Effective Missing Data Prediction for Collaborative Filtering

Hao Ma, Irwin King and Michael R. Lyu
Dept. of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
{ hma, king, lyu }@cse.cuhk.edu.hk

ABSTRACT

Memory-based collaborative filtering algorithms have been widely adopted in many popular recommender systems, although these approaches all suffer from data sparsity and poor prediction quality problems. Usually, the user-item matrix is quite sparse, which directly leads to inaccurate recommendations. This paper focuses the memory-based collaborative filtering problems on two crucial factors: (1) similarity computation between users or items and (2) missing data prediction algorithms. First, we use the enhanced Pearson Correlation Coefficient (PCC) algorithm by adding one parameter which overcomes the potential decrease of accuracy when computing the similarity of users or items. Second, we propose an effective missing data prediction algorithm, in which information of both users and items is taken into account. In this algorithm, we set the similarity threshold for users and items respectively, and the prediction algorithm will determine whether predicting the missing data or not. We also address how to predict the missing data by employing a combination of user and item information. Finally, empirical studies on dataset MovieLens have shown that our newly proposed method outperforms other state-of-the-art collaborative filtering algorithms and it is more robust against data sparsity.

Categories and Subject Descriptors: H.3.3 [Information Systems]: Information Search and Retrieval - Information Filtering

General Terms: Algorithm, Performance, Experimentation.

Keywords: Collaborative Filtering, Recommender System, Data Prediction, Data Sparsity.

1. INTRODUCTION

Collaborative filtering is the method which automatically predicts the interest of an active user by collecting rating information from other similar users or items, and related techniques have been widely employed in some large, famous commercial systems, such as Amazon[1], Ebay[2]. The underlying assumption of collaborative filtering is that the active user will prefer those items which the similar users prefer. The research of collaborative filtering started from memory-based approaches which utilize the entire user-item database to generate a prediction based on user or item similarity. Two types of memory-based methods have been studied: user-based [2, 7, 10, 22] and item-based [5, 12, 17]. User-based methods first look for some similar users who have similar rating styles with the active user and then employ the ratings from those similar users to predict the ratings for the active user. Item-based methods share the same idea with user-based methods. The only difference is user-based methods try to find the similar users for an active user but item-based methods try to find the similar items for each item. Whether in user-based approaches or in item-based approaches, the computation of similarity between users or items is a very critical step. Notable similarity computation algorithms include Pearson Correlation Coefficient (PCC) [16] and Vector Space Similarity (VSS) algorithm [2].

Although memory-based approaches have been widely used in recommendation systems [12, 16], the problem of inaccurate recommendation results still exists in both user-based and item-based approaches. The fundamental problem of memory-based approaches is the data sparsity of the user-item matrix. Many recent algorithms have been proposed to alleviate the data sparsity problem. In [21], Wang et al. proposed a generative probabilistic framework to exploit more of the data available in the user-item matrix by fusing all ratings with a predictive value for a recommendation to be made. Xue et al. [22] proposed a framework for collaborative filtering which combines the strengths of memory-based approaches and model-based approaches by introducing a smoothing-based method, and solved the data sparsity problem by predicting all the missing data in a user-item matrix. Although the simulation showed that this approach can achieve better performance than other collaborative filtering algorithms, the cluster-based smoothing algorithm limited the diversity of users in each cluster and predicting all the missing data in the user-item matrix could bring negative influence for the recommendation of active users.

In this paper, we first use PCC-based significance weighting to compute similarity between users and items, which overcomes the potential decrease of similarity accuracy. Second, we propose an effective missing data prediction algo-

[1]http://www.amazon.com/.
[2]http://www.half.ebay.com/.

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.

rithm which exploits the information both from users and items. Moreover, this algorithm will predict the missing data of a user-item matrix if and only if we think it will bring positive influence for the recommendation of active users instead of predicting every missing data of the user-item matrix. The simulation shows our novel approach achieves better performance than other state-of-the-art collaborative filtering approaches.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of several major approaches for collaborative filtering. Section 3 shows the method of similarity computation. The framework of our missing data prediction and collaborative filtering is introduced in Section 4. The results of an empirical analysis are presented in Section 5, followed by a conclusion in Section 6.

2. RELATED WORK

In this section, we review several major approaches for collaborative filtering. Two types of collaborative filtering approaches are widely studied: *memory-based* and *model-based*.

2.1 Memory-based approaches

The memory-based approaches are the most popular prediction methods and are widely adopted in commercial collaborative filtering systems [12, 16]. The most analyzed examples of memory-based collaborative filtering include user-based approaches [2, 7, 10, 22] and item-based approaches [5, 12, 17]. User-based approaches predict the ratings of active users based on the ratings of similar users found, and item-based approaches predict the ratings of active users based on the information of similar items computed. User-based and item-based approaches often use PCC algorithm [16] and VSS algorithm [2] as the similarity computation methods. PCC-based collaborative filtering generally can achieve higher performance than the other popular algorithm VSS, since it considers the differences of user rating styles.

