2014 Second International Conference on Advanced Cloud and Big Data

Collaborative Filtering for Recommender Systems

Rui-sheng Zhang, Qi-dong Liu, Chun-Gui, Jia-Xuan Wei, Huiyi-Ma
School of Information Science & Engineering
Lanzhou University
Lanzhou, China
wliuqidong@126.com

***Abstract*—Collaborative filtering (CF) predicts user preferences in item selection based on the known user ratings of items. As one of the most common approach to recommender systems, CF has been proved to be effective for solving the information overload problem. CF can be divided into two main branches: memory-based and model-based. Most of the present researches improve the accuracy of Memory-based algorithms only by improving the similarity measures. But few researches focused on the prediction score models which we believe are more important than the similarity measures. The most well-known algorithm to model-based is the matrix factorization. Compared to the memory-based algorithms, matrix factorization algorithm generally has higher accuracy. However, the matrix factorization may fall into local optimum in the learning process which leads to inadequate learning. CF approaches are usually designed to provide products to potential customers. Therefore the accuracy of the methods is crucial. In this paper, we propose various solutions to make a quality recommendation. First, we proposed a new prediction score model for the Memory-based method. Second, we proposed a differential model that considers the adjustment process after the training process in the existing matrix factorization methods. Third, a novel hybrid collaborative filtering is outlined to avoid or compensate for the shortcomings of matrix factorization and neighbor-based methods. In the end, we performed the experiments on MovieLens datasets and the results confirmed the effectiveness of our methods.**

Keywords-collaborative filtering; recommender system; matrix factorization; neighbor-based; hybrid

I. INTRODUCTION

Recommender systems (RS) which use data mining and information filtering techniques to provide products, services and information to potential customers have attracted a lot of attention of researchers. It has been regarded as an important tool to solve the information overload problem. While RS being originally a field dominated by computer scientists, and is now a topic of interest also for mathematicians, physicists, and psychologists [1].

Broadly speaking, the recommender systems can be divided into three main branches according to different strategies – collaborative filtering (CF), contend-based and the combination of both [2]. Among of them, the collaborative filtering is the most widely used since it has the advantage that does not require the creation of explicit profiles [3]. CF can be divided into two main branches: memory-based and model-based.

A. Basic Memory-based Algorithm

Memory-based algorithms, also known as neighbor-based algorithms, operate on entire database of ratings collected by the vendor or service supplier. The Memory-based algorithms are widely used in many large commercial sites, such as Amazon etc. It mainly divided into two branches: user-oriented and item-oriented [21]. In this paper, we will discuss the user-oriented algorithm.

Suppose we want to calculate the prediction score $\hat{r}_{u,i}$ for user u on item i.

First, compute the similarity among users and get the queue $N(u)$ for each user u. Sort users by similarity with user u, from largest to smallest.

Second, select the top-N users from $N(u)$ to form the set $T(u, N)$ for each user u. Then filter the data which $r_{u,i}$ are unknown: $N(u, i) = T(u, N) \cap R(i)$.

Finally, we calculate the prediction score by (1):

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{v \in N(u,i)} s_{u,v} \cdot (r_{v,i} - \bar{r}_v)}{\sum_{v \in N(u,i)} s_{u,v}} \quad (1)$$

Where $\bar{r}_u$ and $\bar{r}_v$ respectively denotes the average item rating for user u and v, $s_{u,v}$ denotes the similarity between users u and v.

B. Basic Matrix Factorization Techniques

The most well-known algorithm to model-based is the matrix factorization (MF). Compared to the Memory-based algorithms, MF generally has higher accuracy. The idea behind MF is simple. For a given dimension n_f, the MF aims to approximate $\mathbf{R}$ as the product of two much smaller matrixes.

$$\mathbf{R} \approx \mathbf{P}\mathbf{Q}^{\mathbf{T}} = \underbrace{\begin{bmatrix} \mathbf{p_1} \\ \mathbf{p_2} \\ \vdots \end{bmatrix}}_{n_u \times n_f} \underbrace{[\mathbf{q_1^T} \quad \mathbf{q_2^T} \quad ...]}_{n_f \times n_m} \quad (2)$$

Where $\mathbf{P}$ is an $n_u \times n_f$ matrix and $\mathbf{Q}$ is an $n_m \times n_f$ matrix. We call $\mathbf{P}$ user factor matrix and $\mathbf{Q}$ item factor matrix. $\mathbf{p_u}$ namely the u-th row of $\mathbf{P}$ is a factor vector for user u and $\mathbf{q_i}$ namely the i-th row of $\mathbf{Q}$ is a factor vector for item i. The algorithm aims to learn $\mathbf{P}$ and $\mathbf{Q}$ and makes the product as close as possible to the matrix $\mathbf{R}$. Therefore, the prediction

978-1-4799-8085-7/14 $31.00 © 2014 IEEE
DOI 10.1109/CBD.2014.47

score $\hat{r}_{u,i}$ for each user u and each item i can be calculated by (3)

$$\hat{r}_{ui} = \mathbf{p_u} \times \mathbf{q_i^T} = \sum_{k=1}^{n_f} p_{u,k} q_{k,i} \tag{3}$$

