Incorporating Personalized Contextual Information in Item-based Collaborative Filtering Recommendation

Min Gao College of Computer Science Chongqing University, Chongqing, China, 400044 gaomin@cqu.edu.cn Zhongfu Wu College of Computer Science Chongqing University, Chongqing, China, 400044 wzf@cqu.edu.cn

Abstract-After reviewing the prior work and problem of collaborative filtering recommendation approaches, an approach incorporating personalized contextual information in item-based collaborative filtering is proposed solve the problem. The approach provides to recommendations based on user personalized contextual information besides the typical information on users and items used in most of the current recommendation systems. In this paper, several approaches are proposed to calculate context-based item differences, learn personalized contextual information for every user and predict ratings based on well-known item-based collaborative filtering Slope One. Finally, we experimentally evaluate our approach and compare it to Slope One. The experimental results show that our approach provide more precision recommendations than Slope One.

Index Terms—recommendation; context; personalization; collaboration filtering

I. INTRODUCTION

Due to the explosive growth of the Web, personalized recommendation systems have been widely accepted by users. Users offer feedback on purchased or consumed items, and recommender systems use the information to predict their preferences for yet unseen items and subsequently recommend items with the highest predicted ratings for а user. Personalized have recommendation approaches gained great momentum both in the commercial and research areas [1].

Personalized recommendation is defined as automatic adjustment, re-structuring, and the presentation of tailored information content for individuals [2]. It builds customer loyalty by creating meaningful one-to-one relationships and understanding user needs in different contexts [3, 4]. It is a process of gathering and analyzing user information for delivering the right information at the right time [5]. In summary, personalized recommendation is the ability to provide tailored content and services to individuals based on knowledge about their preferences and behaviors [6-8]. For example, Amazon is one of the most successful personalized recommendation systems [9, 10].

Lots of research has been done in this domain, rulecollaborative, and hybrid based, content-based, personalized recommendation approaches are proposed. Till now, collaborative filtering is the most popular approach. Traditional collaborative filtering is named user-based collaborative filtering. It was the most successful used technique for building recommendation systems in past [11, 12]. However, it suffers serious scalability problem. It has been proved experimentally that Item-based collaborative filtering can solve the problem. Item-based collaborative filtering is proposed to build offline an item-item similarity matrix for prediction. Since it uses a pre-computed model, it will recommend items quickly. They have been shown to produce recommendation results that in some cases are comparable to traditional user-based collaborative filtering recommender systems [12], but the recommendation results have not been distinctly improved.

In the paper, we incorporate personalized contextual information into item-based filtering approach to improve the recommendation results. Contextual information has been recognized as important factor in recommender systems [13-16]. Personalized contextual information for every user is a subset of contextual parameters in a certain system. It is significant factor for providing recommendations as well. For example, in the case of personalized news delivery system, time and place are usually important contextual information to determine what news needs to be recommended to users, but they are not that useful for some special users working at SOHO (small office/home office). In our research, we

^{*} This work is supported by the Postdoctoral Science Foundation of China under Grant No. 20080440699, and the Natural Science Foundation Project of CQ CSTC under Grant No. 2008BB2183, the Education Science and Technology Research Program of Chongqing Municipal Education Commission under Grant No. KJ071604, and the 11th 5-year Plan Program of Chongqing Municipal Education Commission under Grant No. 2008-ZJ-064. This research is an extended version of the paper presented at the International Symposium on Intelligent Information Systems and Applications 2009[1].

incorporate personalized contextual information to itembased collaborative filtering recommendation.

In this paper we seek to take a first step towards personalized contextual information-based recommendation. We firstly review the state of the art in collaborative filtering approaches including Slope One (Section 2), and then propose a context-based item difference analysis, a personalized context analysis, and a rating estimating method based on Slope One algorithm in Section 3. In Section 4, we experimentally evaluate our recommendation results and compare them to Slope One. Finally, we draw conclusions in Section 5.

