A Joint Framework for Collaborative Filtering and Metric Learning

Tak-Lam Wong[1(✉)], Wai Lam[2], Haoran Xie[1], and Fu Lee Wang[3]

[1] The Education University of Hong Kong, Tai Po, Hong Kong
{tlwong,hxie}@eduhk.hk
[2] Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, Hong Kong
wlam@se.cuhk.edu.hk
[3] Caritas Institute of Higher Education, Tseung Kwan O, Hong Kong
pwang@cihe.edu.hk

Abstract. We have developed a framework for jointly conducting collaborative filtering and distance metric learning based on regularized singular value decomposition (RSVD), which discovers the user matrix and item matrix in the low rank space. Our approach is able to solve RSVD and simultaneously learn the parameters of Mahalanobis distance considering the ratings given by similar users and dissimilar users. One characteristic of our approach is that the learned model can be effectively applied to rating prediction and other relevant applications such as trust prediction, resulting in a solution which is coherent and optimal to both tasks. Another characteristic is that social community information and similarity information can be easily considered in our framework. We have conducted extensive experiments on rating prediction using real-world datasets to evaluate our framework. We have also compared our framework with other existing works to illustrate the effectiveness. Experimental results show that our framework achieves a promising prediction performance and outperforms the existing works.

Keywords: Collaborative filtering · Metric learning · Mahalanobis distance

1 Introduction

Collaborative Filtering (CF) have been extensively investigated due to the fact that there is massive volume of information available on the Web and CF it is readily applicable to real-world applications such as recommendation systems. For example, a number of recommendation systems have been developed to predict the movie rating given by users in the Netflix dataset[1], accomplishing very high accuracy. CF discovers the association of user-item ratings and predict the rating to a previously unseen item given by a user. One of the challenges

[1] http://www.netflixprize.com/.

S. Ma et al. (Eds.): AIRS 2016, LNCS 9994, pp. 184–196, 2016.
DOI: 10.1007/978-3-319-48051-0_14

in CF is to handle the sparsity and high dimensionality of the user-item rating matrix. Due to this, the similarity between users are difficult to be computed directly.

Low rank matrix factorization, which identifies the latent factors of the user-item rating matrix, is one of the most common techniques used in CF. By treating the user-item rating matrix as the target matrix, the objective of matrix factorization is to discover the user matrix and item matrix, whose dot-product can approximate the target matrix. Each column of the user matrix and item matrix essentially represent a user and an item respectively. The user and item matrices are normally of lower rank to address the sparsity and the dimensionality problem and improve the efficiency in the prediction of unknown rating. However, one major limitation of low rank matrix factorization is that the similarity between two users will be unavoidably distorted because the column vectors in the user and item matrices corresponding to the smallest eigenvalues will be discarded and only a few significant columns will be retained. For example, let

$$R = \begin{pmatrix} 3 & 4 & 1 \\ 3 & 4 & 2 \\ 2 & 4 & 1 \end{pmatrix}$$

be the user-rating matrix where (i, j)-th entry corresponds to the rating given by user i to item j. The Euclidean distances between users 1 and 2, users 1 and 3, and users 2 and 3 are 1, 1, and 1.4142 respectively. If we apply low rank matrix factorization and set the rank $k = 2$ to solve $R \approx U'\Sigma V$ where $U, V \in \mathbb{R}^{3\times 2}$ and $\Sigma \in \mathbb{R}^{2\times 2}$[2]. The results are

$$U = \begin{pmatrix} -0.5863 & -0.1738 \\ -0.6176 & 0.7279 \\ -0.5242 & -0.6633 \end{pmatrix}, V = \begin{pmatrix} -0.5381 & 0.4055 \\ -0.7982 & -0.5269 \\ -0.2709 & 0.7470 \end{pmatrix}, \text{ and } \Sigma = \begin{pmatrix} 8.6604 & 0 \\ 0 & 0.8284 \end{pmatrix}.$$

If we set $\hat{R} = U'\Sigma V$ and compute the Euclidean distance according to $\hat{R}$, the distances between users 1 and 2, users 1 and 3, and users 2 and 3 are 0.7946, 0.6738, and 1.4081 respectively. As a result, low rank matrix factorization does not consider the distance between users/items in the learned low rank space. As we can observe in the above example, the Euclidean distance between users 1 and 2 is reduced from 1 to 0.7946, while the Euclidean distance between users 1 and 3 is reduced from 1 to 0.6738. The relative changes of the distance from the original space to the new space are different, even though the two distances are the same in the original space. More importantly, such changes completely depend on the user-item rating matrix and do not consider other useful information in a social network. For example, the distance between users 1 and 2 in the new space should be smaller than the distance between users 1 and 3 if users 1 and 2 are "friends" while users 1 and 3 are not in a social network.

