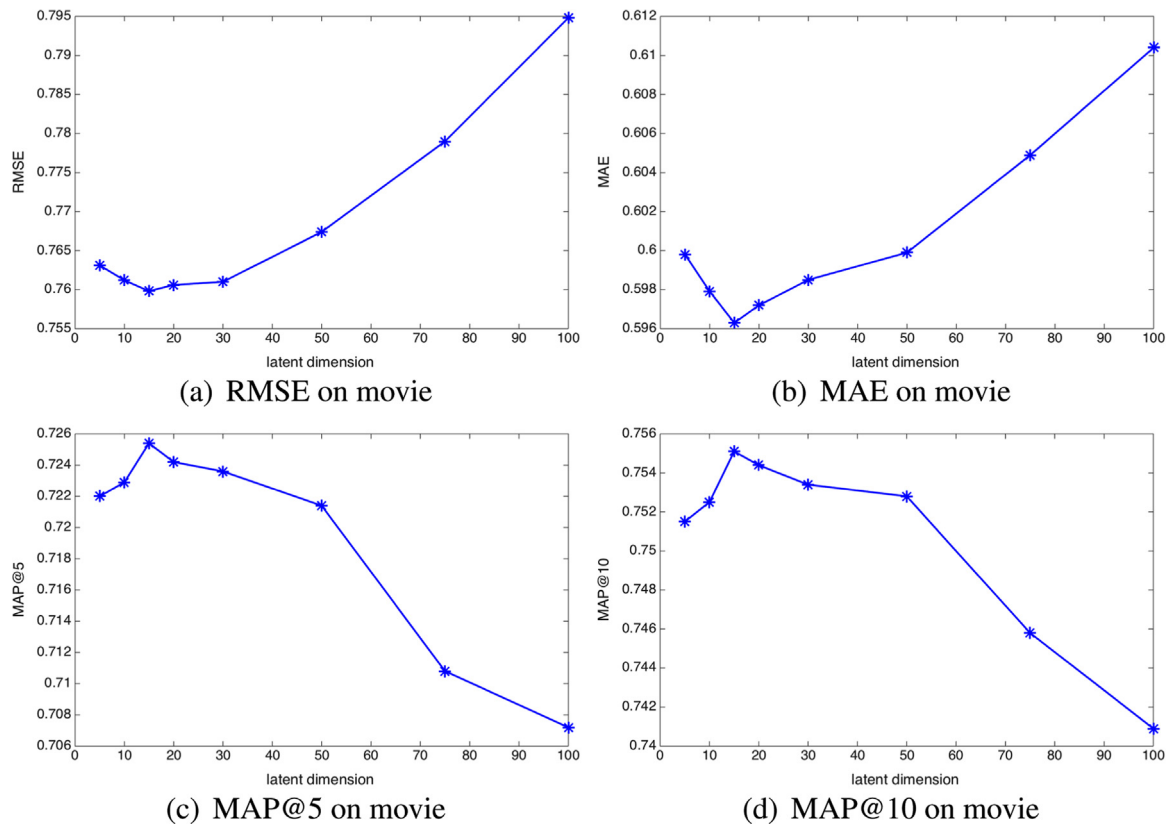
**PMF** [43] is a classical collaborative filtering approach, using fully Bayesian treatment of probabilistic matrix factorization.

**HSR** [44] is a RS framework that incorporates the hierarchical structures of items and user preferences. It only uses the rating information.

**GRMC** [8] improves the basic matrix completion method with graph regularizer, using social relations to regularize the result of



**Fig. 2.** On music, with  $k$  varying in {5, 10, 15, 20, 30, 50, 80, 100}.



**Fig. 3.** On movie, with  $k$  varying in {5, 10, 15, 20, 30, 50, 80, 100}.

matrix completion.

**SR2<sub>pcc</sub>** [7] is a social recommender approach that uses graph regularization term to constrain the trained model, and employs Pearson correlation coefficient (PCC) in the regularization term.

**IrMVL** [28] is a low-rank multi-view matrix completion method. It learns a low-rank common representation for all the views.

We use grid search to determine the value for all the hyperparameters for all the baseline and MVUPL. For MVUPL, we set the learning rate = 0.0001,  $\tau = 1$ , and  $\beta = 1$ . We vary the latent dimension in {5, 10, 15, 20, 30, 50, 80, 100}, and report the corresponding result in (Figs. 2 and 3):

From the results, we can find that MVUPL reaches its best performance when latent dimension  $k=15$ . Thus, we set  $k=15$  for MVUPL in the following experiments. We also set different  $\alpha$  and  $C$  to show MVUPL's sensitivity to these parameters, which will be discussed later.