2.2 Model-based Approaches

In the model-based approaches, training datasets are used to train a predefined model. Examples of model-based approaches include clustering models [11, 20, 22], aspect models [8, 9, 19] and latent factor model [3]. [11] presented an algorithm for collaborative filtering based on hierarchical clustering, which tried to balance robustness and accuracy of predictions, especially when little data were available. Authors in [8] proposed an algorithm based on a generalization of probabilistic latent semantic analysis to continuous-valued response variables. The model-based approaches are often time-consuming to build and update, and cannot cover as diverse a user range as the memory-based approaches do [22].

2.3 Other Related Approaches

In order to take the advantages of memory-based and model-based approaches, hybrid collaborative filtering methods have been studied recently [14, 22]. [1, 4] unified collaborative filtering and content-based filtering, which achieved significant improvements over the standard approaches. At the same time, in order to solve the data sparsity problem, researchers proposed dimensionality reduction approaches in [15]. The dimensionality-reduction approach addressed the sparsity problem by deleting unrelated or insignificant users or items, which would discard some information of the user-item matrix.

3. SIMILARITY COMPUTATION

This section briefly introduces the similarity computation methods in traditional user-based and item-based collaborative filtering [2, 5, 7, 17] as well as the method proposed in this paper. Given a recommendation system consists of M users and N items, the relationship between users and items is denoted by an $M \times N$ matrix, called the user-item matrix. Every entry in this matrix $r_{m,n}$ represents the score value, r, that user m rates an item n, where $r \in \{1, 2, ..., r_{max}\}$. If user m does not rate the item n, then $r_{m,n} = 0$.

3.1 Pearson Correlation Coefficient

User-based collaborative filtering engaging PCC was used in a number of recommendation systems [18], since it can be easily implemented and can achieve high accuracy when comparing with other similarity computation methods. In user-based collaborative filtering, PCC is employed to define the similarity between two users a and u based on the items they rated in common:

$$Sim(a,u) = \frac{\sum_{i \in I(a) \cap I(u)} (r_{a,i} - \overline{r}_a) \cdot (r_{u,i} - \overline{r}_u)}{\sqrt{\sum_{i \in I(a) \cap I(u)} (r_{a,i} - \overline{r}_a)^2} \cdot \sqrt{\sum_{i \in I(a) \cap I(u)} (r_{u,i} - \overline{r}_u)^2}}, \quad (1)$$

where $Sim(a,u)$ denotes the similarity between user a and user u, and i belongs to the subset of items which user a and user u both rated. $r_{a,i}$ is the rate user a gave item i, and $\overline{r}_a$ represents the average rate of user a. From this definition, user similarity $Sim(a,u)$ is ranging from $[0,1]$, and a larger value means users a and u are more similar.

Item-based methods such as [5, 17] are similar to user-based approaches, and the difference is that item-based methods employ the similarity between the items instead of users. The basic idea in similarity computation between two items i and j is to first isolate the users who have rated both of these items and then apply a similarity computation technique to determine the similarity $Sim(i,j)$ [17]. The PCC-based similarity computation between two items i and j can be described as:

$$Sim(i,j) = \frac{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \overline{r}_i) \cdot (r_{u,j} - \overline{r}_j)}{\sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \overline{r}_i)^2} \cdot \sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,j} - \overline{r}_j)^2}}, \quad (2)$$

where $Sim(i,j)$ is the similarity between item i and item j, and u belongs to the subset of users who both rated item i and item j. $r_{u,i}$ is the rate user u gave item i, and $\overline{r}_i$ represents the average rate of item i. Like user similarity, item similarity $Sim(i,j)$ is also ranging from $[0,1]$.

3.2 Significance Weighting

PCC-based collaborative filtering generally can achieve higher performance than other popular algorithms like VSS [2], since it considers the factor of the differences of user rating styles. However PCC will overestimate the similarities of users who happen to have rated a few items identically, but may not have similar overall preferences [13]. Herlocker et

al. [6, 7] proposed to add a correlation significance weighting factor that would devalue similarity weights that were based on a small number of co-rated items. Herlocker's latest research work [13] proposed to use the following modified similarity computation equation:

$$Sim'(a,u) = \frac{Max(|I_a \cap I_u|, \gamma)}{\gamma} \cdot Sim(a,u). \quad (3)$$

This equation overcomes the problem when only few items are rated in common but in case that when $|I_a \cap I_u|$ is much higher than γ, the similarity $Sim'(a,u)$ will be larger than 1, and even surpass 2 or 3 in worse cases. We use the following equation to solve this problem:

$$Sim'(a,u) = \frac{Min(|I_a \cap I_u|, \gamma)}{\gamma} \cdot Sim(a,u), \quad (4)$$

where $|I_a \cap I_u|$ is the number of items which user a and user u rated in common. This change bounds the similarity $Sim'(a,u)$ to the interval $[0,1]$. Then the similarity between items could be defined as:

$$Sim'(i,j) = \frac{Min(|U_i \cap U_j|, \delta)}{\delta} \cdot Sim(i,j), \quad (5)$$

where $|U_i \cap U_j|$ is the number of users who rated both item i and item j.