II. BACKGROUND OF COLLABORATIVE FILTERING

Collaborative filtering is the most successful recommendation technique to date [17, 18]. In a typical collaborative filtering scenario, there is a rating matrix which includes a list of m users (as rows) and a list of n items (as columns) and lots of ratings $r_{u,i}$. A rating $r_{u,i}$ means how user u likes item i which is always scaled from 1 to 5 or 1 to 10. For example, table 1 is a rating matrix *RM*. There are 5 users, 6 items, and several ratings in the matrix. $r_{u,i}=\emptyset$ means the item i are not rated by the corresponding user u. The key step of it is how to extrapolate unknown ratings. It is supposed that users mainly interested in high unknown ratings.

TABLE I. RM: AN EXAMPLE OF RATING MATRIX

	i	i	i	i	i	i
	1	2	3	4	5	6
<i>u</i> 1	5	4	3	4	2	Ø
<i>u</i> 2	3	3	4	Ø	3	4
и 3	Ø	4	Ø	2	3	Ø
и 4	3	2	3	4	5	3
и 5	Ø	Ø	3	Ø	2	Ø

A. User-based Collaborative Filtering

The basic idea of traditional collaborative filtering is to predict the rating of a certain item for a target user based on the opinions of other like-minded users. This traditional filtering is also called user-based collaborative filtering (See Fig. 1).



Fig. 1 Collaborative filtering (User-based)

It makes recommendations by finding correlations among shared likes and dislikes of users. For example, for the target user A, it firstly finds the similar users by the comparisons of users' profiles, e.g., their ratings. Then it filters items which the similar users of user Aliked. For example, in a book recommendation system, the system tries to find similar users to user u; only books that are most liked by similar users will be recommended.

It was very successful in past [11, 12], but some potential challenges have been revealed [18] such as scalability problem. Scalability means that its computation grows rapidly with the number of users and items. It has been proved that item-based collaborative filtering can solve the problem [19]. Researchers have proposed several approaches for similarity computation and rating prediction, such as cosine similarity (See formula 1) and Pearson correlation similarity (See formula 2)

$$Sim(u,v) = \frac{\sum_{i \in I(u) \cap I(v)} (r_{u,i}r_{v,i})}{\sqrt{\sum_{i \in I(u) \cap I(v)} r_{u,i}^{2}} \sqrt{\sum_{i \in I(u) \cap I(v)} r_{v,i}^{2}}}$$
(1)

$$Sim(u,v) = \frac{\sum_{i \in I(u) \cap I(v)} (r_{u,i} - \overline{r_u})(r_{v,i} - \overline{r_v})}{\sqrt{\sum_{i \in I(u) \cap I(v)} (r_{u,i} - \overline{r_u})^2} \sqrt{\sum_{i \in I(u) \cap I(v)} (r_{v,i} - \overline{r_v})^2}}$$
(2)

Here, *u* and *v* are two users. I(u) is the set of items which have been rated by user *u*. Formally, $I(u) = \{i | r_{u,i} \neq 0, i \in I\}$. $I(u) \cap I(v)$ represents a set including the items which have been rated by both user *u* and *v*.

For example, given a rating matrix RM, $I(u_1) \cap I(u_2)$ is $\{i_1, i_2, i_3, i_5\}$, $|I(u_1) \cap I(u_2)|=4$. According to (1),

$$sim(u_{1}, u_{2}) = \frac{\sum_{i \in I(u_{1}) \cap I(u_{2})} (r_{u_{1},i}r_{u_{2},i})}{\sqrt{\sum_{i \in I(u_{1}) \cap I(u_{2})} r_{u_{1},i}^{2}} \sqrt{\sum_{i \in I(u_{1}) \cap I(u_{2})} r_{u_{2},i}^{2}}}$$
$$= \frac{(r_{u_{1},i_{1}}r_{u_{2},i_{1}}) + (r_{u_{1},i_{2}}r_{u_{2},i_{2}}) + (r_{u_{1},i_{3}}r_{u_{2},i_{3}}) + (r_{u_{1},i_{5}}r_{u_{2},i_{5}})}{\sqrt{r_{u_{1},i_{1}}^{2} + r_{u_{1},i_{2}}^{2} + r_{u_{1},i_{3}}^{2} + r_{u_{1},i_{5}}^{2}}} \left[\sqrt{r_{u_{2},i_{1}}^{2} + r_{u_{2},i_{2}}^{2} + r_{u_{2},i_{5}}^{2}} + r_{u_{2},i_{5}}^{2}} \right]$$

≈0.93.