The results of comparison are given in Tables 1, 2, Figs. 4, and 5 with varying  $x = \{10\%, 30\%, 50\%, 70\%, 90\%\}$ , and from which we have the following observations.

- Recommender approaches that consider social relations significantly improve recommender performance in terms of RMSE, MAE, and MAP. For example, on music dataset with  $x=90$ , GRMC obtains 3.84% improvement comparing with PMF in terms of RMSE.
- Comparing with other multi-view learning approaches, MVUPL does much more better. For instance, compared with IrMVL, MVUPL achieves 13.95% RMSE and 13.34% MAE improvement on music dataset with  $x=90$ .

- Our proposed framework MVUPL always achieves the best result in both RMSE and MAE. For example, compared with PMF, MVUPL gains 0.058 absolute RMSE and 0.0164 absolute MAE improvement on music dataset with  $x=90$ . These results indicate that our proposed model MVUPL can significantly improve the prediction performance.
- Our proposed model MVUPL achieves bigger improvement on dataset with smaller  $x$ , such as  $x = 10, 30$ . For example, comparing with PMF, MVUPL achieves 19.78% relative improvement in terms of RMSE on music dataset with  $x=10$ , which indicates that our approach is able to mitigate the sparsity problem. When  $x=90$ , relatively small edges are gained over the comparing algorithms because rating information is sufficient. As a result, the impact of additional information is relatively small.