4. COLLABORATIVE FILTERING FRAMEWORK

In pratice, the user-item matrix of commercial recommendation system is very sparse and the density of available ratings is often less than 1% [17]. Sparse matrix directly leads to the prediction inaccuracy in traditional user-based or item-based collaborative filtering. Some work applies data smoothing methods to fill the missing values of the user-item matrix. In [22], Xue et al. proposed a cluster-based smoothing method which clusters the users using K-means first, and then predicts all the missing data based on the ratings of Top-N most similar users in the similar clusters. The simulation shows this method could generate better results than other collaborative filtering algorithms. But cluster-based method limits the diversity of users in each cluster, and the clustering results of K-means relies on the pre-selected K users. Furthermore, if a user does not have enough similar users, then Top-N algorithm generates a lot of dissimilar users which definitely will decrease the prediction accuracy of the active users.

According to the analysis above, we propose a novel effective missing data prediction algorithm which predicts the missing data when it fits the criteria we set. Otherwise, we will not predict the missing data and keep the value of the missing data to be zero. As illustrated in Fig. 1(a), before we predict the missing data, the user-item matrix is a very sparse matrix and every user only rates few items with $r_{u,i}$; at the same time, other unrated data are covered with shade. Using this sparse matrix to predict ratings for active users always results in giving bad recommendations to the active users. In our approach, we evaluate every shaded block (missing data) using the available information in Fig. 1(a). For every shaded block, if our algorithm achieves confidence in the prediction, then we give this shaded block a predicted rating value $\widehat{r}_{u,i}$. Otherwise, we set the value of this missing data to zero, as seen in Fig. 1(b).

Accordingly, the collaborative filtering is simplified into two simple questions. The first is "Under what circumstance does our algorithm have confidence to predict the shaded block?" and the second is "How to predict?". The following subsections will answer these two questions.

4.1 Similar Neighbors Selection

Similar neighbors selection is a very important step in predicting missing data. If selected neighbors are dissimilar with the current user, then the prediction of missing data of this user is inaccurate and will finally affect the prediction results of the active users. In order to overcome the flaws of Top-N neighbors selection algorithms, we introduce a threshold η. If the similarity between the neighbor and the current user is larger than η, then this neighbor is selected as the similar user.

For every missing data $r_{u,i}$, a set of similar users $S(u)$ towards user u can be generated according to:

$$S(u) = \{u_a | Sim'(u_a, u) > \eta, u_a \neq u\}, \quad (6)$$

where $Sim'(u_a, u)$ is computed using Eq. (4). At the same time, for every missing data $r_{u,i}$, a set of similar items $S(i)$ towards item i can be generated according to:

$$S(i) = \{i_k | Sim'(i_k, i) > \theta, i_k \neq i\}, \quad (7)$$

where θ is the item similarity threshold, and $Sim'(i_k, i)$ is computed by Eq. (5). The selection of η and θ is an important step since a very big value will always cause the shortage of similar users or items, and a relative small value will bring too many similar users or items.

According to Eqs.(6) and (7), we define that our algorithm will lack enough confidence to predict the missing data $r_{u,i}$ if and only if $S(u) = \emptyset \wedge S(i) = \emptyset$, which means that user u does not have similar users and item i does not have similar items either. Then our algorithm sets the value of this missing data to zero. Otherwise, it will predict the missing data $r_{u,i}$ following the algorithm described in Subsection 4.2.

4.2 Missing Data Prediction

User-based collaborative filtering predicts the missing data using the ratings of similar users and item-based collaborative filtering predicts the missing data using the ratings of similar items. Actually, although users have their own rating style, if an item is a very popular item and has obtained a very high average rating from other users, then the active user will have a high probability to give this item a good rating too. Hence, predicting missing data only using user-based approaches or only using item-based approaches will potentially ignore valuable information that will make the prediction more accurate. We propose to systematically combine user-based and item-based approaches, and take advantage of user correlations and item correlations in the user-item matrix.

Given the missing data $r_{u,i}$, according to Eq. (6) and Eq. (7), if $S(u) \neq \emptyset \wedge S(i) \neq \emptyset$, the prediction of missing

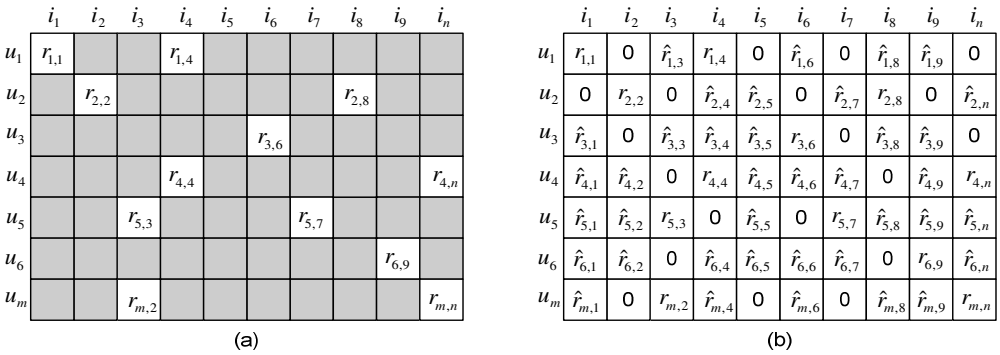

Figure 1: (a) The user-item matrix ($m \times n$) before missing data prediction. (b) The user-item matrix ($m \times n$) after missing data prediction.

data $P(r_{u,i})$ is defined as:

$$P(r_{u,i}) = \lambda \times (\overline{u} + \frac{\sum_{u_a \in S(u)} Sim'(u_a, u) \cdot (r_{u_a,i} - \overline{u}_a)}{\sum_{u_a \in S(u)} Sim'(u_a, u)}) + (1 - \lambda) \times (\overline{i} + \frac{\sum_{i_k \in S(i)} Sim'(i_k, i) \cdot (r_{u,i_k} - \overline{i}_k)}{\sum_{i_k \in S(i)} Sim'(i_k, i)}), \quad (8)$$

where λ is the parameter in the range of $[0, 1]$. The use of parameter λ allows us to determine how the prediction relies on user-based prediction and item-based prediction. $\lambda = 1$ states that $P(r_{u,i})$ depends completely upon ratings from user-based prediction and $\lambda = 0$ states that $P(r_{u,i})$ depends completely upon ratings from item-based prediction.

