

# Coupling Multiple Views of Relations for Recommendation

Bin Fu<sup>1,2</sup>, Guandong Xu<sup>1\*</sup>, Longbing Cao<sup>1</sup>, Zhihai Wang<sup>2</sup>, and Zhiang Wu<sup>3</sup>

<sup>1</sup> Advanced Analytics Institute, University of Technology, Sydney, Australia  
Bin.Fu@student.uts.edu.au, {Guandong.Xu, Longbing.Cao}@uts.edu.au

<sup>2</sup> School of Computer and Information Technology, Beijing Jiaotong University,  
Beijing, China  
zhhwang@bjtu.edu.cn

<sup>3</sup> Jiangsu Provincial Key Lab. of E-Business, Nanjing University of Finance and  
Economics, Nanjing, China  
zawuster@gmail.com

**Abstract.** Learning user/item relation is a key issue in recommender system, and existing methods mostly measure the user/item relation from one particular aspect, e.g., historical ratings, etc. However, the relations between users/items could be influenced by multifaceted factors, so any single type of measure could get only a partial view of them. Thus it is more advisable to integrate measures from different aspects to estimate the underlying user/item relation. Furthermore, the estimation of underlying user/item relation should be optimal for current task. To this end, we propose a novel model to couple multiple relations measured on different aspects, and determine the optimal user/item relations via learning the optimal way of integrating these relation measures. Specifically, matrix factorization model is extended in this paper by considering the relations between latent factors of different users/items. Experiments are conducted and our method shows good performance and outperforms other baseline methods.

**Keywords:** Recommender System, Collaborative Filtering, Matrix Factorization.

## 1 Introduction

Recommender system is a type of technology to overcome the information overload problem by estimating users' preferences and finding potentially desirable items for them [1]. For instance, online bookstore *Amazon* would infer users' preferences by analyzing the explicit ratings they have given, and then recommend to them some books that might catch their interests. Currently, recommender system has attracted extensive attention from both of academia and industry, and various methods have been proposed by researchers for recommending items of different types, e.g., movies [2], books [3], and music [4], etc.

---

\* Corresponding author

Among existing methods, collaborative filtering (CF) has been investigated and extended extensively [5]. The underlying assumption of CF is that similar users/items will give/receive similar ratings in future. Therefore, the success of CF essentially rely on how to learn the similarities, i.e., the extent of *relations*, between users/items. Roughly speaking, User/item relation can be measured using different types of auxiliary data, e.g., historical ratings, social networks [6–8], users and items’ attributes [9], user-generated tags [10], and reviews [11], etc.

				<b>director</b>	<i>Robert</i>	<i>Tom</i>	...	<i>Jack</i>
				...	...	...	...	...
				<b>genre</b>	<i>Musical</i>	<i>Crime</i>		<i>Thriller</i>
<b>age</b>	<b>gender</b>	...	<b>job</b>		$v_1$	$v_2$	...	$v_m$
25	F	...	lawyer	$u_1$	1	?	...	1
27	M	...	doctor	$u_2$	1	?	...	4
35	F	...	teacher	$u_3$	?	4	...	?
...	...	...	...	...	1	?	...	?
36	M	...	teacher	$u_n$	?	1	...	?

**Fig. 1.** An example of movie recommendation

Despite the improvements, existing methods mostly exploit only one type of data, e.g., historical ratings, to learn user/item relations. The potential issues include: (1) Each type of data might suffer the insufficiency problem. Figure 1 is a toy dataset for movie recommendation. It shows that: a) the rating matrix is extremely sparse, and b) an user/item is characterized by few attributes. Hence similarities and relations learned based on such kinds of data might be inappropriate. (2) Each type of data only offers a partial view of the underlying user/item relation, whereas multifaceted factors might be involved. For instance, if user  $u_1$  follows other users’ suggestions when rating items, there might be multiple possible factors: they have *similar attributes*, e.g., they are lawyers; they have shown *similar interest*, e.g., rating movies similarly, or they are *friends* in social networks, etc. Therefore, it is more advisable to couple multiple relations measured from different aspects to approximate the underlying user/item relation.