[2] In CF, sometimes we directly solve $R \approx U'V$ in which Σ is embedded in U and V.

Regularized Singular Value Decomposition (RSVD) is a common technique used to solve the low rank matrix factorization problem and identify the low-rank user matrix and item matrix. Regularization is originally applied in the model to tackle the problem of model complexity and over-fitting. Several approaches have been proposed to use different regularizers to incorporate additional or prior information in learning the model. For example, Ma proposed to consider the user similarity and item similarity in the regularizer [1]. Essentially, it imposes soft constraints that given a pair of similar users, the two column vectors of the user matrix representing the two users are required to be close to each other. Similarly, the two column vectors of the item matrix representing the two items are required to be close to each other. Empirical results illustrate that prior information in the form of regularizer can substantially improve the performance in prediction. One limitation of this approach is that the closeness of two users/items is represented by the Frobenius norm of the difference between two column vectors. In other words, the distance metric is needed to be designed in advance. More importantly, the distance metric chosen does not take the data collected and the goal of the task into account.

We have developed a framework for jointly conducting collaborative filtering and distance metric learning, aiming at simultaneously discovering the user and item matrices for predicting unknown ratings, and learning the distance metric for other applications, in the new low rank space. Unlike existing works which only address the CF problem, or apply the pre-defined similarity measures to represent the closeness between users/items in the learned model, our approach can automatically discover the similarity metric when computing the user and item matrices when solving RSVD. The major idea of our approach is that given an item, a pair of similar users should give similar rating to this item. Moreover, from the discriminative perspective, the distance between them should be as close as possible in the low rank space. On the contrary, the distance between dissimilar users should be as far as possible in the new space. To achieve this, we have incorporated the parameterized Mahalanobis distance, which essentially is a linear transformation of the distance from the original space to a new space, into the regularizer of RSVD. When solving the RSVD, the user matrix, item matrix, and the parameters of the Mahalanobis distance will be learned jointly in our model. In our designed regularizer, we can easily incorporate the similarity information in the original space in our model. For example, trust information is commonly available in social networks. Trusted users can be considered to similar, while untrusted users can be considered to be dissimilar. With this trust information, the solution will naturally consider both user-item rating information and trust information. As a result, the learned user matrix, item matrix, and the parameters of Mahalanobis distance can be applied to coherently tackle both rating prediction and trust prediction problems, reducing possible conflict between the two tasks. Another characteristic of our approach is that collaborative filtering and distance metric learning serve as regularization to each other, leading to the smoothing effect and reducing overfitting.

The contribution of our work is summarized as follows:

1. We have developed a framework for jointly learning the user and item matrices in low rank space, as well as the distance metric in collaborative filtering. Unlike existing works which depend on the pre-defined distance metrics, out framework can learn the distance metric from the collected data. This is accomplished by incorporating Mahalanobis distance to the regulizers when solving RSVD.
2. Our model can easily incorporate the prior social network information such as trust or community information. This allows our model to consider multiple goals of the tasks and be applied to simultaneously solve different problems.
3. We showed that in our model derived from RSVD, collaborative filtering and distance metric learning serve as regularization to each other. As a result, overfitting can be reduced in both tasks naturally.
4. We have conducted extensive experiments to evaluate our framework and compared it existing works. Empirical results in collaborative filtering demonstrate that our approach significantly outperforms the existing works and achieves promising performance.