In practice, some users do not have similar users and the similarities between these users and all other users are less than the threshold η. Top-N algorithms will ignore this problem and still choose the top n most similar users to predict the missing data. This will definitely decrease the prediction quality of the missing data. In order to predict the missing data as accurate as possible, in case some users do not have similar users, we use the information of similar items instead of users to predict the missing data, and vice versa, as seen in Eq. (9) and Eq. (10). This consideration inspires us to fully utilize the information of user-item matrix as follows:

If $S(u) \neq \emptyset \wedge S(i) = \emptyset$, the prediction of missing data $P(r_{u,i})$ is defined as:

$$P(r_{u,i}) = \overline{u} + \frac{\sum_{u_a \in S(u)} Sim'(u_a, u) \cdot (r_{u_a,i} - \overline{u}_a)}{\sum_{u_a \in S(u)} Sim'(u_a, u)}. \quad (9)$$

If $S(u) = \emptyset \wedge S(i) \neq \emptyset$, the prediction of missing data $P(r_{u,i})$ is defined as:

$$P(r_{u,i}) = \overline{i} + \frac{\sum_{i_k \in S(i)} Sim'(i_k, i) \cdot (r_{u,i_k} - \overline{i}_k)}{\sum_{i_k \in S(i)} Sim'(i_k, i)}. \quad (10)$$

The last possibility is given the missing data $r_{u,i}$, user u does not have similar users and at the same time, item i also does not have similar items. In this situation, we choose not to predict the missing data; otherwise, it will bring negative influence to the prediction of the missing data $r_{u,i}$. That is:

If $S(u) = \emptyset \wedge S(i) = \emptyset$, the prediction of missing data $P(r_{u,i})$ is defined as:

$$P(r_{u,i}) = 0. \quad (11)$$

This consideration is different from all other existing prediction or smoothing methods. They always try to predict all the missing data in the user-item matrix, which will predict some missing data with bad quality.

4.3 Prediction for Active Users

After the missing data is predicted in the user-item matrix, the next step is to predict the ratings for the active users. The prediction process is almost the same as predicting the missing data, and the only difference is in the case for a given active user a; namely, if $S(a) = \emptyset \wedge S(i) = \emptyset$, then predicts the missing data using the following equation:

$$P(r_{a,i}) = \lambda \times \overline{r}_a + (1 - \lambda) \times \overline{r}_i. \quad (12)$$

In other situations, if (1) $S(u) \neq \emptyset \wedge S(i) \neq \emptyset$, (2) $S(u) \neq \emptyset \wedge S(i) = \emptyset$ or (3) $S(u) = \emptyset \wedge S(i) \neq \emptyset$, we use Eq. (8), Eq. (9) and Eq. (10) to predict $r_{a,i}$, respectively.

4.4 Parameter Discussion

The thresholds γ and δ introduced in Section 3 are employed to avoid overestimating the users similarity and items similarity, when there are only few ratings in common. If we set γ and δ too high, most of the similarities between users or items need to be multiplied with the significance weight, and it is not the results we expect. However, if we set γ and δ too low, it is also not reasonable because the overestimate problem still exists. Tuning these parameters is important to achieving a good prediction results.

The thresholds η and θ introduced in Section 4.1 also play an important role in our collaborative filtering algorithm. If η and θ are set too high, less missing data need to be predicted; if they are set too low, a lot of missing data need to be predicted. In the case when $\eta = 1$ and $\theta = 1$, our approach will not predict any missing data, and this algorithm becomes the general collaborative filtering without data smoothing. In the case when $\eta = 0$ and $\theta = 0$, our approach will predict all the missing data, and this algorithm

Table 1: The relationship between parameters with other CF approaches

Lambda	Eta	Theta	Related CF Approaches
1	1	1	User-based CF without missing data prediction
0	1	1	Item-based CF without missing data prediction
1	0	0	User-based CF with all the missing data predicted
0	0	0	Item-based CF with all the missing data predicted

converges to the Top-N neighbors selection algorithms, except the number N here includes all the neighbors. In order to simplify our model, we set $\eta = \theta$ in all the simulations.

Finally, parameter λ introduced in Section 4.2 is the last parameter we need to tune, and it is also the most important one. λ determines how closely the rating prediction relies on user information or item information. As discussed before, $\lambda = 1$ states that $P(r_{u,i})$ depends completely upon ratings from user-based prediction and $\lambda = 0$ states that $P(r_{u,i})$ depends completely upon ratings from item-based prediction. This physical interpretation also helps us to tune λ accordingly.