In this paper, we try to address above issues based on the matrix factorization (MF) framework [12]. The key tasks include: (1) Couple multiple views of relations measured from different types of data to approximate the underlying user/item relation. The objective of coupling them together is to capture a better understanding of user/item relation. Similar motivation has been employed in multiple kernel learning, in which different kernels correspond to different notions of similarity [13]. (2) Incorporate user/item relation in the matrix factorization model. To this end, we propose a new recommendation framework based on the MF model, and it can be characterized as follows:

(1) Firstly, multiple types of data are explored and the corresponding relations are coupled in a supervised learning framework. Specifically, for users  $u_i$  and  $u_j$ ,  $k$  different data sources are exploited, thus we get  $k$  relation values. i.e.,  $A^{ij} = (a_1^{ij}, a_2^{ij}, \dots, a_k^{ij})$ , which are then treated as descriptive attributes of

the underlying relation between  $u_i$  and  $u_j$ , i.e.,  $s_{ij}$ . Next, we approximate  $s_{ij}$  as  $s_{ij} = f_{\theta}(A^{ij})$ , thus more appropriate  $s_{ij}$  can be obtained via learning the optimal parameter  $\theta$  of function  $f$  by minimizing a given loss function. Likewise, the same strategy is applied to items.

(2) Secondly, in order to utilize the user/item relation in MF framework, we assume that each user  $u_i$  latent factor vector  $\hat{P}_i$  is determined by two components: his/her basic factor vector  $P_i$ , and other similar users' basic factor vectors  $P_{neighbors}$ . We set  $\hat{P}_i = g(P_i, P_{neighbors}, S)$  to utilize the relations and dependencies between users. Here  $S$  is a matrix that indicates the relation values between all possible pairs of users. The same strategy is applied to items.

We finally compare our model with other state-of-the-art methods on datasets with different types of data to compute user/item relation respectively. To summarize, our main contributions are as follows.

- We propose an innovative model to exploit multiple types of auxiliary data and couple multiple measures of user/item similarity and relation in a supervised learning framework.
- We extend classical MF model to enable that relations between users/items can be exploited when learning each user's/item's latent factor vector.
- We conduct extensive experiments. The results validate the effectiveness of our model, and indicate its applicable scenarios.

The remainder of paper is organized as follows. Section 2 briefly reviews the related work. Problem statement is given in section 3. In section 4, we describe our model in detail. Experimental design and results are presented in section 5, followed by the conclusions in section 6.

## 2 Related work

### 2.1 Compute User/Item Relation using Different Types of Data

According to the types of data source have been utilized to compute user/item similarity, related methods mainly fall into the following categories.

(1) Utilizing historical ratings. This is the most common data have been investigated in traditional CF technique. Two basic methods are user-based approach and item-based approach [2, 5]. User-based approach tries to compute the user relation based on their past ratings, and item-based approach tries to compute item relation based on the ratings they have received. *Cosine* similarity and *Pearson correlation* are the two main measures have been used to compute user/item similarity and relation. (2) Exploiting social networks. Social networks introduce explicit user-user relationship that can be represented as an asymmetric or symmetric matrix [8, 14]. The basic assumption is that users connected in a social network should have similar preferences. Based on this assumption, various models introducing social networks have been proposed [7, 15, 16]. (3) Exploiting user contributed information. As summarized in [17], various types of data generated by users have been explored, such as tags, multimedia content, reviews and comments, etc. User relation is defined by the similarity between corresponding content contributed by the users.

## 2.2 Matrix Factorization (MF) Model

MF model has been extensively investigated and extended recently for recommendation task. It essentially treats the recommendation as a *matrix completion* problem, and solve it by transforming the original rating matrix  $R$  to the multiplication of two matrices  $P^T Q$ , thus  $r_{ij}$ , i.e., the rating assigned to item  $v_j$  by user  $u_i$ , can be approximated as  $r_{ij} \approx P_i^T Q_j$ . Here,  $P_i$  and  $Q_j$  can be treated as a latent factor vector representing user  $u_i$  and item  $v_j$  respectively. The optimal  $P$  and  $Q$  are learned by minimizing a particular loss function.

The relations between users/items' latent factor vectors are ignored in basic MF model. Some of above methods thus try to incorporate the user/item relation into MF model by assuming that similar user/item should have similar latent factor vectors. Specifically, social networks are utilized in [6, 7, 15], and social networks and historical ratings are explored in [16], etc.

Our proposed model is also based on MF model, and similar to these methods. The differences include: (1) Instead of determining user/item relation using a single type of data, our model utilize multiple data sources to measure different user/item relations and couple them together. (2) The weights of different measures and the ultimate user/item relation values are learned in a supervised learning framework, thus they could be optimal for current recommendation task.