2 Related Work

Recommendation systems have been extensively investigated by researchers [2]. Memory-based methods aims at measure the user-user similarity based on the user profile or historical record to predict the rating of items given by a user [3–6]. However, one common shortcoming is the sparsity problem of the raw data. Normally, a user may only rate a relatively small number of the items, out of hundreds or thousands. Given two users, the number of items that are commonly rated is very small. Model-based methods aim at train a model for prediction [7–9]. For example, Zhang and Koren proposed Bayesian hierarchical linear model to tackle the CF problem [10]. In this model, the profile of each user is modeled by a linear model, whose parameters are drawn from a prior distribution. The rating to an item given by a user is then predicted by applying the model with relevant input. Xue et al. proposed a clustering-based method, which first generates clusters of similar users using K-means algorithm [8]. These generated clusters are then exploited to smooth the unknown rating, and hence improve the prediction performance for each individual user. ListCF predicts the ranking of items by a user by measuring the user-user similarity based on the Kullback-Leibler divergence between users' probability distributions over permutations of commonly rated items [11].

Matrix factorization is another commonly used model in CF [12]. The objective of matrix factorization is to discover the user matrix and the item matrix in a low-rank space, such that the dot-product can approximate the original user-item ratings. To address the sparsity problem, regularized singular value decomposition (RSVD) is applied [12,13]. Empirical results have also demonstrated that matrix factorization methods achieved promising performance.

For example, Srebro and Jaakola proposed an approximation method to discover the low rank matrices using EM algorithm and applied in CF [14]. Srebro et al. then proposed another matrix factorization method based on maximum margin principal [15]. This method imposes constraints on the norm of the factorized matrices. Salakhutdinov and Mnih developed different probabilistic matrix factorization models [6,16]. These two models consider the uncertainty involved in the user-item ratings. Instead of predicting the rating, Liu and Yang proposed a method to predict the ranking of items by a user [17].

A number of methods aiming at incorporating additional information in the learned model have been proposed [18,19]. One common method to consider the additional information is to make use of the regularizer in RSVD. For example, Noel et al. proposed to incorporate different forms of regularizer such as feature social regularizer and co-preference regularizer into the objective function when solving RSVD [20]. Ma et al. proposed two regularization models, namely, average-based regularization and individual-based regularization, and applied different similarity measures to consider the social information [21]. Later, Ma developed another method to incorporate the user-user similarity and item-item similarity [1]. Szummer and Yilmaz proposed a method to consider preference regularization to tackle the learning to rank problem in a semi-supervised setting [22].

3 Matrix Factorization

In matrix factorization, there are m users and n items. User i gives item j a rating $r_{ij} = 1, 2, \ldots, r_{max}$, where r_{max} is the maximum value for a rating. Let $R \in \mathbb{R}^{m \times n}$ be the rating matrix where the (i, j)-th entry is equal to r_{ij} if user i has rated item j and 0 otherwise. Note that a user may only rate a few items, hence R is very sparse. Let $\mathcal{E} \equiv \{r_{ij}\}$ for some pairs of i and j be the set of training examples consisting of ratings that user i has rated item j. CF aims at predicting the value of unknown ratings by making use of $\mathcal{E}$. Let $U \in \mathbb{R}^{d \times m}$ and $V \in \mathbb{R}^{d \times n}$, where $d \ll min(m, n)$, be the user matrix and item matrix. We denote $\mathbf{u}_i$ and $\mathbf{v}_j$ be the i-th column vector of U and j-th column vector of V respectively. Matrix factorization treats R as the target matrix and aims at computing U and V such that $R \approx U^\top V$. As a result, the unknown rating to item j given by user i can be predicted by computing $\hat{r}_{ij} = \mathbf{u}_i^\top \mathbf{v}_j$;

Regularized Singular Value Decomposition (RSVD) is a common technique applied to address the sparsity problem in matrix factorization problem. A quadratic loss function is defined as follows:

$$Loss = \frac{1}{2} \sum_{i,j:r_{ij} \in \mathcal{E}} (r_{ij} - \mathbf{u}_i^\top \mathbf{v}_j)^2 + \frac{\lambda_1}{2} \|U\|_F^2 + \frac{\lambda_2}{2} \|V\|_F^2 \quad (1)$$

where $\|\cdot\|_F$ refers to the Frobenius norm. The last two terms are regularizers. The objective of regularization is to avoid large values of U and V, and hence controlling the model complexity and reducing over-fitting. λ_1 and λ_2 are user-defined weighting parameters of the two regularizers. Training of RSVD aims at finding U and V by minimizing the loss function in Eq. 1.