With the changes of parameters, several other famous collaborative filtering methods become special cases in our approach as illustrated in Table 1.

5. EMPIRICAL ANALYSIS

We conduct several experiments to measure the recommendation quality of our new approach for collaborative filtering with other methods, and address the experiments as the following questions: (1) How does our approach compare with traditional user-based and item-based collaborative filtering methods? (2) What is the performance comparison between our effective missing data prediction approach and other algorithms which predict every missing data? (3) How does significance weighting affect the accuracy of prediction? (4) How do the thresholds η and θ affect the accuracy of prediction? How many missing data are predicted by our algorithm, and what is the comparison of our algorithm with the algorithms that predict all the missing data or no missing data? (5) How does the parameter λ affect the accuracy of prediction? and (6) How does our approach compare with the published state-of-the-art collaborative filtering algorithms?

In the following, Section 5.3 gives answers to questions 1 and 6, Section 5.4 addresses question 2, and Section 5.5 describes experiment for the questions 3 to 5.

5.1 Dataset

Two datasets from movie rating are applied in our experiments: MovieLens[3] and EachMovie[4]. We only report the simulation results of MovieLens due to the space limitation. Similar results can be observed from the EachMovie application.

MovieLens is a famous Web-based research recommender system. It contains 100,000 ratings (1-5 scales) rated by 943 users on 1682 movies, and each user at least rated 20 movies. The density of the user-item matrix is:

$$\frac{100000}{943 \times 1682} = 6.30\%.$$

[3]http://www.cs.umn.edu/Research/GroupLens/.

[4]http://www.research.digital.com/SRC/EachMovie/. It is retired by Hewlett-Packard (HP), but a postprocessed copy can be found on http://guir.berkeley.edu/projects/swami/.

Table 2: Statistics of Dataset MovieLens

Statistics	User	Item
Min. Num. of Ratings	20	1
Max. Num. of Ratings	737	583
Avg. Num. of Ratings	106.04	59.45

Table 3: MAE comparison with other approaches (A smaller MAE value means a better performance).

Training Users	Methods	Given5	Given10	Given20
MovieLens 300	EMDP	**0.784**	**0.765**	**0.755**
	UPCC	0.838	0.814	0.802
	IPCC	0.870	0.838	0.813
MovieLens 200	EMDP	**0.796**	**0.770**	**0.761**
	UPCC	0.843	0.822	0.807
	IPCC	0.855	0.834	0.812
MovieLens 100	EMDP	**0.811**	**0.778**	**0.769**
	UPCC	0.876	0.847	0.811
	IPCC	0.890	0.850	0.824

The statistics of dataset MovieLens is summarized in Table 2.

We extract a subset of 500 users from the dataset, and divide it into two parts: select 300 users as the training users (100, 200, 300 users respectively), and the rest 200 users as the active (testing) users. As to the active users, we vary the number of rated items provided by the active users from 5, 10, to 20, and give the name Given5, Given10 and Given20, respectively.

5.2 Metrics

We use the Mean Absolute Error (MAE) metrics to measure the prediction quality of our proposed approach with other collaborative filtering methods. MAE is defined as:

$$MAE = \frac{\sum_{u,i} |r_{u,i} - \widehat{r}_{u,i}|}{N}, \tag{13}$$

where $r_{u,i}$ denotes the rating that user u gave to item i, and $\widehat{r}_{u,i}$ denotes the rating that user u gave to item i which is predicted by our approach, and N denotes the number of tested ratings.

5.3 Comparison

In order to show the performance increase of our effective missing data prediction (EMDP) algorithm, we compare our algorithm with some traditional algorithms: user-based algorithm using PCC (UPCC) and item-based algorithm using PCC (IPCC). The parameters or thresholds for the experiments are empirically set as follows: $\lambda = 0.7$, $\gamma = 30$, $\delta = 25$, $\eta = \theta = 0.4$.

In Table 3, we observe that our new approach significantly improves the recommendation quality of collaborative filtering, and outperforms UPCC and IPCC consistently.

Table 4: MAE comparison with state-of-the-arts algorithms (A smaller MAE value means a better performance).

Num. of Training Users	100			200			300		
Ratings Given for Active Users	5	10	20	5	10	20	5	10	20
EMDP	**0.807**	**0.769**	**0.765**	**0.793**	**0.760**	**0.751**	**0.788**	**0.754**	**0.746**
SF	0.847	0.774	0.792	0.827	0.773	0.783	0.804	0.761	0.769
SCBPCC	0.848	0.819	0.789	0.831	0.813	0.784	0.822	0.810	0.778
AM	0.963	0.922	0.887	0.849	0.837	0.815	0.820	0.822	0.796
PD	0.849	0.817	0.808	0.836	0.815	0.792	0.827	0.815	0.789
PCC	0.874	0.836	0.818	0.859	0.829	0.813	0.849	0.841	0.820

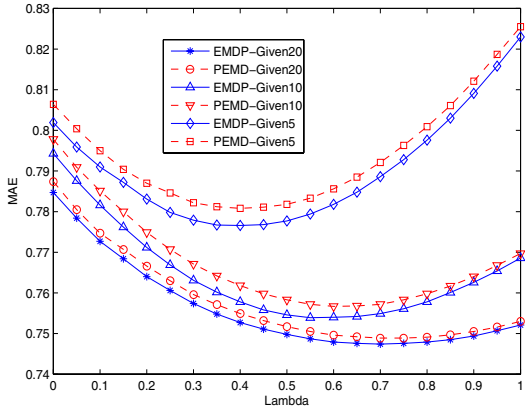

Figure 2: MAE Comparison of EMDP and PEMD (A smaller MAE value means a better performance).