In order to avoid over-fitting, Takács et al, proposed to add a parameter λ for penalizing the square of the Euclidean norm of weights. The method also known as weight decay [20], is widely used in neural networks.

$$(\mathbf{P}^*, \mathbf{Q}^*) = \min_{(\mathbf{P},\mathbf{Q})} \sum_{(u,i)\in\tau} \left(r_{u,i} - \mathbf{p_u q_i^T}\right)^2 + \lambda(\|\mathbf{p_u}\|^2 + \|\mathbf{q_i}\|^2) \tag{4}$$

Where τ is the training set and λ is the regularization factor. Equation (4) states that learning **P** and **Q** on the training set to minimize the sum of squared errors. Generally speaking, we usually take the alternating gradient descent algorithm to find a local minimum of the sum of squared errors.

Suppose: $e_{u,i} = r_{u,i} - \hat{r}_{u,i}\ for\ each (u,i) \in \tau$

$$e'_{u,i} = \frac{1}{2} e_{u,i}^2 + \frac{\lambda}{2}(\|\mathbf{p_u}\|^2 + \|\mathbf{q_i}\|^2) \tag{5}$$

First compute the gradient of $e'_{u,i}$.

$$\frac{\partial}{\partial p_{u,k}} e'_{u,i} = -e_{u,i} \cdot q_{k,i} + \lambda \cdot p_{u,k}\ ,$$

$$\frac{\partial}{\partial q_{k,i}} e'_{u,i} = -e_{u,i} \cdot p_{u,k} + \lambda \cdot q_{k,i} \tag{6}$$

Then update the weights by moving in the direction opposite to the gradient.

$$p'_{u,k} = p_{u,k} + \eta \cdot \left(e_{u,i} \cdot q_{k,i} - \lambda \cdot p_{u,k}\right),$$

$$q'_{k,i} = q_{k,i} + \eta \cdot \left(e_{u,i} \cdot p_{u,k} - \lambda \cdot q_{k,i}\right) \tag{7}$$

Where η is the learning rate.

Even though the CF has been proved to be effective for solving the information overload problem, it still performs not very well in term of accuracy. The reasons are as follows: First, most of the present researches improve the accuracy of Memory-based algorithms only by improving the similarity measures and few researches focused on the prediction score models which we believe are more important than the similarity measure. Second, the existing matrix factorization methods discard the adjustment process after the training process. In this paper, we propose various solutions to make a quality recommendation. And we will point out that the Neighbor-based algorithms are more accurate than Matrix factorization algorithms, but may not be suitable for some situations. The experiment results on MovieLens datasets show that our methods could increase the accuracy of the recommender system dramatically.

II. Related Work

The first works on the field of collaborative filtering were proposed by Goldberg et al. to filter mails from several mailing lists [4]. Breese et al. divided the collaborative filtering into two main groups – Memory-based and Model-based [5].

A. Memory-based Algorithms

Memory-based algorithms, also known as neighbor-based algorithms, operate on entire database of ratings collected by the vendor or service supplier. The Memory-based algorithms are widely used in many large commercial sites, such as Amazon etc.

At present, there have been many ongoing researches focused on developing highly reliable Memory-based algorithms. Most of the researches improve the accuracy of Memory-based algorithms only by improving the similarity measures. Usually there are two models to measure the similarity of users. They are Pearson Correlation Coefficient (PCC) [6] and Vector similarity (VS) [5]. PCC and VS are very simple, but they both have a shortcoming which only consider the co-rated items. It could lead to a problem that two users may have a high similarity only because they have few co-rated items and coincidently ranked these items similarity. Therefore, Hao Ma et al, proposed to add a correlation significance weighting factor that would devalue similarity weights that were based on a small number of co-rated items [7]. In addition to the above methods, reference [8][9] also proposed similarity measures by using the graph theory. Even more, Heng Luo et al, proposed a collaborative filtering framework based on both of local user similarity and global user similarity [10]. The above research about the similarity measure does improve the accuracy of the Memory-based algorithms. But few researches focused on the prediction score models which we believe are more important than the similarity measure. In this paper, our study will fill the gap.

B. Model-based Algorithms

Model-based algorithms are different with Memory-based algorithms. It first uses the database to estimate or learn a model and then apply this model for prediction. Generally speaking, the Model-based algorithms usually have higher accuracy than the Memory-based algorithms.

Among the model-based algorithms, the most representative is the matrix factorization. Over the past times, a lot of matrix factorization techniques have been proposed, including singular value decomposition [11], probabilistic latent semantic analysis [12], probabilistic matrix factorization [13] and etc. Takeing into consideration the personal difference, reference [14][15] proposed a bias feature idea. However, the algorithms mentioned above only consider the learning process, and ignoring the adjustment process after training. In this paper, we proposed a differential model which can be applied to any matrix factorization.

C. Hybrid Algorithms

Hybrid recommender systems by combining each strategy together can provide better performance rather than either strategy alone [16][17]. The most famous is the BellKor's solution winning the Netflix prize [18], which combines predictions from 107 different baseline recommender systems. By Burke's survey [19], the hybrid recommender systems are mainly divided into the following classes. They are mixed, switched, weighted, feature-argumentation and meta-level hybrids. In this paper, we proposed a weighted hybrid methods which can avoid or compensate for the shortcomings of matrix factorization and neighbor-based methods.