The first derivatives of the loss function can be expressed as follows:

$$\frac{\partial Loss}{\partial \mathbf{u}_i} = \sum_{i,j:r_{ij}\in\mathcal{E}} (r_{ij} - \mathbf{u}_i^\top \mathbf{v}_j)\mathbf{v}_j + \lambda_1 \mathbf{u}_i \tag{2}$$

$$\frac{\partial Loss}{\partial \mathbf{v}_j} = \sum_{i,j:r_{ij}\in\mathcal{E}} (r_{ij} - \mathbf{u}_i^\top \mathbf{v}_j)\mathbf{u}_j + \lambda_2 \mathbf{v}_j \tag{3}$$

Since R is very sparse and not of full rank, setting Eqs. 2 and 3 to zero and solving the system the linear equations is not feasible. Instead, stochastic gradient descent is a common technique for finding the nearly optimal $\mathbf{u}_i$ and $\mathbf{v}_j$. $\mathbf{u}_i$ and $\mathbf{v}_j$ are updated iteratively as follows:

$$\mathbf{u}_i^t \leftarrow \mathbf{u}_i^{(t-1)} + \gamma_1 * [(r_{ij} - \mathbf{u}_i^{(t-1)\top}\mathbf{v}_j^{(t-1)})\mathbf{v}_j^{(t-1)} + \lambda_1 \mathbf{u}_i^{(t-1)}] \tag{4}$$

$$\mathbf{v}_i^t \leftarrow \mathbf{v}_i^{(t-1)} + \gamma_2 * [(r_{ij} - \mathbf{u}_i^{(t-1)\top}\mathbf{v}_j^{(t-1)})\mathbf{u}_j^{(t-1)} + \lambda_2 \mathbf{v}_i^{(t-1)}] \tag{5}$$

where $\mathbf{u}_i^t$ and $\mathbf{v}_i^t$ refer to the $\mathbf{u}_i$ and $\mathbf{v}_j$ at the t-th iteration; γ_1 and γ_2 represent the learning rate of the algorithm. This updating rules are applied for each $r_{ij} \in \mathcal{E}$ until the maximum number of iterations is reached.

4 Our Approach

As mentioned in Sect. 1, one shortcoming of typical RSVD in collaborative filtering is that the distance between two users in the low rank space will be distorted. Moreover, it does not consider prior social network information when computing U and V. Though some existing social recommendation approach attempt to incorporate the similarity between users, the pre-defined distance metric cannot effectively capture the characteristics of the data and directly accomplish the goal of the task. In this section, we first discuss the idea of distance metric learning. Next, we will present our joint model for collaborative filtering and distance metric learning

4.1 Distance Metric Learning

Following the notation used above, $U \in \mathbb{R}^{d\times m}$ denotes to the user matrix where the j-th column refers to the j-th user. Mahalanobis distance, denoted by $d_A(u_i, u_j)$, between users i and j is defined as follows:

$$d_A(\mathbf{u}_i, \mathbf{u}_j) = \|\mathbf{u}_i - \mathbf{u}_j\|_A = \sqrt{(\mathbf{u}_i - \mathbf{u}_j)^\top A(\mathbf{u}_i - \mathbf{u}_j)} \tag{6}$$

where $A \in \mathbb{R}^{d\times d}$ is a semi-definite, $A \succeq 0$. In Mahalanobis distance, A refers to the covariance matrix. If we assume all users are independent, $A = I$ and $d_A(\mathbf{u}_i, \mathbf{u}_j)$ becomes the Euclidean distance between $\mathbf{u}_i$ and $\mathbf{u}_j$. Essentially, A acts as a linear transformation of the distance between $\mathbf{u}_i$ and $\mathbf{u}_j$ from the original space to a new space.

In many applications, we may collect a set of similar or dissimilar objects. For example, in social network, we may treat a pair of users who are friends as similar users. On the contrary, two users who do not know each other are dissimilar. Distance metric learning aims at automatically learning the distance function based on the collected data. In our approach, we consider A in Mahalanobis distance as parameters, which can be learned from the training examples. The objective is the discover A such that the distance between similar users can be linearly transformed to a new space such that they are as close as possible. On the contrary, the distance between dissimilar users should be linearly transformed such that their distance in the new space is as far as possible. We denote $\mathcal{S}$ and $\mathcal{D}$ be the set of pairs of similar users and dissimilar users respectively. We can formulate the distance metric problem as an constrained optimization problem as follows:

$$\begin{aligned} \min_{A} \quad & \sum_{(\mathbf{u}_i,\mathbf{u}_j)\in\mathcal{S}} \|\mathbf{u}_i - \mathbf{u}_j\|_A^2 \\ \text{s.t.} \quad & \sum_{(\mathbf{u}_i,\mathbf{u}_j)\in\mathcal{D}} \|\mathbf{u}_i - \mathbf{u}_j\|_A^2 \geq 1, \\ & A \succeq 0. \end{aligned}$$