B. Item-based Collaborative Filtering

Item-based collaborative filtering is proposed by Sarwar[18] in 2001. It is to compute the similarity between items and then to select the most similar items for prediction (See Fig. 2).



Fig. 2 Collaborative filtering (Item-based)

Because the item similarities are more stable than the users' similarities are, they can be computed at a long time phase. Since it uses a pre-computed model, it will recommend a set of items quickly. Similarly to the user similarity computation approaches, Cosine similarity (See formula 3) and Pearson correlation similarity (See formula 4) are also used in item similarity computation.

$$Sim(i, j) = \frac{\sum_{u \in U(i) \cap U(j)} (r_{u,i} * r_{u,j})}{\sqrt{\sum_{u \in U(i) \cap U(j)} r_{u,i}^{2}} \sqrt{\sum_{u \in U(i) \cap U(j)} r_{u,j}^{2}}}$$
(3)

$$Sim(i, j) = \frac{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \overline{r_u})(r_{u,j} - \overline{r_u})}{\sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \overline{r_u})^2} \sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,j} - \overline{r_u})^2}}$$
(4)

Here U(i) includes all users who have rated on item *i*. Formally, $U(i) = \{u | r_{u,i} \neq 0\}$. $U(i) \cap U(j)$ includes the users who have rated on both item *i* and *j*. $\overline{r_u}$ is the average of user *u*'s ratings. And also there are a number ways to estimate prediction, the most important step in a collaborative filtering system, such as weighted sum (See formula 5) and regression (See formula 6 in Section *B*). Here S(i) includes all similar items of item *i*.

$$p_{u,i} = \frac{\sum_{j \in S(i)} (sim(i, j) * r_{u,j})}{\sum_{j \in S(i)} |sim(i, j)|}$$
(5)

For example, in the rating matrix RM, $U(i_1) \cap U(i_6)$ is $\{u_2, u_4\}$, and $|U(i_1) \cap U(i_6)|$ is 2. According to (3)

$$Sim(i_{1}, i_{6}) = \frac{\sum_{u \in U(i_{1}) \cap U(i_{6})} (r_{u,i_{1}} * r_{u,i_{6}})}{\sqrt{\sum_{u \in U(i_{1}) \cap U(i_{6})} r_{u,i_{1}}^{2}} \sqrt{\sum_{u \in U(i_{1}) \cap U(i_{6})} r_{u,i_{6}}^{2}}}$$

$$=\frac{(r_{u_{1},i_{1}}*r_{u_{1},i_{6}})+(r_{u_{2},i_{1}}*r_{u_{2},i_{6}})+(r_{u_{4},i_{1}}*r_{u_{4},i_{6}})}{\sqrt{r_{u_{1},i_{1}}^{2}+r_{u_{2},i_{1}}^{2}+r_{u_{4},i_{1}}^{2}}\cdot\sqrt{r_{u_{1},i_{6}}^{2}+r_{u_{2},i_{6}}^{2}+r_{u_{4},i_{6}}^{2}}}\approx0.989$$

 $Sim(i_2, i_6) \approx 0.998$, $Sim(i_3, i_6) = 1$, $Sim(i_4, i_6) = 1$, $Sim(i_5, i_6) \approx 0.926$. Note that Sim(i, j) = Sim(j, i). According to (5),

$$p_{u_{1},i_{6}} = \frac{\sum_{j \in S(i_{6})} (sim(i_{6}, j) * r_{u_{1},j})}{\sum_{j \in S(i_{6})} |sim(i_{6}, j)|}$$
$$= \frac{(sim(i_{6},i_{1}) * r_{u_{1},i_{1}}) + \dots + (sim(i_{6},i_{5}) * r_{u_{1},i_{5}})}{sim(i_{6},i_{1}) + \dots + sim(i_{6},i_{5})}$$
$$\approx 3.62$$