## 3 Problem Statement

Based on MF model, we assume that each user/item's latent factor should be dependent on other similar users'/items' latent factors. Thus critical tasks include (1) Estimate the similarities between users/items, here similarity means the extent of user/item relation, we will use this term in following part. (2) Model dependencies between similar users/items in MF model. The targeted problem can be formulated as follows.

Let  $S^U = [S_{ij}^U]_{n \times n}$  denote the similarities between users, where  $s_{ij}^U$  is the similarity between user  $u_i$  and  $u_j$ . As stated before, user/item similarity can be measured from different aspects, e.g., similarity based on ratings, similarity based on connectivity in social networks, etc. We then treat these values as descriptive features of the ultimate user similarity. Formally,  $S_{ij}^U$  should be determined as

$$S_{ij}^U = f_{\theta}(A^{uij}) \quad (1)$$

Here  $A^{uij} = \langle a_1^{uij}, \dots, a_z^{uij} \rangle$  is a set of values consists of different similarity measures for  $u_i$  and  $u_j$ .  $f$  is a function that generates the final similarity and  $\theta$  is a set of parameters that need to be estimated. Since different measures are computed on different data sources, thus multiple types of data are explored.

Likewise, let  $S^V = [S_{ij}^V]_{m \times m}$  denote the similarities between items, where  $s_{ij}^V$  indicates the similarity of item  $v_i$  and  $v_j$ . Same strategy is applied to learn item similarity, we can get

$$S_{ij}^V = f_{\theta}(A^{vij}) \quad (2)$$

Here  $A^{vij} = \langle a_1^{vij}, \dots, a_z^{vij} \rangle$  is a set of different similarity measures for item  $v_i$  and  $v_j$ , and  $\vartheta$  are the parameters that need to be estimated.

To model the dependency relations between users/items, we let  $P$  and  $Q$  denote users' and items' basic latent factors as in traditional MF model. However,  $P$  and  $Q$  should be updated due to the influence between similar users and items. Thus,  $u_i$ 's final factor vector  $\hat{P}_i$  is  $\hat{P}_i = g(P, S^U)$ . Here  $g$  is the function determines the final factor vectors. It is clear that  $\hat{P}_i$  is determined by other users' basic factor vector with respect to their similarities. Same strategy is applied to items, and we get  $\hat{Q}_j = g(Q, S^V)$ .

Finally, rating  $r_{ij}$  is estimated as  $\hat{r}_{ij} = \hat{P}_i^T \hat{Q}_j$ . Now, the specific task is how to design function  $f$ ,  $g$ , and estimate the parameters  $\langle \theta, \vartheta, P, Q \rangle$ , which is elaborated in next section.

## 4 Coupling Multiple Views of Relations

### 4.1 Multiple Views of Similarity

Given a particular type of data, there are various ways to compute similarities between users/items. For example, given the rating matrix  $R_{m \times n}$ , similarity between  $u_i$  and  $u_j$  can be defined as the similarity between corresponding row  $R_i$  and  $R_j$  in  $R$ . *Pearson correlation* can be used to compute the similarity between these rows. There are other ways of measuring similarity given more definitions or more types of auxiliary data. We couple these measures using the framework shown in (1). The results of different measures are then fed into function  $f$  to generate the ultimate similarity value. In this paper, we choose the sigmoid function for  $f$ , since it is a differentiable function which makes the following learning process feasible, and its output range is  $[0, 1]$  that is natural for representing the similarity. The definition of  $f$  for determining user similarity is thus define as:

$$S_{ij}^U = f_\theta(A^{u_{ij}}) = \frac{1}{1 + e^{-\theta^T A^{u_{ij}}}} \quad (3)$$

Similarly, to learn the ultimate item similarity  $S^V$ , we get the following formula.

$$S_{ij}^V = f_\vartheta(A^{v_{ij}}) = \frac{1}{1 + e^{-\vartheta^T A^{v_{ij}}}} \quad (4)$$

The key problem is how to estimate the optimal values of parameters  $\theta, \vartheta$ .

### 4.2 Dependency Propagation among Users/Items

Due to the similarity, an user's preference should be influenced by his neighbours, i.e., a set of similar users. Likewise, an item should also show common characteristics with other similar items. In other words, the latent factor vector of an users/item should be dependent on others' with respect to their similarities.