Next, in order to compare our approach with other state-of-the-arts algorithms, we follow the exact evaluation procedures which were described in [21, 22] by extracting a subset of 500 users with more than 40 ratings. Table 4 summarizes our experimental results. We compare with the following algorithms: Similarity Fusion (SF) [21], Smoothing and Cluster-Based PCC (SCBPCC) [22], the Aspect Model (AM) [9], Personality Diagnosis (PD) [14] and the user-based PCC [2]. Our method outperforms all other competitive algorithms in various configurations.

5.4 Impact of Missing Data Prediction

Our algorithm incorporates the option not to predict the missing data if it does not meet the criteria set in Section 4.1 and Section 4.2. In addition, it alleviates the potential negative influences from bad prediction on the missing data. To demonstrate the effectiveness of our approach, we first conduct a set of simulations on our effective missing data prediction approach. The number of training users is 300, where we set $\gamma = 30$, $\delta = 25$, $\eta = \theta = 0.5$, and vary λ from zero to one with a step value of 0.05. We then plot the graph with the ratings of active users of Given5, Given10 and Given20, respectively. As to the method in predicting every missing data (PEMD), we use the same algorithm, and keep the configurations the same as EMDP except for Eq. (11). In PEMD, when $S(u) = \emptyset$ and $S(i) = \emptyset$, we predict the missing data $r_{u,i}$ using the nearest neighbors of the missing data instead of setting the value to zero. In this experiment, we set the number of nearest neighbors to 10. The intention of this experiment is to compare the performance of our EMDP algorithm with PEMD under the same configurations. In other words, we intend to determine the effectiveness of our missing data prediction algorithm, and whether our approach is better than the approach which will predict every missing data or not.

In Fig. 2, the star, up triangle, and diamond in solid line represent the EMDP algorithm in Given20, Given10 and Given5 ratings respectively, and the circle, down triangle, and square in dashed line represent the PEMD algorithm in Given20, Given10 and Given5 ratings respectively. All the solid lines are below the respectively comparative dashed lines, indicating our effective missing data prediction algorithm performs better than the algorithm which predict every missing data, and predicting missing data selectively is indeed a more effective method.

5.5 Impact of Parameters

5.5.1 γ and δ in Significance Weighting

Significance weighting makes the similarity computation more reasonable in practice and devalues some similarities which look similar but are actually not, and the simulation results in Fig. 3 shows the significance weighting will promote the collaborative filtering performance.

In this experiment, we first evaluate the influence of γ, and select 300 training users, then set $\lambda = 0.7$, $\eta = \theta = 0.5$, $\delta = 26$. We vary the range of γ from 0 to 50 with a step value of 2. Fig. 3(a),(b),(c) shows how γ affects MAE when given ratings 20, 10, 5 respectively, and Fig. 3(d) shows that the value of γ also impacts the density of the user-item matrix in the process of missing data prediction. The density of the user-item matrix will decrease according to the increase of the value of γ. More experiments show that δ has the same features and impacts on MAE and matrix density as γ; however, we do not include the simulation results due to the space limitation.

5.5.2 Impact of λ

Parameter λ balances the information from users and items. It takes advantages from these two types of collaborative filtering methods. If $\lambda = 1$, we only extract information from users, and if $\lambda = 0$, we only mine valuable information from items. In other cases, we fuse information from users and items to predict the missing data and furthermore, to predict for active users.

Fig. 4 shows the impacts of λ on MAE. In this experiment, we test 300 training users, 200 training users and 100 training users and report the experiment results in Fig. 4(a), Fig. 4(b) and Fig. 4(c) respectively. The initial values of other parameters or thresholds are: $\eta = \theta = 0.5$, $\gamma = 30$, $\delta = 25$.