C. Slope One Scheme

The Slope One is a typical item-based collaborative filtering approach. It was proposed at February, 2005 by Lemire and Maclanchlan. It works on comparing the intuitive principle of a popular differential between items [17]. It computes the deviation between items rather than similarity. The deviation of item *i* and *j* $d_{i,j}$ is computed by the average difference between item arrays of *i* and *j* (See formula 7). In turn, the deviation of items will be used to predict an unknown item, given their ratings of the other. The prediction is based on a linear regression model (See formula 6).

$$r_{u,n} = \overline{r_u} + \overline{d} \tag{6}$$

Here $r_{u,n}$ is a unknown rating, $\overline{r_u}$ is the average of all

known rating $r_{u,i}$ of user u, \overline{d} is the average of $d_{i,n}$. It has been proved that the Slope One scheme achieves accuracy comparable to that of the adjusted cosine itembased and Pearson scheme. The Slope One has won the wide attention of researchers and companies because it is simple and efficient. It is used by the Bell/MSN Website inDiscover.net as of Nov. 2004 [17].

D. The problem of Slope One

The Slope One Scheme does not take additional personalized contextual information into consideration. However, it is significant for recommendation. On one hand a user usually has different preferences in different context; and on the other hand some context is most important for him/her, but others are not.

Context is "any information that can be used to characterize the situation of an entity" [20, 21]. Context has several contextual parameters in a certain system. Personalized contextual information for a user includes the contextual parameter that the user is most sensitive to. However, if the user is not sensitive to any parameter, then all contextual parameters are his/her personalized contextual information. For instance, in the case of personalized content delivery system, personalized contextual information can be {time}, {place}, or {time and place}. In order to provide more accurate recommendations, we proposed a personalized contextual information-based (PCI-based) approach to incorporate personalized contextual information into Slope One.

III. PERSONALIZED CONTEXTUAL INFORMATION-BASED RECOMMENDATION

Contextual information in recommendation systems is additional information, besides information on Users and Items. It is relevant in identifying pertinent subsets of data, building richer rating estimation models, and providing various types of constraints on recommendation outcomes [14]. Personalized contextual information for a user is the contextual parameter that the user is sensitive to.

Our vision of the PCI-based recommendation is an extension of item-based collaborative recommendation. In this section we analyze how to obtain context-based item differences, how to identify personalized context parameters, and rating estimation.

Suppose that there are a training dataset DS and test dataset TS, in order to identify personalized context parameters for users, we divide the DS to sub-training dataset D and sub-test data set T.

A. Context-based Item Difference Computation

Item difference matrix includes all deviations between items. Context-based item difference matrixes are the deviations in different context. They are the base of personalized contextual information analysis and personalized contextual information-based prediction. It takes two steps to get them.

Step1. For training dataset D, we extract subset $\{Dc_l, Dc_2, ..., Dc_n\}$ in different context $\{C_l, C_2, ..., C_n\}$ from D. Every subset D_k includes the data belongs to a certain context C_k . For example, suppose RM is a dataset D, Table 2 is a subset Dc_l of its.

Table 2. A	An exampl	e subset DI	of RM
------------	-----------	-------------	-------

	i	i	i	i	i	i
	1	2	3	4	5	6
<i>u</i> 1	5	4	3	4	2	Ø
<i>u</i> 2	3	3	4	Ø	3	4
и 3	Ø	4	Ø	2	3	Ø

Step2. For every two items *i* and *j* of Dc_k , to calculate average deviation $d_{i,j,c,k}$ of them using formula 7 of Slope One scheme.

$$d_{i,j,c_k} = \sum_{\substack{u \in U(i) \cap U(j) \\ U(i) \in Dc_k \land U(j) \in Dc_k}} \frac{(r_{u,i} - r_{u,j})}{|U(i) \cap U(j)|}$$
(7) (7)

For example, given a rating matrix RM, $U(i_1) \cap U(i_2)$ is $\{u_1, u_2, u_4\}$, and $|U(i_1) \cap U(i_2)|$ is 3. Thus,

$$\begin{aligned} d_{i_1,i_2} &= \\ \frac{1}{3}((r_{u_1,i_1} - r_{u_1,i_2}) + (r_{u_2,i_1} - r_{u_2,i_2}) + (r_{u_4,i_1} - r_{u_4,i_2})) \\ &\approx 0.667. \\ d_{i_1,i_2,c_1} &= \frac{1}{2}((r_{u_1,i_1} - r_{u_1,i_2}) + (r_{u_2,i_1} - r_{u_2,i_2})) \\ &\approx 0.5. \end{aligned}$$