Let us focus on *user* firstly. In our model, each user  $u_i$  is assumed to have an initial factor vector  $P_i$ . Then,  $u_i$  will get an ultimate factor vector  $\hat{P}_i$  due to

the dependency propagation among similar users. We formulate this dependency propagation as follows:

$$\hat{P}_i = (1 - \alpha_u) \frac{\sum_{l \in \mathbf{Nu}(i)} S_{il}^U P_l}{|\mathbf{Nu}(i)|} + \alpha_u P_i \quad (5)$$

Where  $\mathbf{Nu}(i)$  denotes  $u_i$ 's neighbours, a set of users who are most similar with user  $u_i$  according to  $S^U$ . It is shown that  $\hat{P}_i$  is determined by two components, i.e., his initial factor vector  $P_i$ , and his neighbours' factor vectors. Here  $\alpha_u$  is an adjustable parameter to control the influence of dependency propagation. Since each user depends on his/her neighbours, a dependency graph is actually formed. The dependencies are propagated over the whole graph, thus each user essentially is influenced by all other users directly or indirectly. For items, we can get a similar formulation. Given an item  $v_j$ , its ultimate factor vector  $\hat{Q}_j$  is:

$$\hat{Q}_j = (1 - \alpha_v) \frac{\sum_{l \in \mathbf{Nv}(j)} S_{jl}^V Q_l}{|\mathbf{Nv}(j)|} + \alpha_v Q_j \quad (6)$$

Here  $\mathbf{Nv}(j)$  denotes item  $v_j$ 's neighbours that are most similar with it according to  $S^V$ , and  $\alpha_v$  is the parameter to trade off the two components.

Once the  $\hat{P}_i$  and  $\hat{Q}_j$  are determined, the rating  $r_{ij}$  can be approximated as  $r_{ij} \approx \hat{P}_i^T \hat{Q}_j$ . By now, the whole process of our model is outlined, and a graphical illustration is given in Fig. 2, in which  $P_{i1\dots k}$  are factor vectors of user  $u_i$ 's  $k$  neighbours, and  $Q_{j1\dots k}$  are factor vectors of item  $v_j$ 's  $k$  neighbours.  $A^U = [A^{uij}]_{n \times n}$  is a matrix, and its element  $A^{uij}$  is the precomputed vector of similarity values for user  $u_i$  and  $u_j$ . Similarly,  $A^V = [A^{vij}]_{m \times m}$  is also a matrix, and  $A^{vij}$  is the precomputed vector of similarity values for item  $v_i$  and  $v_j$ . Finally,  $\langle \theta, \vartheta, P, Q \rangle$  are the parameters need to be estimated.

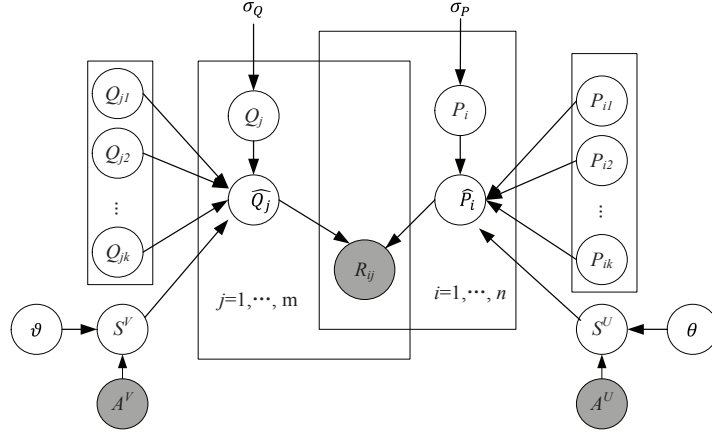
### 4.3 Parameter Estimation

To begin with, we define the loss function which will be used in our model to learn the optimal parameters. Its definition is:

$$L(\theta, \vartheta, P, Q) = \frac{1}{2} \sum_{(i,j) \in \mathbf{D}} (r_{ij} - \hat{P}_i^T \hat{Q}_j)^2 + \lambda (\|P\|_F^2 + \|Q\|_F^2) \quad (7)$$

here  $\mathbf{D}$  is the training set that contains a set of  $(u_i, v_j, r_{ij})$  tuples that record the already known  $r_{ij}$  for pair of  $(u_i, v_j)$ .  $\lambda$  is the parameter to control the influence of the regularization term, and  $\|\cdot\|_F$  denotes the Frobenius norm.  $\hat{P}, \hat{Q}$  are defined in (5) and (6). Now, the task is to learn optimal parameters  $\theta^*, \vartheta^*, P^*, Q^*$  that can minimize above loss function.