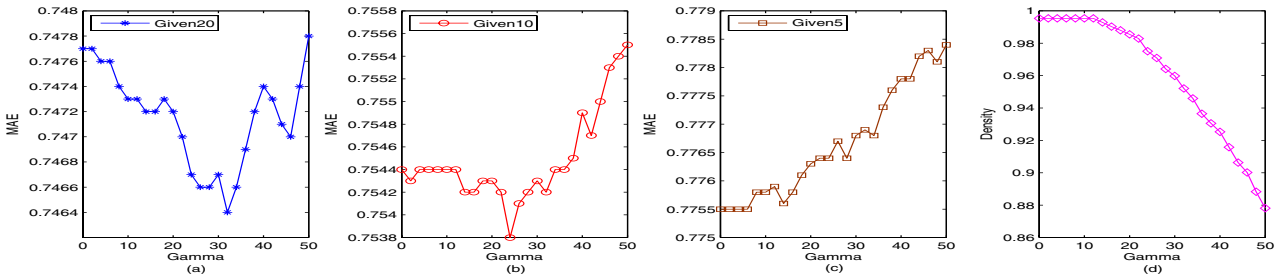

Figure 3: Impact of Gamma on MAE and Matrix Density

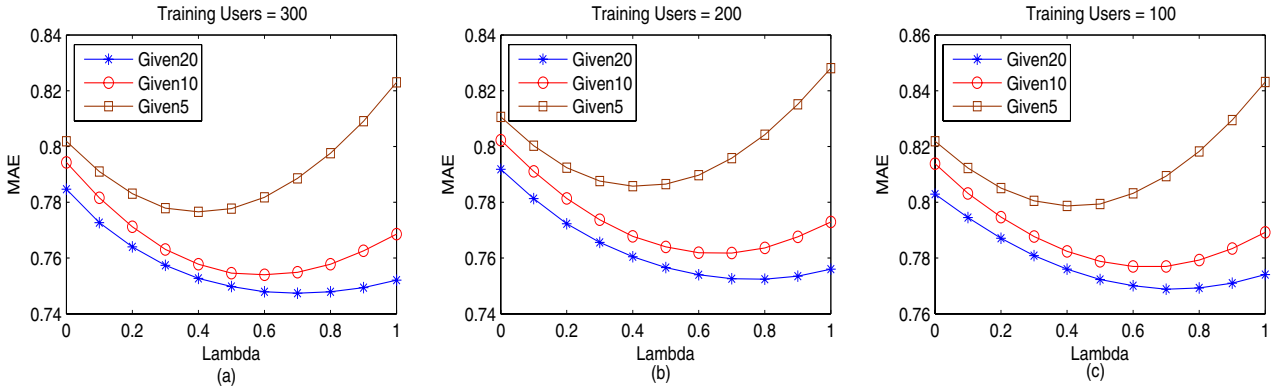

Figure 4: Impact of Lambda on MAE

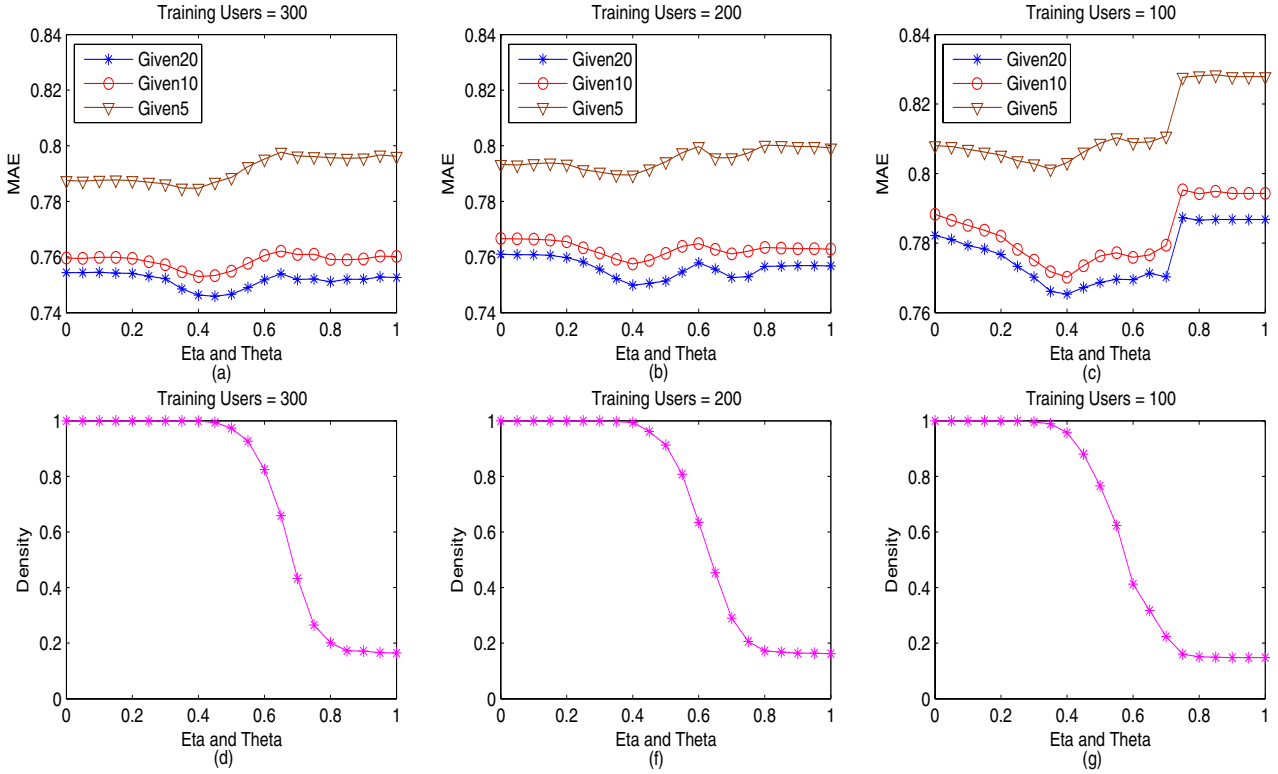

Figure 5: Impact of Eta and Theta on MAE and Density

Observed from Fig. 4, we draw the conclusion that the value of λ impacts the recommendation results significantly, which demonstrates that combining the user-based method with the item-based method will greatly improve the recommendation accuracy. Another interesting observation is when following the increase of the number of ratings given (from 5 to 10, and from 10 to 20), the value of $\arg\min_\lambda(MAE)$ of each curve in Fig. 4 shifts from 0.3 to 0.8 smoothly. This implies the information for users is more important than that for items if more ratings for active users are given. On the other hand, the information for items would be more important if less ratings for active users are available; however, less ratings for active users will lead to more inaccuracy of the recommendation results.