The first constraint ensure that the distance between dissimilar users cannot be smaller than 1; the second constraint ensure that the A needs to be semi-positive definite. Note that it is a convex optimization problem. To simplify the learning and improve the efficiency, we set A to a diagonal matrix. As a result, the problem can further be derived to an unconstrained optimization problem as follows:

$$\min_{A} \sum_{(\mathbf{u}_i,\mathbf{u}_j)\in\mathcal{S}} \|\mathbf{u}_i - \mathbf{u}_j\|_A^2 - \log \sum_{(\mathbf{u}_i,\mathbf{u}_j)\in\mathcal{D}} \|\mathbf{u}_i - \mathbf{u}_j\|_A^2 \tag{7}$$

where A is a diagonal matrix. Similarly, regularization is commonly applied to avoid overfitting in learning [23].

4.2 RSVD with Distance Metric Learning

Recall that the objective of our framework is to jointly solve RSVD and distance metric learning. To achieve this, we develop a regularizer based on the aforementioned metric learning problem and integrate to RSVD. The rationale of our approach is to simultaneously solve the RSVD and distance metric learning in a single coherent model. In essence, the loss function of RSVD with distance metric learning is expressed as follows:

$$\begin{aligned} Loss^{new} = & \frac{1}{2} \sum_{i,j:r_{ij}\in\mathcal{E}} (r_{ij} - \mathbf{u}_i^\top \mathbf{v}_j)^2 + \frac{\lambda_1}{2}\|U\|_F^2 + \frac{\lambda_2}{2}\|V\|_F^2 \\ & + \frac{\lambda_3}{2}\{ \sum_{(\mathbf{u}_i,\mathbf{u}_j)\in\mathcal{S}} \|\mathbf{u}_i - \mathbf{u}_j\|_A^2 - \frac{\lambda_4}{2} \log \sum_{(\mathbf{u}_i,\mathbf{u}_j)\in\mathcal{D}} \|\mathbf{u}_i - \mathbf{u}_j\|_A^2 \} \end{aligned} \tag{8}$$

The first three terms and the fourth term on the right hand side refer to the loss function of RSVD and metric learning respectively. To solve RSVD and distance metric learning, we jointly minimize $Loss^{new}$ with respect to U, V, and A.

The first derivatives of the loss function with respect to u_i, v_j and A can be expressed as follows:

$$\frac{\partial Loss^{new}}{\partial \mathbf{u}_i} = \sum_{i,j:r_{ij}\in\mathcal{E}} (r_{ij} - \mathbf{u}_i^\top \mathbf{v}_j)\mathbf{v}_j + \lambda_1 \mathbf{u}_i + \lambda_3 \sum_{(\mathbf{u}_i,\mathbf{u}_j)\in\mathcal{S}} A(\mathbf{u}_i - \mathbf{u}_j) - \frac{\lambda_4}{\sum_{(\mathbf{u}_i,\mathbf{u}_j)\in\mathcal{D}} \|\mathbf{u}_i - \mathbf{u}_j\|_A^2} \sum_{(\mathbf{u}_i,\mathbf{u}_j)\in\mathcal{D}} A(\mathbf{u}_i - \mathbf{u}_j) \tag{9}$$

$$\frac{\partial Loss^{new}}{\partial \mathbf{v}_j} = \sum_{i,j:r_{ij}\in\mathcal{E}} (r_{ij} - \mathbf{u}_i^\top \mathbf{v}_j)\mathbf{u}_j + \lambda_2 \mathbf{v}_j \tag{10}$$

$$\frac{\partial Loss^{new}}{\partial A} = \frac{\lambda_3}{2} \sum_{(\mathbf{u}_i,\mathbf{u}_j)\in\mathcal{S}} (\mathbf{u}_i - \mathbf{u}_j)(\mathbf{u}_i - \mathbf{u}_j)^\top - \frac{\lambda_4}{2\sum_{(\mathbf{u}_i,\mathbf{u}_j)\in\mathcal{D}} \|\mathbf{u}_i - \mathbf{u}_j\|_A^2} \sum_{(\mathbf{u}_i,\mathbf{u}_j)\in\mathcal{D}} (\mathbf{u}_i - \mathbf{u}_j)(\mathbf{u}_i - \mathbf{u}_j)^\top \tag{11}$$