We use stochastic gradient descent method to solve this task. The key issue in the process is to compute the partial derivatives of a single training instance's loss with respect to the parameters. Without loss of generality, we show how to



**Fig. 2.** Graphical illustration of our proposed model

compute the partial derivative for a tuple  $(u_i, v_j, r_{ij})$ . Specifically, for  $(u_i, v_j, r_{ij})$ , the loss function becomes:

$$L_{ij}(\theta, \vartheta, P, Q) = \frac{1}{2}(r_{ij} - \hat{P}_i^T \hat{Q}_j)^2 + \lambda(\|P\|_F^2 + \|Q\|_F^2) \quad (8)$$

For simplicity, we set  $e_{ij} = (r_{ij} - \hat{P}_i^T \hat{Q}_j)$ . The partial derivatives of all parameters are computed as follows.

According to (3),  $\theta = \langle \theta_1, \dots, \theta_z \rangle$  is a vector of parameters used to determine user similarities  $S^U$ , so it only exists in  $\hat{P}_i$ . According to (5), for any element  $\theta_t$  in  $\theta$ , the partial derivative is

$$\frac{\partial L_{ij}}{\partial \theta_t} = -e_{ij} \frac{\partial \hat{P}_i^T}{\partial \theta_t} \hat{Q}_j = \frac{-e_{ij}(1 - \alpha_u)}{|\mathbf{Nu}(i)|} \cdot \left( \sum_{l \in \mathbf{Nu}(i)} \frac{\partial S_{il}^U}{\partial \theta_t} P_l \right)^T \hat{Q}_j \quad (9)$$

Here  $|\mathbf{Nu}(i)|$  is the number of  $u_i$ 's neighbours, the remaining problem is  $\partial S_{il}^U / \partial \theta_t$ . According to (3), it can be computed as follows.

$$\frac{\partial S_{il}^U}{\partial \theta_t} = \frac{\partial \frac{1}{1 + e^{-\theta^T A^{uil}}}}{\partial \theta_t} = \frac{e^{-\theta^T A^{uil}}}{(1 + e^{-\theta^T A^{uil}})^2} A_t^{uil} \quad (10)$$

As defined before,  $A^{uil}$  is a vector of similarity measures for user  $u_i$  and  $u_j$ , and  $A_t^{uil}$  is the  $t$ th element of  $A^{uil}$ .

Similarly,  $\vartheta$  is a vector of parameters to determine item similarities  $S^V$ . According to (6), for any element  $\vartheta_t$  in  $\vartheta$ , the partial derivative is:

$$\frac{\partial L_{ij}}{\partial \vartheta_t} = -e_{ij} \hat{P}_i^T \frac{\partial \hat{Q}_j}{\partial \vartheta_t} = \frac{-e_{ij}(1 - \alpha_v)}{|\mathbf{Nv}(j)|} \cdot \hat{P}_i^T \left( \sum_{l \in \mathbf{Nv}(j)} \frac{\partial S_{jl}^V}{\partial \vartheta_t} Q_l \right) \quad (11)$$

Here  $|\mathbf{Nv}(j)|$  is the number of item  $v_j$ 's neighbours. According to (4),  $\partial S_{jl}^V / \partial \vartheta_t$  can be computed as follows.

$$\frac{\partial S_{jl}^V}{\partial \vartheta_t} = \frac{\partial \frac{1}{1+e^{-\vartheta^T A^{vjl}}}}{\partial \vartheta_t} = \frac{e^{-\vartheta^T A^{vjl}}}{(1+e^{-\vartheta^T A^{vjl}})^2} A_t^{vjl} \quad (12)$$

Here  $A^{vjl}$  is a vector of similarity measures for item  $v_j$  and  $v_l$ , and  $A_t^{vjl}$  is the  $t$ th element of  $A^{vjl}$ .