All the $d_{i,j}$ forms the context-based item deviation matrix DMc_k for C_k (See formula 8), e.g. DMc_l is the deviation matrix for the context parameter C_l . Note that DM is the matrix for whole dataset D.

$$DM_{c_k} = \begin{pmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,n} \\ d_{2,1} & d_{2,2} & \dots & d_{1,n} \\ \dots & \dots & \dots & \dots \\ d_{n,1} & d_{n,2} & \dots & d_{n,n} \end{pmatrix}$$
(8)

For example, Table 3 is the DM of rating matrix Table 1. And Table 4 is the DMc_1 for D_1 .

Table 3. DM of the rating matrix RM

	<i>i</i> 1	i_2	i3	i 4	i_5	i 6
1	0	0 .67	0. 33	(0. 5	0.5
2	- 0.67	0	- 0.33	(0.5	- 1
3	- 0.33	0 .33	0	1	- 0.25	0
4	0	0	1	(0	1
5	0.5	0.5	0. 25	(0	0 .5
6	0. 5	1	0	1	0.5	0

Table 4. Difference matrix DMc_1 for D_1

	i	i	i	i	i	i
	1	2	3	4	5	6
	0	0	0	0	1	-
1	0	.5	.5	.5	.5	0.5
2	-0.5	0	0	1	1	- 1
3	-0.5	0	0	- 1	1	0
4	0.5	- 1	1	0	0 .5	Ø
5	- 1.5	-1	- 1	- 0.5	0	-1



B. Personalized Context Analysis

Contextual information in a system includes several contextual parameters. The purpose of this procedure is to identify personalized contextual parameter for every user.

Step1. On each subset Dc_k , run a traditional recommendation algorithm A to predict the ratings in subtest training set T, e.g., using Cosine similarity, Pearson correlation similarity, or Slope One.

Step2. Compute performances for prediction results in different context C_k .

Given *P* is a evaluation metric, we calculate *A*'s performance $P_A(u_i, C_k)$ for every user in every context. All these performances for users and contexts comprise a performance matrix (*PM*, see formula 9). *PM* includes users as rows and contexts as columns.

$$PM = \begin{pmatrix} P_{A(1,d_1)} & P_{A(1,d_2)} & \dots & P_{A(1,d_n)} \\ P_{A(2,d_1)} & P_{A(2,d_2)} & \dots & P_{A(2,d_n)} \\ \dots & \dots & \dots & \dots \\ P_{A(n,d_1)} & P_{A(n,d_2)} & \dots & P_{A(n,d_n)} \end{pmatrix}$$
(9) (9)

For example, Table 5 is the PM of RM and its subset.

	a	~	~	~
	C	C	C	C
	1	2	3	4
и	0	0	0	0
1	.85	.72	.64	.7
и	0	0	0	0
2	.73	.83	.76	.81
и	0	0	0	0
3	.68	.74	.8	.82

Table 5. An example of PM

Step3. Get personalized contextual parameter for every user.

There is the best performance value in each row in the PM. The column of the best performance value represents the personalized contextual parameter.

Given that the lower the PM value, the better the recommendation results are, then the personalized contextual parameters for all users are as following table.

Table 6. An example of personalized contextual parameters

U	Personalized Context
ser	Parameter
u_1	C_3
u_2	C_{I}
u_3	C_{I}

C. Personalized Contextual Information-based Prediction

Personalized contextual parameter will determine that which difference matrix will be used in rating prediction. The procedure includes 4 steps.

Step1. For training dataset *DS*, we extract subset $\{DSc_1, DSc_2, ..., DSc_n\}$ in different context $\{C_1, C_2, ..., C_n\}$ from *DS*. Every subset DSc_k includes the data belongs to a certain context C_k .

Step2. Given the target user u, we get his/her personalized contextual parameter C_u , and then get the DSc_u .