5.5.3 Impact of η and θ

η and θ also play a very important role in our collaborative filtering approach. As discussed in Section 4, η and θ directly determine how many missing data need to be predicted. If η and θ are set too high, most of the missing data cannot be predicted since many users will not have similar users, and many items will not have similar items either. On the other hand, if η and θ are set too low, every user or item will obtain too many similar users or items, which causes the computation inaccuracy and increases the computing cost. Accordingly, selecting proper values for η and θ is as critical as determining the value for λ. In order to simplify our model, we set $\eta = \theta$ as employed in our experiments.

In the next experiment, we select 500 users from MovieLens dataset and extract 300 users for training users and other 200 as the active users. The initial values for every parameter and threshold are: $\lambda = 0.7$, $\gamma = 30$, $\delta = 25$. We vary the values of η and θ from 0 to 1 with a step value of 0.05. For each training user set (100, 200, 300 users respectively), we compute the MAE and density of the user-item matrix. The results are showed in Fig. 5.

As showed in Fig. 5(a), given 300 training users and given 20 ratings for every active user, this algorithm will achieve the best performance around $\eta = \theta = 0.50$, and the related density of user-item matrix in Fig. 5(d) is 92.64% which shows that 7.36% missing data of this user-item matrix are not predicted. In this experiment, the number of data that was not predicted is $0.0736 \times 500 \times 1000 = 36800$. We observe that around $\eta = \theta = 0.70$, this algorithm already achieves a very good MAE value which is almost the same as the best MAE values in Fig. 5(b). The related matrix density is 29.00%, which illustrates that more than 70% data of user-item matrix are not predicted. Nevertheless, the algorithm can already achieve satisfactory performance.

6. CONCLUSIONS

In this paper, we propose an effective missing data prediction algorithm for collaborative filtering. By judging whether a user (an item) has other similar users (items), our approach determines whether to predict the missing data and how to predict the missing data by using information of users, items or both. Traditional user-based collaborative filtering and item-based collaborative filtering approaches are two subsets of our new approach. Empirical analysis shows that our proposed EMDP algorithm for collaborative filtering outperforms other state-of-the-art collaborative filtering approaches.

For future work, we plan to conduct more research on the relationship between user information and item information since our simulations show the algorithm combining these two kinds of information generates better performance. Lastly, another research topic worthy of studying is the scalability analysis of our algorithm.

7. ACKNOWLEDGMENTS

We thank Mr. Shikui Tu, Ms. Tu Zhou and Mr. Haixuan Yang for many valuable discussions on this topic. This work is fully supported by two grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK4205/04E and Project No. CUHK4235/04E).

8. REFERENCES

[1] J. Basilico and T. Hofmann. Unifying collaborative and content-based filtering. In *Proc. of ICML*, 2004.
[2] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. of UAI*, 1998.
[3] J. Canny. Collaborative filtering with privacy via factor analysis. In *Proc. of SIGIR*, 2002.
[4] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin. Combining content-based and collaborative filters in an online newspaper. In *Proc. of SIGIR*, 1999.
[5] M. Deshpande and G. Karypis. Item-based top-n recommendation. *ACM Trans. Inf. Syst.*, 22(1):143–177, 2004.
[6] J. Herlocker, J. A. Konstan, and J. Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, 5:287–310, 2002.
[7] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proc. of SIGIR*, 1999.
[8] T. Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proc. of SIGIR*, 2003.
[9] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):89–115, 2004.
[10] R. Jin, J. Y. Chai, and L. Si. An automatic weighting scheme for collaborative filtering. In *Proc. of SIGIR*, 2004.
[11] A. Kohrs and B. Merialdo. Clustering for collaborative filtering applications. In *Proc. of CIMCA*, 1999.
[12] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, pages 76–80, Jan/Feb 2003.
[13] M. R. McLaughlin and J. L. Herlocker. A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *Proc. of SIGIR*, 2004.
[14] D. M. Pennock, E. Horvitz, S. Lawrence, and C. L. Giles. Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach. In *Proc. of UAI*, 2000.
[15] J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proc. of ICML*, 2005.
[16] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proc. of ACM Conference on Computer Supported Cooperative Work*, 1994.
[17] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proc. of the WWW Conference*, 2001.
[18] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating word of mouth. In *Proc. of SIGCHI Conference on Human Factors in Computing Systems*, 1995.
[19] L. Si and R. Jin. Flexible mixture model for collaborative filtering. In *Proc. of ICML*, 2003.
[20] L. H. Ungar and D. P. Foster. Clustering methods for collaborative filtering. In *Proc. Workshop on Recommendation System at the 15th National Conf. on Artificial Intelligence*, 1998.
[21] J. Wang, A. P. de Vries, and M. J. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proc. of SIGIR*, 2006.
[22] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, and Z. Chen. Scalable collaborative filtering using cluster-based smoothing. In *Proc. of SIGIR*, 2005.