For  $P = [P_1, \dots, P_n]$ , only  $P_i$  and  $P_l (l \in \mathbf{Nu}(i))$  are involved. Therefore, the partial derivative  $\partial L_{ij} / \partial P_i$  is

$$\frac{\partial L_{ij}}{\partial P_i} = -e_{ij} \alpha_u \hat{Q}_j + \lambda P_i \quad (13)$$

For any  $P_l (l \in \mathbf{Nu}(i))$ ,  $\partial L_{ij} / \partial P_l$  is

$$\frac{\partial L_{ij}}{\partial P_l} = \frac{-e_{ij}(1-\alpha_u)S_{il}^U}{|\mathbf{Nu}(i)|} \hat{Q}_j + \lambda P_l \quad (14)$$

Similarly, for  $Q = [Q_1, \dots, Q_m]$ , only  $Q_j$  and  $Q_l (l \in \mathbf{Nv}(j))$  are involved. Therefore,  $\partial L_{ij} / \partial Q_j$  is computed as

$$\frac{\partial L_{ij}}{\partial Q_j} = -e_{ij} \alpha_v \hat{P}_i + \lambda Q_j \quad (15)$$

For any  $Q_l (l \in \mathbf{Nv}(j))$ ,  $\partial L_{ij} / \partial Q_l$  is computed as

$$\frac{\partial L_{ij}}{\partial Q_l} = \frac{-e_{ij}(1-\alpha_v)S_{jl}^V}{|\mathbf{Nv}(j)|} \hat{P}_i + \lambda Q_l \quad (16)$$

We now have described the overall process of our model and how to compute all the necessary components. We call our model as CMR (Couple Multiple Relations). Note that each user/item's neighbours would change in each iteration, due to the dynamical updating of the similarities. Arguably, our model could learn more appropriate user and item similarities by explicitly optimizing the loss function. The validation is shown in next section.

## 5 Experimental Results

### 5.1 Evaluation Metrics and Datasets

The popular metrics, *MAE* and *RMSE*, are used to measure the performance of all the models. Let  $T$  denote a test dataset that contains a set of  $(u_i, v_j, r_{ij})$  tuples, and  $\hat{r}_{ij}$  denote the prediction of  $r_{ij}$ . These metrics are defined as follows.

- Mean Absolute Error (MAE).

$$MAE = \frac{\sum_{(i,j) \in T} |r_{ij} - \hat{r}_{ij}|}{|T|} \quad (17)$$



- Root Mean Squared Error (RMSE).

$$RMSE = \sqrt{\frac{(r_{ij} - \hat{r}_{ij})^2}{|T|}} \quad (18)$$

For both of the metrics, a smaller value means a better performance.

We use 2 datasets, *MovieLens1M* and *Yelp*, to train and evaluate the models. They include different types of data, enabling us to compute user/item similarity from diverse aspects.

- *MovieLens1M* is a dataset regarding recommending movies. Several types of data are included: (1) User attribute. Each user is represented by a vector of attributes, including age, gender, and occupation, etc. (2) Movie attribute. We just keep the category set of each movie as its attributes. (3) Rating set which consists of a set of  $\langle user, item, rating \rangle$  tuples. We remove users and items which have made/received less than 50 ratings, and finally there are 4297 users and 2514 movies left.
- *Yelp* is a dataset regarding recommending Point-of-Interest (POI), such as restaurant, hotel, etc. Several types of data are included: (1) Social networks. For each user, his/her friends are explicitly given. (2) POI attribute. We similarly keep the category set of each POI as its attributes. (3) A set of ratings. The original dataset contain POIs in several cities in United States. We just keep POIs in city "Las Vegas" and users who have rated any of these POIs and the corresponding ratings.

## 5.2 Baselines and Setting

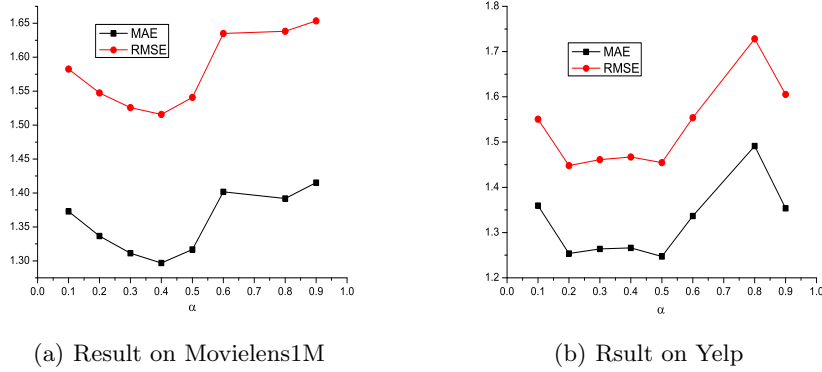
We compare the proposed *CMR* model with other three classical models. These models include:

- *MF* model [12]. This model assumes that users/items are independent from each other, the similarities are ignored.
- *SoRec* model [6]. It takes one type of user relation into consideration. For dataset *MovieLens1m*, user relation is defined as the similarity between users' ratings. For dataset *Yelp*, relations between users are derived from the social networks, which means the similarity is 1 if two users are connected in the social networks, otherwise is 0.
- $SR_{i+}^{u+}$  model [16]. It incorporates one type of user relation and one type of item relation into MF. Item relation is defined as the similarity of items' ratings in both of two datasets. To compute user similarity in the two datasets, we adopt the same approach adopted in *SoRec*.
- our *CMF* model. the aim of our model is to couple multiple measures of user/item relation. Specifically, two types of user relation are considered: a) similarity between users' ratings, and b) similarity between users' attributes (*MovieLens1m*) or connectivities in social networks (*Yelp*). Two types of item relation are also considered: a) similarity between items' ratings, and b) similarity between items' attributes.

Pearson correlation is used to compute the similarity between users' and item's ratings, and Euclidean distance is used to measure the similarity between users' and item's attributes. After trying different values, the length of latent factor vector is set to be 10 in all models. In *SoRec*,  $SR_{i+}^{u+}$ , and *CMF*, the number of neighbours for each user or item is also set to be 10. Since all models use stochastic gradient descent, we also experiment with different values of learning rate. Finally, the learning rate is set to 0.3 in the baseline models and 0.05 in our model. 0.005 is chosen for the regularization term weight  $\lambda$  in all models. For simplicity,  $\alpha_u$  and  $\alpha_v$  in our model are assigned with same value, and we use  $\alpha$  to replace  $\alpha_u, \alpha_v$  in following part.

### 5.3 Results and Analysis

We begin with investigating the impact of  $\alpha$  on our model's performance. Both of the two datasets are split into training set (80%), and test set (20%). We vary  $\alpha$ 's value from 0 to 1, and the results on two datasets are shown in Fig. 3.



**Fig. 3.** The impact of  $\alpha$  on our model

In our model,  $\alpha$  is used to control the influence of similar users/items on current user/item. Seen from Fig. 3, experiments on two datasets show the similar results. That is, optimal performance is obtained when  $\alpha$  is around 0.4 – 0.5, which means the influence of user/relation should be exploited to an appropriate extent. Both of overemphasizing ( $\alpha \rightarrow 0$ ) or ignoring ( $\alpha \rightarrow 1$ ) the influence of user/item relation are inadvisable.

Next, we create another 4 training sets by sampling 20%, 40%, 60%, and 80% of original training set. Models are evaluated on these training sets, and the results are shown in Table 1-2. Each table is divided into two parts. The left part contains results in terms of MAE, and the right part contains results in terms of RMSE. Bold items indicate the corresponding models that performs best.  $\alpha$  is set to be 0.4 in our model.

Tables 1-2 show similar results. They indicate that our model is significantly superior to the other models. Our model performs best on 4 out of the 5 datasets in terms of MAE and RMSE both on MovieLens1M and Yelp. Therefore, the

**Table 1.** Performance on MovieLens1M

Data set	MF	SoRec	$SR_{i+}^{u+}$	CMR	MF	SoRec	$SR_{i+}^{u+}$	CMR
100%	1.3947	1.369	1.3669	<b>1.2969</b>	1.6884	1.6485	1.6722	<b>1.5158</b>
80%	1.4199	1.4252	1.4252	<b>1.3229</b>	<b>1.4199</b>	1.6918	1.6918	1.5424
60%	1.4258	1.4245	1.4245	<b>1.2760</b>	1.6939	1.6919	1.6919	<b>1.4935</b>
40%	1.4024	<b>1.3580</b>	1.3915	1.3863	1.6755	1.6487	1.6624	<b>1.6061</b>
20%	1.3892	1.4167	1.4167	<b>1.3143</b>	1.6886	1.6849	1.6849	<b>1.5327</b>

**Table 2.** Performance on Yelp

Data set	MF	SoRec	$SR_{i+}^{u+}$	CMR	MF	SoRec	$SR_{i+}^{u+}$	CMR
100%	1.4599	1.4216	1.2207	<b>1.2661</b>	1.7542	1.7186	1.4823	<b>1.4671</b>
80%	1.4350	1.4042	1.3120	<b>1.1318</b>	1.7424	1.6970	1.5697	<b>1.3301</b>
60%	1.4873	1.4565	1.4146	<b>1.1852</b>	1.7586	1.7239	1.6752	<b>1.3853</b>
40%	1.4264	1.4167	<b>1.3661</b>	1.4219	1.7100	1.6964	<b>1.6239</b>	1.6254
20%	1.4614	1.4342	1.1677	<b>1.2130</b>	1.7814	1.7199	1.4263	<b>1.4159</b>

results validate our model’s effectiveness due to its capability of coupling multiple views of user/item similarity and learning the optimal user/item similarities for current task.