The rest of the paper is organized as follows. The next section provides a description of the differential model proposed by us. Section 4 provides a description of the improved Neighbor-based methods proposed by us. In section 5, we describe the hybrid method in detail. Section 6 provides the experimental results. Finally, conclusions and future works are provided in section 7.

TABLE I. MATHEMATICAL NOTATIONS

Notation	Description
n_u	Number of users
n_m	Number of items
n_f	Number of factors
τ	Training set
T	Test set
$R(u)$	A set of items rated by user u
$R(i)$	A set of users who actually rated item i
$N(u)$	The neighbor queue of user u, sort items by similarity with user u, from largest to smallest.
$T(u,N)$	A set consist of the first N users in $N(u)$
$N(u,i)$	$N(u,i) = T(u,N) \cap R(i)$
$U(u,v)$	$U(u,v) = R(u) \cap R(v)$, a set of items both rated by u and v

III. IMPROVED MATRIX FACTORIZATION ALGORITHMS

In this section, we first introduce BRISMF [15] method proposed by Takács. Then the differential model proposed by us is outlined which can apply to any matrix factorization and make a quality recommendation. Table I lists mathematical notations used in this paper.

A. BRISMF

Considering the personal difference, for example, some users tend to rate all items higher or lower than the average, Paterek et al, proposed the bias feature idea which by extending Matrix Factorization (MF) with biases for users and items [14]. At the same time, Takács et al, proposed a BRISMF [15] which share some common features with Paterek's algorithm, see (8).

$$\hat{r}_{u,i} = \mathbf{p_u} \cdot \mathbf{q_i^T} + b_u + b_i' \tag{8}$$

Where b_u denotes the bias for users and b_i' denotes the bias for items. Equation (8) could also be written as follows:

$$\hat{r}_{u,i} = \sum_{k=1}^{n_f} p_{u,k} \cdot q_{k,i} + b_u \times 1 + 1 \times b_i' \tag{9}$$

Suppose $p_{u,n_f+1} = b_u$, $q_{n_f+1,i} = 1$ and $p_{u,n_f+2} = 1$,$q_{n_f+2,i} = b_i'$ then:

$$\hat{r}_{u,i} = \sum_{k=1}^{n_f} p_{u,k} \cdot q_{k,i} + p_{u,n_f+1} \cdot q_{n_f+1,i} + p_{u,n_f+2} \cdot q_{n_f+2,i} = \sum_{k=1}^{n_f+2} p_{u,k} \cdot q_{k,i} \tag{10}$$

From (10) we can observe that BRISMF is also a product of two much smaller matrixes. The only difference between BRISMF and MF is the $n_f + 2$ column of **P** and the $n_f + 1$ row of **Q** are constant value 1. Therefore, BRISMF incorporate bias features into MF by fixing the $n_f + 2$ column of **P** and the $n_f + 1$ row of **Q** to the constant value of 1 and drop the application of (7) when update $\mathbf{p}_{\bullet,\mathbf{n_f+2}}$ and $\mathbf{q}_{\mathbf{n_f+1},\bullet}$ in the learning process. $\mathbf{p}_{\bullet,\mathbf{n_f+1}}$ denotes the bias for users and $\mathbf{q}_{\mathbf{n_f+2},\bullet}$ denotes the bias for items. Then, the prediction score can be calculated by (10).

B. Differential Model

Matrix factorization algorithm may fall into local optimum in the learning process which leads to inadequate learning. If the problem mentioned above is inevitable or the model has been trained and cannot be modified, we need some remedy after learning. However, most of the matrix factorization techniques at present only considering the learning process, and ignoring the adjustment process after training. In this paper, we proposed a differential model which can be applied to any matrix factorization algorithms. The experiment result on Movielens data set show that the differential model can greatly improve the accuracy of matrix factorization.

The idea behind differential model is simple. After getting the user factor matrix **P** and the item factor matrix **Q**, we calculate e_u and e_i' on training data set. Here, e_u is the average difference between the prediction rating and the reality rating for the u-th row of **P**, and e_i' is the average difference between the prediction rating and the reality rating for the i-th column.

For each $(u,i) \in \tau$:

$$e_{u,i} = r_{u,i} - \hat{r}_{u,i} \tag{11}$$

Then, e_u and e_i can be calculated as follows:

$$e_u = \frac{\sum_{j \in R(u)} e_{u,j}}{card(R(u))}, e_i' = \frac{\sum_{v \in R(i)} e_{v,i}}{card(R(i))} \tag{12}$$

Where $R(u)$ is a set of items rated by user u and $R(i)$ is a set of users who actually rated item i. Then we modified the final result by adding e_u and e_i' .

$$\hat{r}'_{u,i} = \hat{r}_{u,i} + e_u + e'_i \tag{13}$$

This method not only can be applied to MF, but also can work well with BRISMF. MF and BRISMF combine the differential model are denoted by DMF and DBRISMF respectively. We will compare these methods with MF and BRISMF in the following experiment.