#### 4.3. Sensitivity to parameters: $\alpha$ and $C$

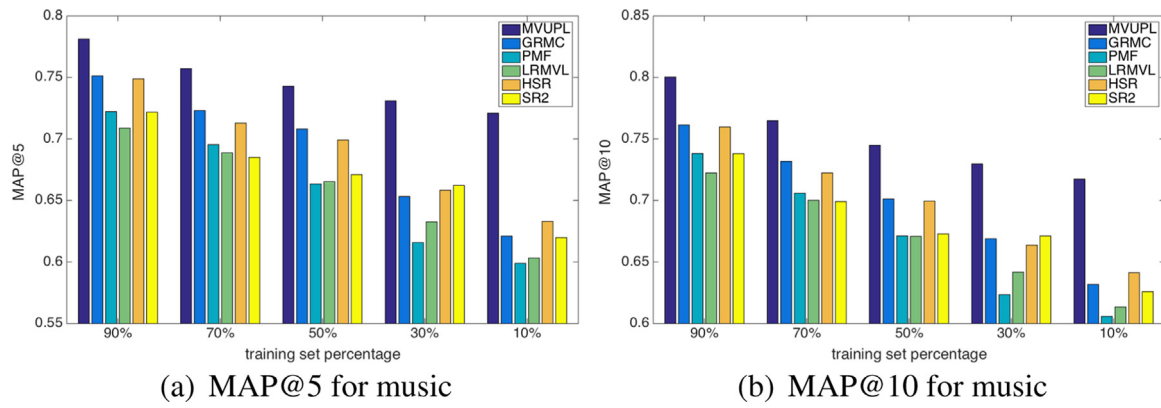
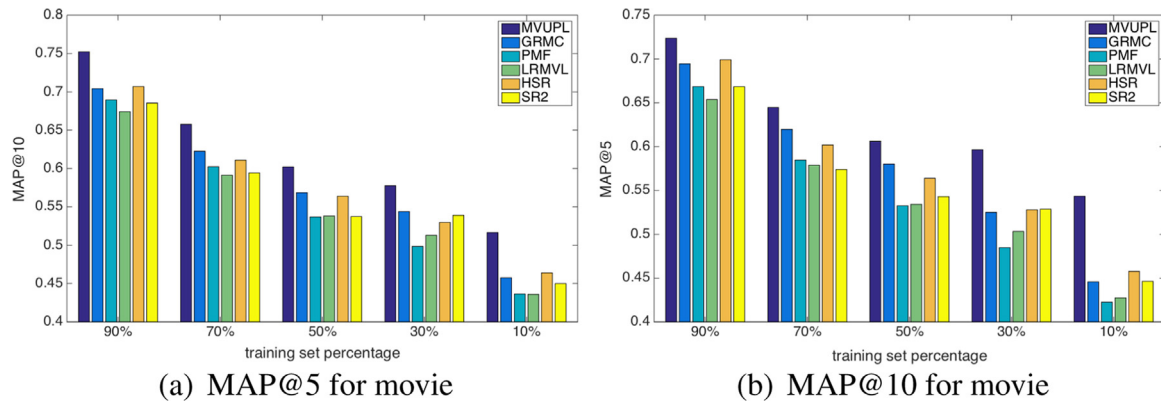
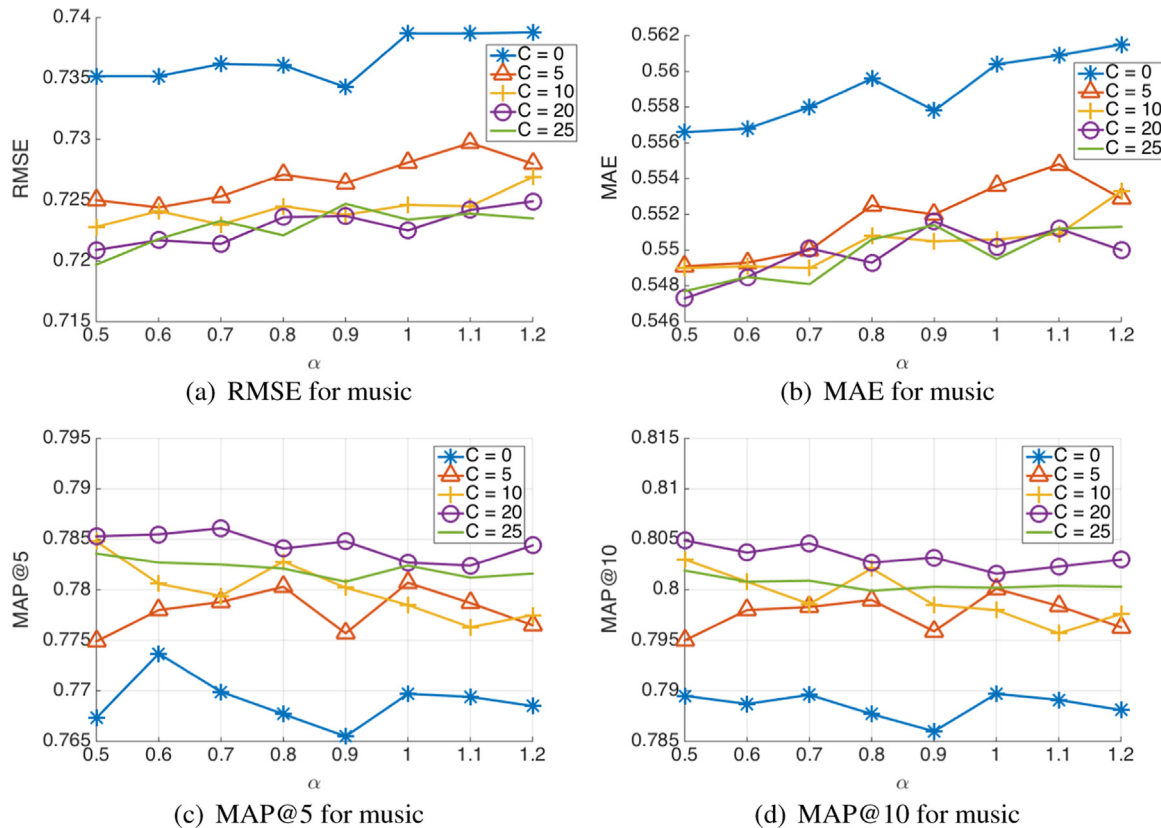
MVUPL has four hyperparameters:  $\alpha$ ,  $C$ ,  $\beta$ , and  $\tau$ .  $\alpha$  controls the contribution from tagging view,  $C$  controls the contribution from social relations,  $\beta$  controls the Frobenius-norm regularizer, and  $\tau$  controls the nuclear norm regularizer. We find out that  $\alpha$  and  $C$  are more important from both analysis and experiments, which means they have more impact on our results. Therefore, we investigate the sensitivity of MVUPL by vary  $\alpha$  and  $C$ . We vary  $\alpha$  in {0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2} and vary  $C$  in {0, 5, 10, 20, 25} on two dataset with 90% as the training set. We calculate the corresponding RMSE, MAE, MAP@5, MAP@10, and then show the result in Figs. 6 and 7. Based on these results, we have following observations:

**Table 1**  
RMSE and MAE on music, with  $\alpha = 0.5$  and  $C=20$ .

| #Train             | RMSE                                  |                                       |                                       |                                       |                                       |
|--------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|
|                    | 10%                                   | 30%                                   | 50%                                   | 70%                                   | 90%                                   |
| PMF                | 1.5242 $\pm$ 0.0073                   | 1.0140 $\pm$ 0.0047                   | 0.8728 $\pm$ 0.0067                   | 0.8155 $\pm$ 0.0047                   | 0.7777 $\pm$ 0.0084                   |
| HSR                | 1.4821 $\pm$ 0.0077                   | 0.9747 $\pm$ 0.0065                   | 0.8417 $\pm$ 0.0079                   | 0.7913 $\pm$ 0.0041                   | 0.7541 $\pm$ 0.0081                   |
| SR2 <sub>pcc</sub> | 1.5465 $\pm$ 0.0110                   | 1.0276 $\pm$ 0.0046                   | 0.8856 $\pm$ 0.0032                   | 0.8282 $\pm$ 0.0031                   | 0.7910 $\pm$ 0.0087                   |
| GRMC               | 1.2773 $\pm$ 0.0044                   | 0.9069 $\pm$ 0.0037                   | 0.8025 $\pm$ 0.0041                   | 0.7680 $\pm$ 0.0037                   | 0.7478 $\pm$ 0.0038                   |
| LRMVL              | 1.6093 $\pm$ 0.0076                   | 1.1953 $\pm$ 0.0032                   | 0.9613 $\pm$ 0.0047                   | 0.8673 $\pm$ 0.0056                   | 0.8364 $\pm$ 0.0065                   |
| MVUPL              | <b>1.2334 <math>\pm</math> 0.0143</b> | <b>0.8809 <math>\pm</math> 0.0063</b> | <b>0.7830 <math>\pm</math> 0.0056</b> | <b>0.7412 <math>\pm</math> 0.0034</b> | <b>0.7197 <math>\pm</math> 0.0065</b> |
| MAE                |                                       |                                       |                                       |                                       |                                       |
| PMF                | 0.9972 $\pm$ 0.0060                   | 0.6809 $\pm$ 0.0024                   | 0.6075 $\pm$ 0.0027                   | 0.5794 $\pm$ 0.0027                   | 0.5637 $\pm$ 0.0028                   |
| HSR                | 0.9042 $\pm$ 0.0051                   | 0.6333 $\pm$ 0.0032                   | 0.5767 $\pm$ 0.0033                   | 0.5584 $\pm$ 0.0022                   | 0.5476 $\pm$ 0.0026                   |
| SR2 <sub>pcc</sub> | 1.0897 $\pm$ 0.0083                   | 0.7167 $\pm$ 0.0035                   | 0.6334 $\pm$ 0.0019                   | 0.6019 $\pm$ 0.0034                   | 0.5820 $\pm$ 0.0049                   |
| GRMC               | 0.8424 $\pm$ 0.0037                   | 0.6578 $\pm$ 0.0021                   | 0.5967 $\pm$ 0.0026                   | 0.5818 $\pm$ 0.0027                   | 0.5704 $\pm$ 0.0026                   |
| LRMVL              | 1.1513 $\pm$ 0.0111                   | 0.8512 $\pm$ 0.0022                   | 0.6913 $\pm$ 0.0030                   | 0.6358 $\pm$ 0.0036                   | 0.6293 $\pm$ 0.0030                   |
| MVUPL              | <b>0.8374 <math>\pm</math> 0.0085</b> | <b>0.6310 <math>\pm</math> 0.0022</b> | <b>0.5771 <math>\pm</math> 0.0031</b> | <b>0.5568 <math>\pm</math> 0.0026</b> | <b>0.5473 <math>\pm</math> 0.0037</b> |