## 6 Conclusions

This work is motivated by the fact that each single way of computing user/item relation only captures a partial view of the underlying relation. Moreover, no explicit loss function is referred to when learning the relations in existing work. Therefore, we propose a novel framework for coupling multiple views of relations between users/items in this paper. In our proposed framework, multiple relations measured from different types of auxiliary data are coupled to approximate the underlying user/item relation. In conclusions, our model’s advantages include: (1) Multiple similarity measures are coupled together, thus a more accurate and comprehensive approximate of the user/item relation could achieve. (2) Motivated by metric learning etc., we learn the user/item similarity by explicitly optimizing a particular loss function, thus the similarities acquired would be optimal for current recommendation task. (3) Both of the dependencies between users and between items are exploited based on MF model. The experimental results validate the effectiveness and superiority of our model. In following work, we aim to explore more types of auxiliary data to achieve more appropriate approximation of the underlying relations.

**Acknowledgments.** This research has been partially supported by Beijing Natural Science Foundation, China (4142042), and National Centre for International Joint Research on E-Business Information Processing, China (2013B01035).

## References

1. Bobadilla, J., Ortega, F., Hernando, A., Gutierrez, A.: Recommender Systems Survey. *Knowledge-Based Systems* 46, 109–132 (2013)
2. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based Collaborative Filtering Recommendation Algorithms. In: *Proceedings of the 10th International Conference on World Wide Web*, pp. 285–295. (2001)
3. Ziegler, C.-N., McNee, S. M., Konstan, J. A., Lausen, G.: Improving Recommendation Lists through Topic Diversification. In: *Proceedings of the 14th International Conference on World Wide Web*, pp. 22–32. (2005)
4. Kaminskis, M., Ricci, F., Schedl, M.: Location-Aware Music Recommendation Using Auto-tagging and Hybrid Matching. In: *Proceedings of the Seventh ACM Conference on Recommender Systems*, pp. 17–24. ACM (2013)
5. Su, X., Khoshgoftaar, T. M.: A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence* 2009, 4 (2009)
6. Ma, H., King, I., Lyu, M. R.: Learning to Recommend with Explicit and Implicit Social Relations. *ACM Transactions on Intelligent Systems and Technology* 2, 20 (2011)
7. Jamali, M., Ester, M.: A transitivity Aware Matrix Factorization Model for Recommendation in Social Networks. In: *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pp. 2644–2649. (2011)
8. Tang, J., Hu, X., Liu, H.: Social Recommendation: A Review. *Social Network Analysis and Mining* 3, 1113–1133 (2013)
9. Singh, A. P., Gordon, G. J.: Relational Learning via Collective Matrix Factorization. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 650–658. ACM, (2008)
10. Xu, G., Gu, Y., Dolog, P., Zhang, Y., Kitsuregawa, M.: SemRec: A Semantic Enhancement Framework for Tag Based Recommendation. In: *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*. (2011)
11. Tang, J., Gao, H., Hu, X., Liu, H.: Context-Aware Review Helpfulness Rating Prediction. In: *Proceedings of the 7th ACM Conference on Recommender Systems*, pp. 1–8. (2013)
12. Salakhutdinov, R., Mnih, A.: Probabilistic Matrix Factorization. In: *Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems*, pp. 1257–1264. (2008)
13. Gonen, M., Alpaydn, E.: Multiple Kernel Learning Algorithms. *The Journal of Machine Learning Research* 12, 2211–2268 (2011)
14. Konstan, I., Stathopoulos, V., Jose, J. M.: On Social Networks and Collaborative Recommendation. In: *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 195–202 (2009)
15. Ma, H., Yang, H., Lyu, M. R., King, I.: Sorec: Social Recommendation using Probabilistic Matrix Factorization. In: *Proceedings of the 17th ACM Conference on Information and knowledge management*, pp. 931–940. (2008)
16. Ma, H.: An Experimental Study on Implicit Social Recommendation. In: *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 73–82. (2013)
17. Shi, Y., Larson, M., Hanjalic, A.: Collaborative Filtering beyond the User-Item Matrix: a Survey of the State of the Art and Future Challenges. *ACM Computing Surveys* 47, 3 (2014)