IV. Improved Neighbor-Based Algorithm

In this section, we first discuss the similarity measures in Neighbor-based algorithm. Then, we point out the weakness of the prediction score model and present the two plans proposed by us.

A. Similarity Measures

Similarity measures play an important role in neighbor-based algorithm since they are used both for selecting the neighbor members and for weighting, so how to calculate the similarity between two users/items is a key issue of collaborative filtering algorithm [22]. Usually there are two models to measure the similarity of users. They are Pearson Correlation Coefficient (PCC) [6] and Vector similarity (VS) [5].

The PCC method defines the similarity between user u and v as:

$$s_{u,v} = \frac{\sum_{i\in U(u,v)}(r_{u,i}-\bar{r}_u)\cdot(r_{v,i}-\bar{r}_v)}{\sqrt{\sum_{i\in U(u,v)}(r_{u,i}-\bar{r}_u)^2}\cdot\sqrt{\sum_{i\in U(u,v)}(r_{v,i}-\bar{r}_v)^2}} \tag{14}$$

While the VS method defines the similarity as

$$s_{u,v} = \frac{\sum_{i\in U(u,v)} r_{u,i}\cdot r_{v,i}}{\sqrt{\sum_{i\in U(u,v)} r_{u,i}^2}\cdot\sqrt{\sum_{i\in U(u,v)} r_{v,i}^2}} \tag{15}$$

Where $U(u,v) = \{i | r_{u,i} \neq \emptyset \cap r_{v,i} \neq \emptyset\}$ denotes the item set which both rated by user u and v.

PCC and VS are very simple, but they both have a shortcoming which only considering the co-rated items. Since the data set is sparse, it may lead to two bad consequences.

First, there are no co-rated items between user u and v render the similarity measure useless.

Second, there are few co-rated items between user u and v, maybe only one or two, render the result unreliable.

However, the user factor matrix **P** in the matrix factorization learning process is not sparse. Therefore, we can run the PCC and VS methods on user factor matrix **P** instead of the user item matrix **R**. Since the user factor matrix **P** is much smaller than **R**, the methods run on **P** are faster.

VS method runs on user factor matrix **P**, called VS-P for short:

$$s_{u,v} = \frac{\sum_{k=1}^{n_f} p_{u,k}\cdot p_{v,k}}{\sqrt{\sum_{k=1}^{n_f} p_{u,k}^2}\cdot\sqrt{\sum_{k=1}^{n_f} p_{v,k}^2}} \tag{16}$$

PCC method runs on user factor matrix **P**, called PCC-P for short:

$$s_{u,v} = \frac{\sum_{k=1}^{n_f}(p_{u,k}-\bar{p}_u)\cdot(p_{v,k}-\bar{p}_v)}{\sqrt{\sum_{k=1}^{n_f}(p_{u,k}-\bar{p}_u)^2}\cdot\sqrt{\sum_{k=1}^{n_f}(p_{v,k}-\bar{p}_v)^2}} \tag{17}$$

Where

$$\bar{p}_u = \frac{\sum_{k=1}^{n_f} p_{u,k}}{n_f}, \bar{p}_v = \frac{\sum_{k=1}^{n_f} p_{v,k}}{n_f} \tag{18}$$

B. Improved Neighbor-based Algorithm

At present, there have been many ongoing researches focused on developing highly reliable Memory-based algorithms. Most of the researches improve the accuracy of Memory-based algorithms only by improving the similarity measures. But few researches focused on the prediction score models which we believe are more important than the similarity measure.

The prediction score model can be written as follows:

$$\hat{r}_{u,i} = \frac{\sum_{v\in N(u,i)} s_{u,v}\cdot(r_{v,i}+dvi_{u,v})}{\sum_{v\in N(u,i)} s_{u,v}} \tag{19}$$

While the most important step is calculating the deviation $dvi_{u,v}$ between user u and v. At present, (1) is widely used as the prediction model in many papers. It can also write as follows:

$$\hat{r}_{u,i} = \frac{\sum_{v\in N(u,i)} s_{u,v}\cdot(r_{v,i}+\bar{r}_u-\bar{r}_v)}{\sum_{v\in N(u,i)} s_{u,v}} \tag{20}$$

It can be observed from (20), the deviation $dvi_{u,v}$ between user u and v are calculating by $\bar{r}_u - \bar{r}_v$. However, most users only partial rate the items. Since the items are different with each other, some items are good (ie, the average rating is high) and some items are bad (ie, the average rating is low), the method take in (20) is not a good choice. For example, suppose user u's rated items are mostly good and user v's rated items are mostly bad, the deviation $dvi_{u,v}$ calculating by $\bar{r}_u - \bar{r}_v$ are higher than the real value. For that reason, in this paper we proposed two plans.

Plan 1:

$$dvi_{u,v} = \frac{\sum_{j\in U(u,v)}(r_{u,j}-r_{v,j})}{|U(u,v)|} \tag{21}$$

Where $|U(u,v)| = card\big(U(u,v)\big)$. We just consider the co-rated items when calculating the deviation $dvi_{u,v}$ between user u and v.