We can then solve the optimization problem by iterative methods like the efficient gradient descent method. One characteristic of our approach is that U, V, and A are jointly varied to optimize $Loss^{new}$. This leads to a solution optimizing both tasks of collaborative filtering and distance metric learning. On the other hand, collaborative filtering and distance metric learning serve regularization to each other resulting to the smoothing effect and reducing over-fitting.

5 Discovery of Similar Users

Recalled that our preference regularizer in Eq. 8 contains the similarity between users. In this paper we employ three different similarity measures to discover similar users.

Jaccard Similarity. Jaccard similarity mainly consider the items that both users have rated, without considering the actual ratings given to these items. Let Q_h and Q_i be the set of items that users h and i have rated respectively. Jaccard similarity is defined as follows:

$$sim(h,i) = \frac{|Q_h \cap Q_i|}{|Q_h \cup Q_i|} \tag{12}$$

Pearson Correlation Coefficient. Pearson correlation coefficient (PCC) aims at measuring the relationship between the ratings given to the items that are rated by two users. Let Q_h and Q_i be the set of items that users h and i have rated respectively. PCC is defined as follows:

$$pcc(h,i) = \frac{\sum_{j\in Q_h\cap Q_i} (r_{hj} - \bar{r_h})(r_{ij} - \bar{r_i})}{\sqrt{\sum_{j\in Q_h\cap Q_i} (r_{hj} - \bar{r_h})^2 \sum_{j\in Q_h\cap Q_i} (r_{ij} - \bar{r_i})^2}} \tag{13}$$

where $\bar{r_h}$ refers to the mean of the ratings to all items given by user h. Since $-1 \leq pcc(h,i) \leq 1$, we define our similarity as follows:

$$sim(h,i) = \frac{1 + pcc(h,i)}{2} \tag{14}$$

Kendall Rank Correlation Coefficient. Unlike PCC, Kendall rank correlation coefficient, denoted as τ, is to measure the relation between the ranking of the items that are rated by two users. Let Q_h and Q_i be the set of items that users h and i have rated respectively. $\tau(h,i)$ is defined as follows:

$$\tau(h,i) = \frac{\sum_{j,k \in Q_h \cap Q_i} sign((r_{hj} - r_{hk})(r_{ij} - r_{ik}))}{\frac{1}{2}|Q_h \cap Q_i|(|Q_h \cap Q_i| - 1)} \tag{15}$$

Since $-1 \leq \tau(h,i) \leq 1$, we define our simiarity as follows:

$$sim(h,i) = \frac{1+\tau(h,i)}{2} \tag{16}$$

The computation of PCC and τ coefficient are computationally expensive. To reduce the computational time, for any pair of users, we randomly sample N items that are rated by them to compute pcc and τ coefficient. In our experiments, N is set to 10. Next, given a user i, the top-K similar users such that the similarity is greater than 0.75 are considered to be similar to user i and constitute $S(i)$ in Eq. 8.

6 Experimental Results

We have conducted experiments on two real-world datasets to evaluate the effectiveness of our framework. The first dataset we used is the MovieLens dataset[3]. This dataset consists of 100,000 ratings (between 1 and 5) from 943 users on 1,642 movies. We call this dataset *ml-100k*. Another dataset is the Epinions dataset[4]. This dataset consists of 664,823 ratings (between 1 and 5) from 49,290 users on 139,738 different items. We call this dataset *epinions*. In each dataset, we randomly divided the data into five portions, namely u1 to u5, with equal number of ratings. In each run of the experiments, we treated four portions as the set of training examples and the remaining portion as the test data. For example, we utilized u1-u4 for training and u5 for testing. As a result, we conducted 5 runs of experiments, each of which utilized different portions as testing data, for each dataset.