Step2. Compute DMc_u for DSc_u according to formula 7.

Step3. Calculate prediction for $r_{u,i}$ by formula 10. According to the Slope One scheme, $r_{u,i} - d_{i,j}$ is a prediction for $r_{u,j}$ according to $r_{u,i}$.

$$p_{u,j} = \frac{\sum_{i \in R_u} (r_{u,i} - d_{i,j})}{|R_u|} = \overline{r_u} + \frac{\sum_{i \in R_u} d_{i,j}}{|R_u|}$$
(10)

Here R_u is all known ratings of user u. $|R_u|$ is the number of ratings in R_u . $d_{i,j}$ is a deviation between item i and j in D_k .

For example, in rating matrix *RM*, R_{u_3} is { i_2, i_4, i_5 }, and $|R_{u_3}|$ is 3. According to (10),

$$p_{u_{3},i_{6}} = \frac{\sum_{i \in R_{u_{3}}} (r_{u_{3},i} - d_{i,i_{6}})}{|R_{u_{3}}|}$$
$$= \frac{(r_{u_{1},i_{2}} - d_{i_{2},i_{6}}) + (r_{u_{1},i_{4}} - d_{i_{4},i_{6}}) + (r_{u_{1},i_{5}} - d_{i_{5},i_{6}})}{3}$$

If we take weight of item deviation into consideration, the prediction for $r_{u,j}$ can be estimated by formula 11. And $|U(i) \cap U(j)|$, the weight for deviation $d_{i,j}$, is the number of the users who have rated on both item *i* and *j*.

$$p_{u,j} = \frac{\sum_{i \in R_u} (r_{u,i} - d_{i,j})^* |U(i) \cap U(j)|}{\sum_{i \in R_u} |U(i) \cap U(j)|}$$
(11)

The steps in Section 3.1 and 3.2 are in pre-processing phases and usually performed "offline". The steps for prediction are computed runtime which only take time to get known ratings and deviations from matrices. It runs fast, and the growth of computation can be represented by a linear trend with the number of know ratings.

IV. EXPERIMENTAL EVALUATION

A. Data Set

We used data from the well-known MovieLens project (http://movielens.umn.edu). MovieLens is a free service provided by GroupLens Research at the University of Minnesota (http://www.movielens.org). It is a web-based research recommendation system that debuted in fall 1997 [18]. Each week hundreds of users visit it to rate and receive recommendations for movies. The site had over 43000 users who had rated more than 3500 different movies.

There are two datasets in the Movielens project. One includes 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 users who joined MovieLens in 2000. Another is data sets consists of 100,000 ratings (1-5) from 943 users on 1682 movies. Each user has rated at least 20 movies.

We selected the second one as our dataset. The data set was divided into a training set (80% of the data) and a test set (20% of the data) five times. These training and test sets are named U1base, U2base, U3base, U4base, and U5base, and U1test, U2 test, U3test, U4test, and U5test respectively.

We used U2base and U2test to analyze the personalized contextual information for every user. And then we used other training sets and test sets to evaluate our algorithm. The contextual parameters in our experiment include age, gender, occupation, zip, and time (work time and rest time).

B. Evaluation Metric

Mean Absolute Error (MAE) is a widely used metric for deviation of predictions from their true values. So we used MAE values to measure the prediction results of our algorithm and the real ratings, and then compare it with Slope One algorithm.

For all predictions $\{p_1, p_2, ..., p_n\}$ and their real ratings $\{r_1, r_2, ..., r_n\}$. MAE is the average of absolute error between all $\{p_i, r_i\}$ pairs (See formula 12). The lower the MAE, the more accurately the predictions are, and the better the recommendation approach is.

$$MAE = \frac{\sum_{i=1}^{N} |p_i - r_i|}{N}$$
(12)

C. Experimental Procedure and Results

Without loss of generality, we started our experiments by first computing personalized contextual parameters for all users. In this procedure, U2base and U2test were used. There are 80,000 ratings from 943 users on 1648 movies in U2base and 20,000 ratings from 653 users on 1420 movies in U2test. And then we used U1, U3, U4, and U5 for full experimental evaluation of our algorithm and Slope One.