Plan 2:

Plan 1 only considers the co-rated items render large amount of data useless. If the co-rated items between two users are few, the deviation calculating by Plan 1 is unreliable. Therefore, we proposed Plan 2.

First, we calculate the bias for user u and v.

$$B_u = \frac{\sum_{i \in R(u)}(r_{u,i} - B_i')}{|R(u)|}, B_v = \frac{\sum_{i \in R(v)}(r_{v,i} - B_i')}{|R(v)|} \quad (22)$$

In order to eliminate the effects caused by items difference, we minus B_i' for each items when calculating B_u and B_v. Where

$$B_i' = \frac{\sum_{u \in R(i)}(r_{u,i} - B_u)}{|R(I)|}, \quad (23)$$

denotes the bias for item i. Then we can calculate the deviation $dvi_{u,v}$ by (24)

$$dvi_{u,v} = B_u - B_v \quad (24)$$

We can see that there is a cross-dependence between (22) and (23). That is, to find the values of B_u and B_v one needs to know the values of B_i', and vice-versa. For convenience, we adopt (25) to calculate the approximation value of B_i'.

$$B_i' = \frac{\sum_{u \in R(i)} r_{u,i}}{|R(i)|} \quad (25)$$

The experiment results show that the two plans both improve the accuracy of neighbor-based algorithms.

V. Hybrid Collaborative Filtering Algorithms

Generally speaking, the matrix factorization methods usually yield a better accuracy than Neighbor-based algorithms. But it is not always right. According to our observation, the neighbor-based algorithms perform well when the available neighbor is large and perform poor when the available neighbor is few. Sometimes the available neighbor is zero render the neighbor-based methods useless. For that reason, we proposed a weighted hybrid methods which can avoid or compensate for the shortcomings of matrix factorization and neighbor-based methods. We first explain what the available neighbor is?

In the neighbor-based algorithm we need to select the top-N users from $N(u)$ to form the set $T(u, N)$ for each user u. Here $N = |T(u, N)|$ denotes the number of neighbors which is parameter can set by us.

However, not all of the users in $T(u, N)$ are available. We need to filter the data which $r_{u,i}$ are unknown, i.e., $N(u, i) = T(u, N) \cap R(i)$. Here $r = |N(u, i)|$ is the number of available neighbor.

The hybrid model proposed by us as follows:

$$\hat{r}_{u,i} = \omega_1 \cdot \hat{r}_{u,i}^{NB} + \omega_2 \cdot \hat{r}_{u,i}^{MF},$$

$$\omega_1 = r/N, \omega_2 = (N - r)/N \quad (26)$$

Where $\hat{r}_{u,i}^{MF}$ is the prediction score calculate by DBRISMF, $\hat{r}_{u,i}^{NB}$ is the prediction score calculate by Plan 1. We will compare this model with the approach Neighbor Based Correction of MF proposed by Takács [23] in the next section.

VI. Experiment

In this section, we are about to give a detailed description of the data set and the evaluation metrics. We will verify all the methods mentioned above, and make a presentation of the results and conclusions.

A. Data Set

In this paper, we take the 100k MovieLens data set [24]. The data was collected through the MovieLens web site by the GroupLens Research Project at the University of Minnesota. This data set consists of 100,000 ratings (1-5) from 943 users on 1682 movies, each user has rated at least 20 movies.

B. Evaluation Metrics

In order to verify the accuracy of the algorithms, we have used the Mean Absolute Error (MAE) and Root mean-squared Error (RMSE) metrics as evaluation metrics. The MAE and RMSE were two quantity used to measure how close forecasts or predictions are to the eventual outcomes in statistics.

$$\text{MAE} = \frac{1}{|T|}\sum_{(u,i) \in T} |r_{ui} - \hat{r}_{ui}|$$

$$\text{RMSE} = \sqrt{\frac{1}{|T|}\sum_{(u,i) \in T}(r_{ui} - \hat{r}_{ui})^2}. \quad (27)$$

C. Experimental Results

1) Verify the effectiveness of the differential model

In this section, we will compare MF, BRISMF, DMF and DBRISMF in term of accuracy. Here, the main training parameters were set to $\eta = 0.001$ and $\lambda = 0.07$ .To eliminates the effect of the random choosing and gets a more accurate result, we performed 5-fold cross validation to test the algorithm in the whole experiment.

It can be observed from Fig.1, the MAE and RMSE become smaller as the factor increase and at about $\text{Factor} = 350$ reached extreme. In each factor the methods proposed by us (ie DMF and DBRISMF) yield a better accuracy than MF and BRISMF as the MAE and RMSE is significantly lower. This indicates that the proposed differential model can apply to any matrix factorization techniques and improve the quality of recommendation. From Fig.1, we can also see the DMF algorithm not only performed better than MF but also performed better than BRISMF. Therefore, we can say that the differential model proposed by us can dramatically improve the quality of recommendation.