**Table 2**  
RMSE and MAE on movie, with  $\alpha = 0.8$  and  $C=10$ .

| #Train             | RMSE                                  |                                       |                                       |                                       |                                       |
|--------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|
|                    | 10%                                   | 30%                                   | 50%                                   | 70%                                   | 90%                                   |
| PMF                | 1.5523 $\pm$ 0.0099                   | 0.9441 $\pm$ 0.0069                   | 0.8390 $\pm$ 0.0043                   | 0.7937 $\pm$ 0.0049                   | 0.7831 $\pm$ 0.0084                   |
| HSR                | 1.2688 $\pm$ 0.0139                   | 0.8755 $\pm$ 0.0059                   | 0.8068 $\pm$ 0.0049                   | 0.7754 $\pm$ 0.0050                   | 0.7705 $\pm$ 0.0074                   |
| SR2 <sub>pcc</sub> | 1.2881 $\pm$ 0.0110                   | 0.9395 $\pm$ 0.0046                   | 0.8647 $\pm$ 0.0032                   | 0.8295 $\pm$ 0.0031                   | 0.8175 $\pm$ 0.0087                   |
| GRMC               | 1.2942 $\pm$ 0.0104                   | 0.8935 $\pm$ 0.0068                   | 0.8192 $\pm$ 0.0036                   | 0.7874 $\pm$ 0.0043                   | 0.7785 $\pm$ 0.0077                   |
| LRMVL              | 1.3109 $\pm$ 0.0150                   | 0.8824 $\pm$ 0.0064                   | 0.8189 $\pm$ 0.0038                   | 0.8028 $\pm$ 0.0048                   | 0.8063 $\pm$ 0.0074                   |
| MVUPL              | <b>1.2452 <math>\pm</math> 0.0145</b> | <b>0.8451 <math>\pm</math> 0.0061</b> | <b>0.7863 <math>\pm</math> 0.0042</b> | <b>0.7592 <math>\pm</math> 0.0045</b> | <b>0.7546 <math>\pm</math> 0.0071</b> |
| MAE                |                                       |                                       |                                       |                                       |                                       |
| PMF                | 1.1474 $\pm$ 0.0089                   | 0.6908 $\pm$ 0.0040                   | 0.6324 $\pm$ 0.0020                   | 0.6072 $\pm$ 0.0028                   | 0.6050 $\pm$ 0.0053                   |
| HSR                | 0.9277 $\pm$ 0.0081                   | 0.6565 $\pm$ 0.0030                   | 0.6164 $\pm$ 0.0019                   | 0.5955 $\pm$ 0.0027                   | 0.5987 $\pm$ 0.0044                   |
| SR2 <sub>pcc</sub> | 0.9324 $\pm$ 0.0079                   | 0.7057 $\pm$ 0.0027                   | 0.6638 $\pm$ 0.0018                   | 0.6416 $\pm$ 0.0024                   | 0.6343 $\pm$ 0.0049                   |
| GRMC               | 0.9682 $\pm$ 0.0083                   | 0.6837 $\pm$ 0.0042                   | 0.6358 $\pm$ 0.0027                   | 0.6161 $\pm$ 0.0028                   | 0.6153 $\pm$ 0.0051                   |
| LRMVL              | 0.9675 $\pm$ 0.0131                   | 0.6625 $\pm$ 0.0037                   | 0.6288 $\pm$ 0.0023                   | 0.6225 $\pm$ 0.0028                   | 0.6363 $\pm$ 0.0048                   |
| MVUPL              | <b>0.9147 <math>\pm</math> 0.0081</b> | <b>0.6406 <math>\pm</math> 0.0026</b> | <b>0.6076 <math>\pm</math> 0.0017</b> | <b>0.5918 <math>\pm</math> 0.0024</b> | <b>0.5944 <math>\pm</math> 0.0046</b> |

Fig. 4. MAP@5 and MAP@10 on music, with  $\alpha = 0.5$  and  $C=20$ .Fig. 5. MAP@5 and MAP@10 on movie, with  $\alpha = 0.8$  and  $C=10$ .Fig. 6. Sensitivity to  $\alpha$  and  $C$  on movie.