1) Results in experimental procedure

a) Context-based Item Deviation Matrix Computing

If the contextual information were not considered, the item deviation matrix DM (See formula 13) would be calculated on whole training set U2base.

$$DM = \begin{pmatrix} 0 & 0.702 & 0.957 & \dots \\ -0.702 & 0 & 0.286 & \dots \\ -0.957 & -0.286 & 0 & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix}$$
(13)

When we set sex as the first contextual information, we got the DMc_1 (See formula 14) from the subset of U2base where the user's sex is male or female. So did $DMc_2, ..., DMc_n$.

$$DMc_{1} = \begin{pmatrix} 0 & 0.793 & 0.9 & \dots \\ -0.793 & 0 & 0.352 & \dots \\ -0.9 & -0.352 & 0 & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix}$$
(14)

b) Context-based Item Deviation Matrix Computing

For every context, we predicted all unknown rating. And then we calculated MAE_u values for every user in every context (See formula 15). Here $p_{u,i}$ is prediction of user u on item i, and $r_{u,i}$ its real ratings.

$$MAE_{u} = \frac{\sum_{i=1}^{n} |p_{u,i} - r_{u,i}|}{n}$$
(15)

At last we got a MAE matrix (MM, see formula 16). For every user, the column name of the minimal value is the personalized contextual parameter for him/her.

$$MM = \begin{array}{cccc} & C_{all} & C_1 & C_2 & \dots \\ & U_1 & 0.869 & 0.884 & 0.856 & \dots \\ & U_2 & 0.701 & 0.615 & 0.659 & \dots \\ & 1.179 & 1.236 & 0.983 & \dots \\ & \dots & \dots & \dots & \dots \end{array}$$
(16)

c) Prediction

Once we got personalized context for every user, we started to the full experiment. We got all context-based item difference matrices of training sets of U1, U3, U4, and U5, and predicted every unknown rating in their test sets.





2) Comparison of Prediction Results

Once we got personalized context for every user, we started to the full experiment. We got all context-based item difference matrices of training sets of U1, U3, U4, and U5, and predicted every unknown rating in their test sets.

There are two notations for the experiments. One is Size of Items which represents how many known ratings of an item in the training sets. For example, 15-40 includes the items which have already more than 15 and less than (or equal to) 40 ratings in average in U1, U3, U4, and U5 training sets. Another one is Size of Users' Ratings which means how many known ratings of a user in the training sets. For example, 25-40 includes the users who have already rated more than 25 and less than (or equal to) 40 items.

To compare our approach with the basic Slope One algorithm and determine the sensitivity of the Size of Items in the training set, we performed the experiments where we computed MAE for different number of items. Our results are shown in Fig.3. The blue columns are MAE for Slope One algorithm, and the purple ones are for our algorithm. It can be observed from the chart that our algorithm out performs the basic Slope One algorithm at all values of item size. For example, at the item size of 40-80, Slope One and our algorithm show MAE of 0.771 and 0.759 respectively. At item size of 150-250, our algorithm get optimum value. But an item size of above 250 both algorithms perform worse. We believe this happens as the regression model suffers from data over fitting at high density levels.



FIG.4. Sensitive of the number of user known ratings

To determine the sensitivity of the number of user known ratings in the training set, we then performed the experiment where we computed MAE for different number of user known ratings. The Fig.4 shows our results. For example, at the sizes of users' ratings 9-25 and 25-40, the MAE values are 0.782 and 0.727 respectively. As can be seen from the chart, the MAE values get better as the size of user known ratings increase.

V. CONCLUSIONS

Recommender systems help users find items they would be interested in. Currently item-based collaborative filtering approaches are most popular in recommender systems. The Slope One is a well-known approach of them. In this paper we analyzed how to compute item differences, and how to learn personalized contextual information and predict ratings for unknown items based on Slope One approach. Experimental results show that personalized contextual information is helpful to improve the prediction results of Slope One algorithm.

In the research, we only choose a contextual parameter for a user and do not take its weight into consideration. In the future, more parameters will be considered to achieve a weighted personalized contextual information-based prediction.