Three sets of experiments were conducted to evaluate our framework. In the first set of experiments, we applied the standard RSVD method on the datasets. This can be regarded as our baseline method. We call this *RSVD approach*. In the second set of experiments, we implemented the existing method described in [1] and applied it on the datasets. We implemented the SR_{i+-}^{u+-} approach described in [1]. We call this *Ma's approach*. We compared with this approach because it also aims at improving collaborative filtering via regularization. However, it only considers the closeness between users and the closeness between items in the learned model. In the third sets of experiments, we applied our framework, using different similarity measures as described above. We call this *Our approach*.

[3] The dataset can be freely downloaded in http://www.grouplens.org/.

[4] The dataset can be freely downloaded in http://www.trustlet.org/wiki/Downloaded_Epinions_dataset.

Table 1. The prediction performance of RSVD approach, Ma's approach, and Our approach on the dataset ml-100k.

Testing data	RSVD approach	Ma's approach	Our approach		
			Jaccard	PCC	τ coefficient
u1	0.757	0.728	0.712	0.711	0.709
u2	0.749	0.718	0.711	0.700	0.702
u3	0.749	0.722	0.712	0.701	0.703
u4	0.750	0.725	0.710	0.710	0.709
u5	0.752	0.724	0.713	0.707	0.714
Average	0.751	0.723	0.712	0.706	0.707

Table 2. The prediction performance of RSVD approach, Ma's approach, and Our approach on the dataset epinions.

Testing data	RSVD approach	Ma's approach	Our approach		
			Jaccard	PCC	τ coefficient
u1	0.826	0.804	0.783	0.774	0.780
u2	0.824	0.803	0.780	0.779	0.783
u3	0.825	0.801	0.779	0.791	0.784
u4	0.824	0.802	0.782	0.798	0.783
u5	0.823	0.800	0.787	0.781	0.787
Average	0.824	0.802	0.782	0.785	0.783

In all these approaches, we set the dimension d in matrix factorization to 10. We also followed [1] to set the parameters λ_1, λ_2, γ_1, and γ_2 to 0.01, 0.01, 0.005, and 0.005 respectively. In our approach, we also set λ_3 to 0.01, so that all regularizers have the same weighting. The maximum iteration when running stochastic gradient descent optimization was set to 50,000. Since the ratings of the datasets we used in the experiments are discrete, we round the predicted ratings of the three approaches to the nearest integer.

We adopted the commonly used evaluation metric, namely, Mean-Absolute-Error (MAE), which is defined as follows:

$$MAE = \frac{\sum_{r_{ij} \in \mathcal{T}} |r_{ij} - \hat{r}_{ij}|}{|\mathcal{T}|} \tag{17}$$

where $\mathcal{T}$ refers to the set of testing data.

Table 1 shows the prediction performance on the dataset ml-100k. Each row of the table refers to a run of the experiments. The first column of the table refers to the portion of the dataset used as testing data in this run. The second and third columns contains the prediction performance of RSVD approach and Ma's approach respectively. The fourth column is divided into three sub-columns, each

of which contains the prediction performance of our approach using different similarity measures. The first, second, and third sub-columns refer to the Jaccard similarity, Pearson correlation coefficient (PCC), and Kendall rank correlation coefficient (τ) respectively. The last row of the table shows the average prediction performance. The average MAE of our approach using Jacaard similarity, PCC, and τ coefficient are 0.712, 0.706, and 0.707 respectively. They outperform RSVD approach and Ma's approach, whose average MAE are 0.751 and 0.723 respectively. Among the three different similar measure, our approach achieves similar prediction performance. Table 2 shows the prediction performance of different approaches on the dataset epinions. The format of Table 2 is the same as that of Table 1. Similarly, our approach achieves the best performance, with average MAE of 0.782, 0.785, and 0.783 for Jaccard similarity, PCC, and τ coefficient respectively.

7 Conclusions and Future Work

We have developed a framework for improving rating prediction in collaborative filtering by making use of preference regularization. Our framework is designed based on the idea that similar users should retain the distance in the low-rank space after RSVD. One characteristic of our framework is that collaborative filtering and distance metric learning serve as regularization to each other and naturally reduce overfitting to both. Another characteristic is that social community information and similarity information can be easily considered in our framework. We have conducted several sets of experiments on two real-world datasets to evaluate our framework. We have compared our framework with exiting works. The experimental results show that our framework achieves a very promising performance.