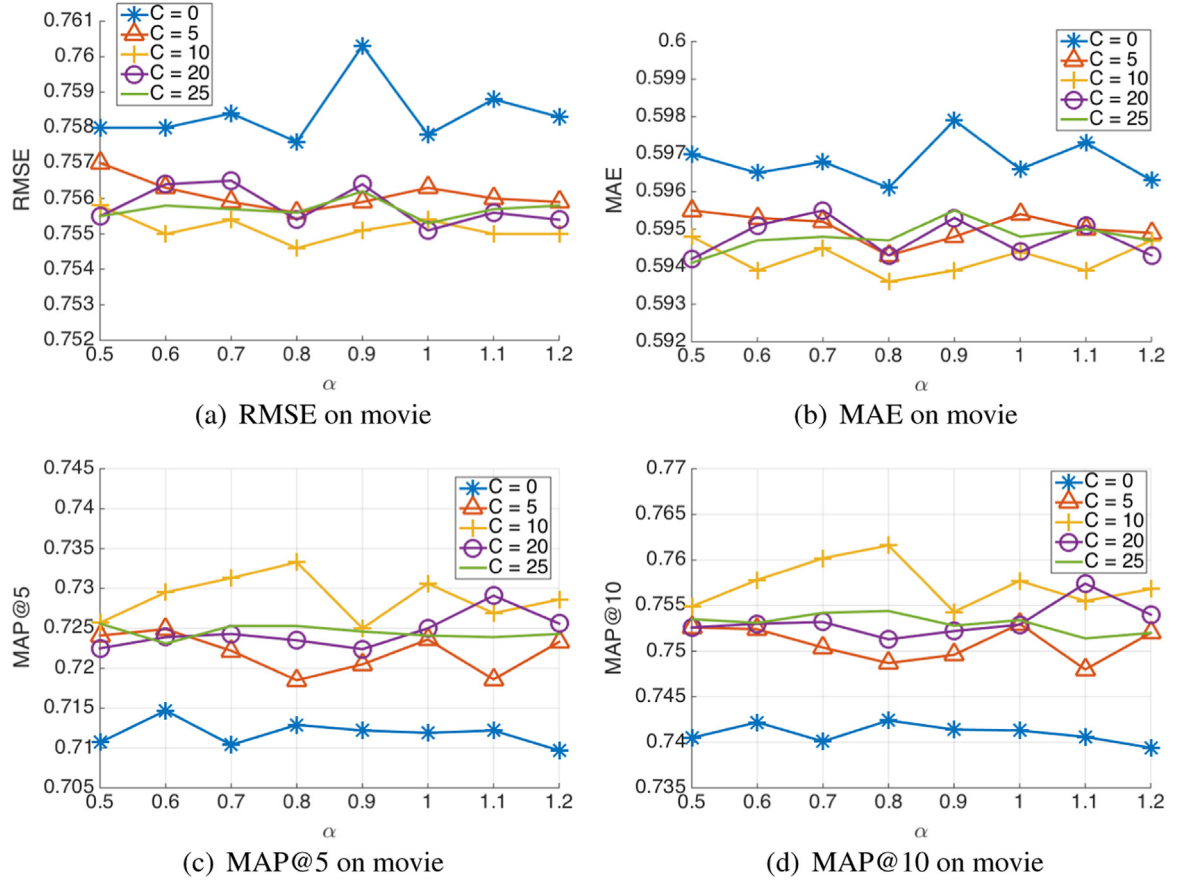
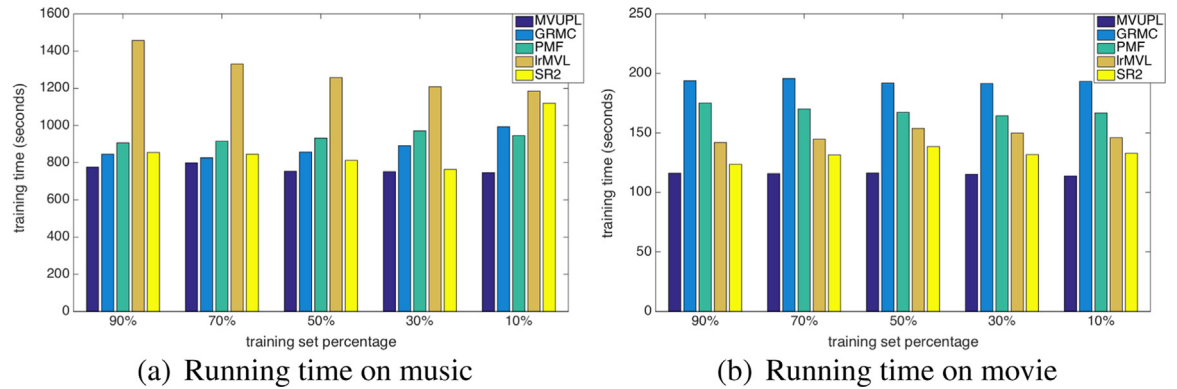
Fig. 7. Sensitivity to  $\alpha$  and  $C$  on music.

Fig. 8. Running time comparison.

- The prediction performance degrades when  $C=0$  on both datasets.
- The prediction performance is relatively stable and not sensitive when  $\alpha$  range in  $[0.5, 1]$  and  $C$  range in  $[10, 20]$ .
- When  $\alpha = 0.5$  and  $C=20$ , MVUPL has the best performance on music dataset, and when  $\alpha = 0.8$  and  $C=10$ , MVUPL has the best performance on movie dataset.

#### 4.4. Running time comparison

In this subsection, we compare the running time performance of the various method used in the experiment. Fig. 8 reports the running time for our method and baseline approaches on music

and movie datasets with  $x = \{10, 30, 50, 70, 90\}$ . The results illustrate that our proposed model MVUPL is the fastest on both datasets with all percentage as the training set. Furthermore, it gives the strong evidence that the side information can speed up the learning process of matrix completion.

#### 5. Conclusion

In this paper, we first introduce a new social recommendation problem by utilizing microblogging data, and then proposed a novel social recommendation framework termed MVUPL to solve this problem. Particularly, we learn user preference by learning a low-

dimensional common representation of multi-view information including rating information, social relations, item side information, and tagging information. Moreover, we develop an efficient alternating direction method of multipliers (ADMM) scheme to solve the introduced model. The intensive experimental results provide strong evidence that MVUPL outperforms other state-of-the-art RS algorithms in terms of both prediction and running time.