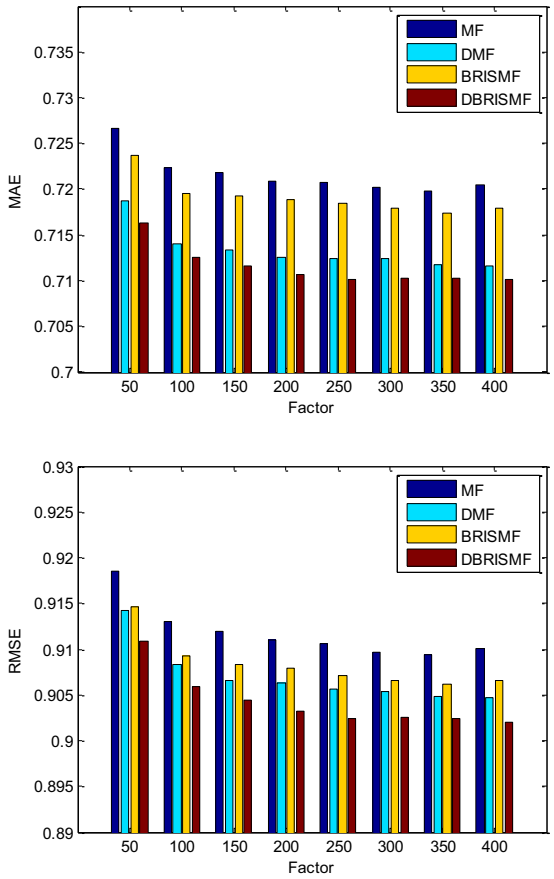

Figure 1. Experiment result on different matrix factorization techniques

2) Comparison of similarity measures

PCC and VS are two excellent algorithms to calculate the similarity between users. But the data set is sparse there are few co-rated items, render the algorithm useless. Considering the user factor matrix **P** is not sparse, VS and PCC run on **P** can easily eliminate the problems. The methods run on user factor matrix **P** denote PCC-P and VS-P respectively, and the methods run on user item matrix R denote PCC-R and VS-R respectively. Here, the user factor matrix **P** is get from BRISMF at $\text{Factor} = 350$. For each algorithm, we implemented (1) to generate the prediction.

From Fig.2 we can see that the PCC method yield a better accuracy than VS. That is because the PCC method takes the personal difference into consideration. We can also see that the methods run on **P** are better than the methods run on **R**, especially when a small number of neighbors. That is because for PCC-R and VS-R, two users may have a high similarity only because they have few co-rated items and coincidently ranked these items similarity. Since there are only few neighbors, those unreliable neighbors may reduce the accuracy of the algorithm. As the neighbors increased, the other neighbors may devalue the effect of the unreliable neighbors. In the next section, we will take PCC-P as the similarity measure as the PCC-P performed best in all of the similarity measures.

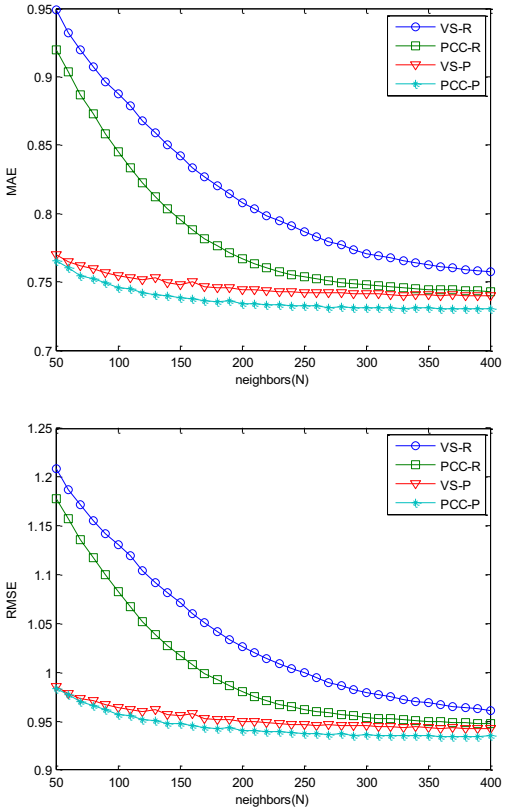

Figure 2. Experiment result on different similarity measures

3) Comparison of neighbor-based algorithms

We have got the right similarity measures, now we will analyze the effects of prediction score model on neighbor-based algorithms. As we discussed in 2.3.3, the most important step for prediction score model is to calculate the deviation $dvi_{u,v}$ between user u and v. The two plans proposed by us are denoted as DVI1 and DVI2 respectively. In this section, we will compare the Plans proposed by us with the NB method which take the (1) as the prediction score model. For each algorithm, we implemented PCC-P to compute the similarity between users and used the algorithm to generate the prediction.

Since the similarity measure run on user factor matrix **P**, the results may have some small floating. From Fig3 we can see that the methods proposed by us yield a better accuracy than NB. We can also see that the DVI2 performed best when there are a few neighbors and DVI1 performed best when there are a large amount of neighbors. That is because the DVI1 only considering the co-rated items render a large amount of data useless. Sometimes the deviation $dvi_{u,v}$ calculates by DVI1 is unreliable when there are a few neighbors. As for the reason that DVI2 performed not better than DVI1 when there are a lot of neighbors, the reason is probably because we take (23) to calculate the approximate

value of B_i' and to some extent deduce the accuracy of the algorithm.