Acknowledgments. The work described in this paper is substantially supported by grants from the Education University of Hong Kong (Project Codes: RG 30/2014-2015R and RG 18/2015-2016R).

References

1. Ma, H.: An experimental study on implicit social recommendation. In: Proceedings of the Thirty-sixth international ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 73–82 (2013)
2. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A.: Recommender systems survey. Knowl. Based Syst. **46**, 109–132 (2013)
3. Deshpande, M., Karypis, G.: Item-based top-n recommendation. ACM Trans. Inf. Syst. **22**(1), 143–177 (2004)
4. Jin, R., Chai, J.Y., Si, L.: An automatic weighting scheme for collaborative filtering. In: Proceedings of the Twenty-Seventh Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 337–344 (2004)

5. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput. **7**(1), 76–80 (2003)
6. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: Proceedings of Advances in Neural Information Processing Systems, pp. 1257–1264 (2007)
7. Hofmann, T.: Latent semantic models for collaborative filtering. ACM Trans. Inf. Syst. **22**(1), 89–115 (2004)
8. Xue, G.R., Lin, C., Yang, Q., Xi, W., Zeng, H.J., Yu, Y., Chen, Z.: Scalable collaborative filtering using cluster-based smoothing. In: Proceedings of the Twenty-Eighth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 114–121 (2005)
9. Si, L., Jin, R.: Flexible mixture model for collaborative filtering. In: Proceedings of the Twentieth International Conference on Machine Learning, pp. 704–711 (2003)
10. Zhang, Y., Koren, J.: Efficient bayesian hierarchical user modeling for recommendation system. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 47–54 (2007)
11. Huang, S., Wang, S., Liu, T.Y., Ma, J., Chen, Z., Veijalainen, J.: Listwise collaborative filtering. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 343–352 (2015)
12. Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Comput. **42**(8), 30–37 (2009)
13. Paterek, A.: Improving regularized singular value decomposition for collaborative filtering. In: Proceedings of the KDD Cup Workshop at SIGKDD 2007, pp. 39–42 (2007)
14. Srebro, N., Jaakkola, T.: Weighted low-rank approximations. In: Proceedings of the 20th International Conference on Machine Learning, pp. 720–727 (2003)
15. Srebro, N., Rennie, J.D.M., Jaakkola, T.: Maximum-margin matrix factorization. In: Proceedings of Advances in Neural Information Processing Systems, pp. 1329–1336 (2004)
16. Salakhutdinov, R., Mnih, A.: Bayesian probabilistic matrix factorization using markov chain monte carlo. In: Proceedings of the Twenty-Fifth International Conference on Machine Learning, pp. 880–887 (2008)
17. Liu, N.N., Yang, Q.: Eigenrank: a ranking-oriented approach to collaborative filtering. In: Proceedings of the Thirty-First Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 83–90 (2008)
18. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434 (2008)
19. Koren, Y.: Collaborative filtering with temporal dynamics. In: Proceedings of the Fifteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 447–456 (2009)
20. Noel, J., Sanner, S., Tran, K.N., Christen, P., Xie, L., Bonilla, E.V., Abbasnejad, E., Nicolas, D.P.: New objective functions for social collaborative filtering. In: Proceedings of the Twenty-First International Conference on World Wide Web, pp. 859–868 (2012)
21. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: Proceedings of the Fourth ACM International Conference on Web search and Data Mining, pp. 287–296 (2011)
22. Szummer, M., Yilmaz, E.: Semi-supervised learning to rank with preference regularization. In: Proceedings of the Twentieth ACM International Conference on Information and Knowledge Management, pp. 269–278 (2011)

23. Jin, R., Wang, S., Zhou, Y.: Regularized distance metric learning: theory and algorithm. In: Advances in Neural Information Processing Systems 22, Neural Information Processing Systems, pp. 862–870 (2009)
24. Yu, K., Schwaighofer, A., Tresp, V., Ma, W.Y., Zhang, H.: Collaborative ensemble learning: combining collaborative and content-based information filtering via hierarchical bayes. In: Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence, pp. 616–623 (2003)
25. Tang, J., Gao, H., Hu, X., Liu, H.: Exploiting homophily effect for trust prediction. In: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, pp. 53–62 (2013)