## 6. Acknowledgements

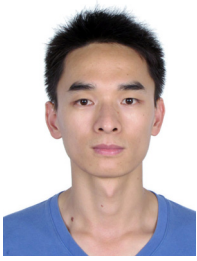
This work was supported by National Basic Research Program of China (973 Program) under Grant 2013CB336500, National Program for Special Support of Top-Notch Young Professionals, National Natural Science Foundation of China under Grant 61233011 and Grant 61125203, Fundamental Research Funds for the Central Universities 2016QNA5015 and the China Knowledge Centre for Engineering Sciences and Technology. The Project is Supported by the Key Laboratory of Advanced Information Science and Network Technology of Beijing (XDXX1603)

## References

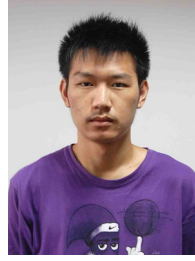
- [1] P. Resnick, H.R. Varian, Recommender systems - introduction to the special section, *Commun. ACM* 40 (3) (1997) 56–58.
- [2] J. Tang, X. Hu, H. Liu, Social recommendation: a review, *Social Netw. Anal. Mining* 3 (4) (2013) 1113–1133.
- [3] T.C. Zhou, H. Ma, M.R. Lyu, I. King, Userrec: A user recommendation framework in social tagging systems, in: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010.
- [4] H. Wang, B. Chen, W. Li, Collaborative topic regression with social regularization for tag recommendation, in: IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3–9, 2013.
- [5] H. Gao, J. Tang, X. Hu, H. Liu, Content-aware point of interest recommendation on location-based social networks, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 1721–1727.
- [6] Y. Zhen, W. Li, D. Yeung, Tagcofi: tag informed collaborative filtering, in: Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys, 2009, pp. 69–76.
- [7] H. Ma, D. Zhou, C. Liu, M.R. Lyu, I. King, Recommender systems with social regularization, in: Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM, 2011, pp. 287–296.
- [8] Z. Zhao, L. Zhang, X. He, W. Ng, Expert finding for question answering via graph regularized matrix completion, *IEEE Trans. Knowl. Data Eng.* 27 (4) (2015) 993–1004.
- [9] C. Chen, X. Zheng, Y. Wang, F. Hong, Z. Lin, Context-aware collaborative topic regression with social matrix factorization for recommender systems, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014, pp. 9–15.
- [10] G. Hu, X. Dai, Y. Song, S. Huang, J. Chen, A synthetic approach for recommendation: Combining ratings, social relations, and reviews, in: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI, 2015, pp. 1756–1762.
- [11] A. Shepitsen, J. Gemmell, B. Mobasher, R.D. Burke, Personalized recommendation in social tagging systems using hierarchical clustering, in: Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys, 2008, pp. 259–266.
- [12] M. Degemmis, P. Lops, G. Semeraro, P. Basile, Integrating tags in a semantic content-based recommender, in: Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys, 2008, pp. 163–170.
- [13] S. Sen, J. Vig, J. Riedl, Tagommenders: connecting users to items through tags, in: Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20–24, 2009, pp. 671–680.
- [14] Y. Luo, J. Tang, J. Yan, C. Xu, Z. Chen, Pre-trained multi-view word embedding using two-side neural network, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014, pp. 1982–1988.
- [15] C. Xu, D. Tao, C. Xu, Large-margin multi-view information bottleneck, *IEEE Trans. Pattern Anal. Mach. Intell.* 36 (8) (2014) 1559–1572.
- [16] A. Zien, C.S. Ong, Multiclass multiple kernel learning, in: Machine Learning, Proceedings of the Twenty-Fourth International Conference ICML, 2007, pp. 1191–1198.
- [17] R. Bharath, C. Xiang, T.H. Lee, Shape classification using invariant features and contextual information in the bag-of-words model, *Pattern Recognit.* 48 (3) (2015) 894–906.
- [18] Y. Lu, F. Xie, T. Liu, Z. Jiang, D. Tao, No reference quality assessment for multiply-distorted images based on an improved bag-of-words model, *IEEE Signal Process. Lett.* 22 (10) (2015) 1811–1815.
- [19] M. Xu, R. Jin, Z. Zhou, Speedup matrix completion with side information: Application to multi-label learning, in: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems, 2013, pp. 2301–2309.
- [20] Z. Zhao, R. Song, X. Xie, X. He, Y. Zhuang, Mobile query recommendation via tensor function learning, in: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI, 2015, pp. 4084–4090.
- [21] Y. Koren, R.M. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, *IEEE Comput.* 42 (8) (2009) 30–37.
- [22] Y. Koren, R.M. Bell, Advances in collaborative filtering, in: Recommender Systems Handbook, 2011, pp. 145–186.
- [23] E.J. Candès, B. Recht, Exact matrix completion via convex optimization, *Found. Comput. Math.* 9 (6) (2009) 717–772.
- [24] Z. Ding, M. Shao, Y. Fu, Latent low-rank transfer subspace learning for missing modality recognition, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014, pp. 1192–1198.
- [25] F. Nie, H. Huang, C.H.Q. Ding, Low-rank matrix recovery via efficient Schatten p-norm minimization, in: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22–26, 2012, Toronto, Ontario, Canada, 2012.
- [26] B. Recht, M. Fazel, P.A. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, *SIAM Rev.* 52 (3) (2010) 471–501.
- [27] Z. Guan, L. Zhang, J. Peng, J. Fan, Multi-view concept learning for data representation, *IEEE Trans. Knowl. Data Eng.* 27 (11) (2015) 3016–3028.
- [28] M. Liu, Y. Luo, D. Tao, C. Xu, Y. Wen, Low-rank multi-view learning in matrix completion for multi-label image classification, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 2778–2784.
- [29] J.S. Breese, D. Heckerman, C.M. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: UAI '98: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998.
- [30] B.M. Sarwar, G. Karypis, J.A. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the Tenth International World Wide Web Conference, WWW, 2001.
- [31] X. Su, T.M. Khoshgoftaar, A survey of collaborative filtering techniques, *Adv. Artif. Intell.*
- [32] W. Pan, A survey of transfer learning for collaborative recommendation with auxiliary data, *Neurocomputing* 177 (2016) 447–453.
- [33] S. Tan, J. Bu, X. Qin, C. Chen, D. Cai, Cross domain recommendation based on multi-type media fusion, *Neurocomputing* 127 (2014) 124–134.
- [34] Z. Guan, C. Wang, J. Bu, C. Chen, K. Yang, D. Cai, X. He, Document recommendation in social tagging services, in: Proceedings of the 19th International Conference on World Wide Web, WWW, 2010, pp. 391–400.
- [35] W. Zhao, Z. Guan, Z. Liu, Ranking on heterogeneous manifolds for tag recommendation in social tagging services, *Neurocomputing* 148 (2015) 521–534.
- [36] H. Zhou, L. Li, Regularized matrix regression, *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* 76 (2) (2014) 463–483.
- [37] L. Luo, Y. Xie, Z. Zhang, W. Li, Support matrix machines, in: Proceedings of the 32nd International Conference on Machine Learning, ICML, 2015, pp. 938–947.
- [38] T. Goldstein, B. O'Donoghue, S. Setzer, R.G. Baraniuk, Fast alternating direction optimization methods, *SIAM J. Imaging Sci.* 7 (3) (2014) 1588–1623.
- [39] J. Cai, E.J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, *SIAM J. Optim.* 20 (4) (2010) 1956–1982.
- [40] G.H. Golub, C.F.V. Loan, *Matrix Computations*, 3 ed., Johns Hopkins University Press, 1996.
- [41] H.-P. Zhang, H.-K. Yu, D.-Y. Xiong, Q. Liu, Hhmm-based Chinese lexical analyzer ictclas, in: Proceedings of the Second SIGHAN Workshop on Chinese Language Processing, vol. 17, SIGHAN '03, 2003, pp. 184–187.
- [42] Y. Wu, C. DuBois, A.X. Zheng, M. Ester, Collaborative denoising auto-encoders for top-n recommender systems, in: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, 2016, pp. 153–162.
- [43] R. Salakhutdinov, A. Mnih, Probabilistic matrix factorization, in: Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, 2007, pp. 1257–1264.
- [44] S. Wang, J. Tang, Y. Wang, H. Liu, Exploring implicit hierarchical structures for recommender systems, in: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI, 2015, pp. 1813–1819.