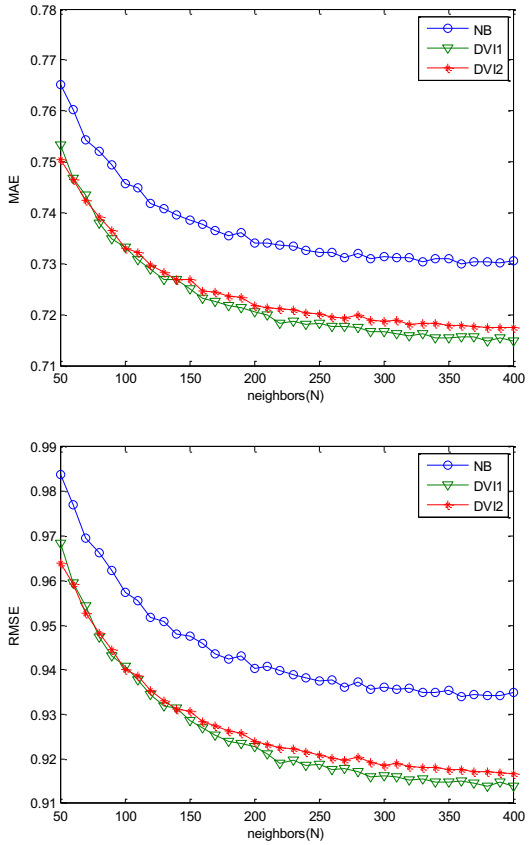

Figure 3. Experiment result on different neighbor-based algorithms

4) Comparison of hybrid methods

Generally speaking, the matrix factorization algorithm yield a better accuracy than neighbor-based algorithm. However, it not always right.

TABLE II. BRISMF VS DVI1

Algorithms	**MAE**	**RMSE**	
BRISMF	0.71737	**0.90620**	Factor=350
DVI1	**0.71490**	0.91385	Neighbors=400

The data in table 1 are the result performed by 5-fold cross validation. 350 and 400 were respectively chosen as the number of Factors and the number of neighbors because BRISMF and DVI1 performed best in this case. From TABLE Ⅱ we can see that BRISMF's RMSE is lower than DVI1, but DVI1 yield a better accuracy in term of MAE. The only difference between MAE and RMSE is the RMSE punish those inaccurate forecasts. The experimental results in TABLE Ⅱ show that DVI1 is more accurate algorithm than BRISMF, but may not be suitable for some situations. The inaccurate forecasts increased the value of RMSE. From Fig.3 we can see that as the neighbors increase, MAE and RMSE become smaller. Therefore, we draw a conclusion that the neighbor-based algorithms perform well when the available neighbor is large and perform poor when the available neighbor is few. For that reason, we proposed a weighted hybrid methods which can avoid or compensate for the shortcomings of matrix factorization and neighbor-based methods. For detailed description, see section 5.

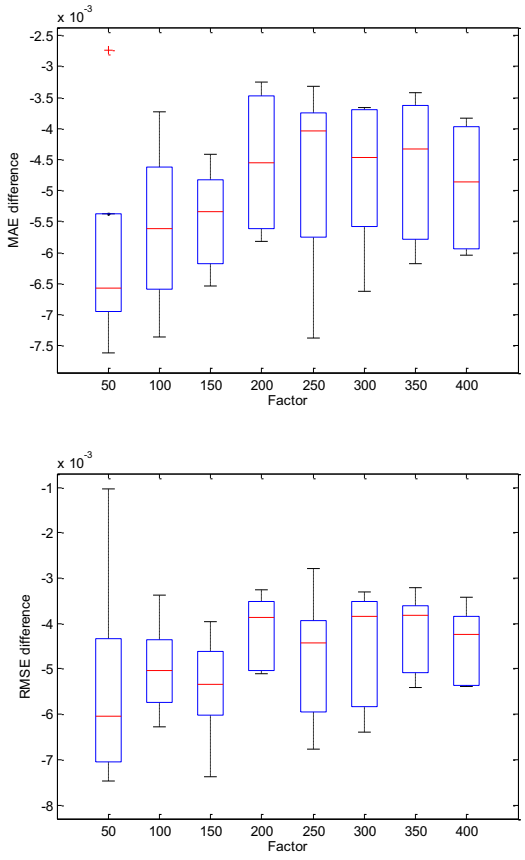

Figure 4. Comparison of hybrid methods

Fig.4 is a box plot on five samples. The data in Fig.4 is the MAE difference and RMSE difference between the hybrid method proposed by us and the NB-MF method. For example, -0.001 means that the MAE or RMSE of the method proposed by us is 0.001 lower than NB-MF. It can be observed from Fig.4, the hybrid method proposed by us yield a better accuracy than NB-MF on different Factor or different samples. MAE and RMSE are both at least 0.003 lower than NB-MF.

The proposed hybrid model is proved to improve the quality of recommendation. In the following we give a table that records the best results of each method.

From table Ⅲ, we can see that the hybrid method proposed by us yield a better accuracy than DBRISMF and DVI1. The experiment result in table Ⅲ confirmed the effectiveness of our methods.