**Hanqing Lu** is currently a bachelor candidate major in computer science at Zhejiang University, Hangzhou, China. He is working in the Digital Media and Design Lab at Zhejiang University and is expected to get his B. Eng. degree in 2017. His research interests include machine learning, data mining and information retrieval.



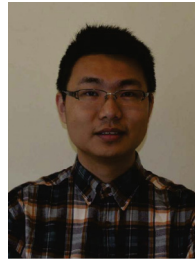
**Chaochao Chen** is currently a Ph.D. candidate at the college of Computer Science at Zhejiang University. Also, he has been a joint Ph.D. student at University of Illinois at Urbana-Champaign since 2014. He received his Bachelor degree from Northeastern University in 2011. His current research is focused on recommender systems and machine learning.



**Hanyi Zhang** is currently a bachelor candidate in College of Computer Science, Zhejiang University, Hangzhou, China. He is working in Digital Media Computing and Design Lab at Zhejiang University and is expected to get B.E. degree in 2017. His research interests mainly include sparse representation and machine learning.



**Ming Kong** is studying for the B.S. degree in computer science from Zhejiang University (ZJU). His research interests include machine learning and data mining.



**Zhou Zhao** received the B.S. and the Ph.D. degrees in computer science from the Hong Kong University of Science and Technology (HKUST), in 2010 and 2015, respectively. He is currently a lecturer with the College of Computer Science, Zhejiang University. His research interests include machine learning and data mining.