TABLE III. BEST RESULT FOR EACH ALGORITHMS

Algorithms	Evaluation metrics		
	MAE	***RMSE***	
VS-R	0.75707	0.96126	Neighbors=400
PCC-R	0.74296	0.94750	Neighbors=400
VS-P	0.73956	0.94242	Neighbors=400, Factor=350
NB(PCC-P)	0.73063	0.93499	Neighbors=400, Factor=350
DVI1	0.71490	0.91385	Neighbors=400, Factor=350
DVI2	0.71742	0.91677	Neighbors=400, Factor=350
MF	0.71985	0.90939	Factor=350
DMF	0.71180	0.90478	Factor=350
BRISMF	0.71737	0.90620	Factor=350
DBRISMF	0.71023	0.90243	Factor=350
NB-MF	0.71292	0.90482	Factor=350
HYBRID	**0.70827**	**0.90059**	Neighbors=400, Factor=350

VII. CONCLUSION

In this paper, we propose various solutions to make a quality recommendation. The methods we mentioned in this paper are related to many collaborative filtering techniques include the matrix factorization techniques and the neighbors-based methods. The experiment result on MovieLens data set confirmed the effectiveness of our methods. In the future work, we will apply these methods to some larger data sets to verify the feasibility.

ACKNOWLEDGMENT

We are indebted to the Chunhui Plan Cooperation and Research Project supported by Ministry of Education (No.Z2012114).

REFERENCES

[1] Lü L, Medo M, Yeung C H, et al. Recommender systems[J]. Physics Reports, 2012, 519(1): 1-49.USA: Abbrev. of Publisher, year, ch. x, sec. x, pp. xxx–xxx.

[2] Kantor P B, Rokach L, Ricci F, et al. Recommender systems handbook[M]. Springer, 2011.

[3] Koren, Yehuda. "Factor in the neighbors: Scalable and accurate collaborative filtering." ACM Transactions on Knowledge Discovery from Data (TKDD) 4.1 (2010): 1.

[4] Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35(12):61–70, 1992

[5] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In Proc. of UAI-98, 14th Conf. on Uncertainty in Artificial Intelligence, pages 43–52, Madison, Wisconsin, USA, 1998.

[6] Resnick P, Iacovou N, Suchak M, et al. GroupLens: an open architecture for collaborative filtering of netnews[C]//Proceedings of the 1994 ACM conference on Computer supported cooperative work. ACM, 1994: 175-186.

[7] Ma H, King I, Lyu M R. Effective missing data prediction for collaborative filtering[C]//Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2007: 39-46.

[8] Gori M, Pucci A. ItemRank: A Random-Walk Based Scoring Algorithm for Recommender Engines[C]//IJCAI. 2007, 7: 2766-2771.

[9] Fouss F, Pirotte A, Renders J M, et al. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation[J]. Knowledge and Data Engineering, IEEE Transactions on, 2007, 19(3): 355-369.

[10] Luo H, Niu C, Shen R, et al. A collaborative filtering framework based on both local user similarity and global user similarity[J]. Machine Learning, 2008, 72(3): 231-245.

[11] M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Application of dimensionality reduction in recommender system—a case study. In Proc. of WebKDD-00, Web Mining for E-Commerce Workshop, at 6th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, Boston, Massachusetts, USA, 2000.

[12] Hofmann. Latent semantic models for collaborative filtering. ACM Trans. Inf. Syst., 22(1):89–115, 2004.

[13] Salakhutdinov and A.Mnih. Probabilistic matrix factorization. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, Advances in Neural Information Processing Systems 20.MIT Press, Cambridge, Massachusetts, USA, 2008.

[14] Paterek, Arkadiusz. "Improving regularized singular value decomposition for collaborative filtering." Proceedings of KDD cup and workshop. Vol. 2007. 2007.

[15] Takacs, Gabor, et al. "On the gravity recommendation system." Proceedings of KDD cup and workshop. Vol. 2007. 2007.

[16] Yuan-hong, Wu, and Tan Xiao-qiu. "A real-time recommender system based on hybrid collaborative filtering." Computer Science and Education (ICCSE), 2010 5th International Conference on. IEEE, 2010.

[17] Xu, Hai-Ling, et al. "Comparison study of Internet recommendation system." Journal of software 20.2 (2009): 350-362.

[18] Bell, Robert M., Yehuda Koren, and Chris Volinsky. "The BellKor solution to the Netflix prize." KorBell Team's Report to Netflix (2007).

[19] Burke, Robin. "Hybrid recommender systems: Survey and experiments." User modeling and user-adapted interaction 12.4 (2002): 331-370.

[20] R. Duda, P. E. Hart., and D. G. Stork. Pattern Classification. John Wiley and Sons.

[21] Treerattanapitak, Kiatichai, and Chuleerat Jaruskulchai. "Exponential fuzzy C-means for collaborative filtering." Journal of Computer Science and Technology 27.3 (2012): 567-576.

[22] Jin R, Chai J Y, Si L. An automatic weighting scheme for collaborative filtering[C]//Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2004: 337-344.

[23] Takács, Gábor, et al. "Scalable collaborative filtering approaches for large recommender systems." The Journal of Machine Learning Research 10 (2009): 623-656.

[24] http://grouplens.org/datasets/movielens/