ACKNOWLEDGMENT

We gratefully acknowledge valuable input from Tao Zhou, MSc student, from the School of Computer Science, Chongqing University, and Guoxi Cui, PhD candidate in Information Systems, from Informatics Research Center, the University of Reading.

REFERENCES

- Jiang, F. and M. Gao, Collaborative Filtering Approach based on Item and Personalized Contextual Information. International Symposium on Intelligent Information Systems and Applications, 2009, In Press.
- [2] Eirinaki, M. and M. Vazirgiannis, Web mining for web personalization. ACM Transactions on Internet Technology, 2003. 3(1): p. 1-27.
- [3] Perugini, S. and N. Ramakrishnan, Personalizing web sites with mixed-initiative interaction. IT Professional, 2003. 5(2): p. 9-15.
- [4] Riecken, D., Introduction: personalized views of personalization. Communications of the ACM, 2000. 43(8): p. 26-28.
- [5] Frias-Martinez, E., et al., Automated user modeling for personalized digital libraries. International Journal of Information Management, 2006. 26(3): p. 234-248.
- [6] Chiu, W. Web site personalization. 2001 [cited 2008 01 June]; Available from: http://www.ibm.com/developerworks/websphere/library /techarticles/hipods/personalize.html.
- [7] Kuo, Y.F. and L.S. Chen, Personalization technology application to Internet content provider. Expert Systems with Applications, 2001. 21(4): p. 203-215.
- [8] Liang, T.-P., et al., A semantic-expansion approach to personalized knowledge recommendation. Decision Support Systems, 2007. In Press, Corrected Proof.
- [9] Adomavicius, G. and A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. Knowledge and Data Engineering, IEEE Transactions on, 2005. 17(6): p. 734-749.

- [10] Linden, G., B. Smith, and J. York, Amazon. com recommendations: item-to-item collaborative filtering. IEEE Internet computing, 2003. 7(1): p. 76-80.
- [11] Das, A.S., et al., Google news personalization: scalable online collaborative filtering. Proceedings of the 16th international conference on World Wide Web, 2007: p. 271-280.
- [12] Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., Riedl, J.: GroupLens: applying collaborative filtering to Usenet news. Communications of the ACM 40 (1997) 77-87
- [13] Li, Y., L. Lu, and L. Xuefeng, A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-Commerce. Expert Systems With Applications, 2005. 28(1): p. 67-77.
- [14] Chedrawy, Z. and S.S.R. Abidi. Case based reasoning for information personalization: using a context-sensitive compositional case adaptation approach. in Engineering of Intelligent Systems, 2006 IEEE International Conference on. 2006: p. 1-6.
- [15] Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. ACM Transactions on Information Systems (TOIS) 2005, 23(1): p. 103-145
- [16] GAO, M., ZHONGFU, W., Incorporating pragmatic information in personalized recommendation systems. The 11th International Conference on Informatics and Semiotics in Organisations. (2009) 156-164.
- [17] GAO, M., ZHONGFU, W., LIU, K. Pragmatic Grid for personalized resource provision. Service Operations and Logistics, and Informatics, 2008. IEEE/SOLI 2008. IEEE International Conference on. (2008)
- [18] Lemire, D. and A. Maclachlan. Slope One Predictors for Online Rating-Based Collaborative Filtering. Society for Industrial Mathematics. 2005: p. 21-25.
- [19] Sarwar, B., Karypis, G., Konstan, J., Reidl, J., Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th international conference on World Wide Web, 2001: p. 285-295.
- [20] Kitts, B., D. Freed, and M. Vrieze, Cross-sell: a fast promotiontunable customer-item recommendation method based on conditionally independent probabilities. Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, 2000: p. 437-446.

- [21] Dey, A.K., Understanding and using context. Personal and Ubiquitous Computing, 2001. 5(1): p. 4-7.
- [22] Dey, A.K. and G.D. Abowd, Towards a better understanding of context and context-Awareness. CHI 2000 Workshop on the What, Who, Where, When, and How of Context-Awareness, 2000.
- [23] Dey, A.K., Abowd, G.D.: Towards a better understanding of context and context-Awareness. CHI 2000 Workshop on the What, Who, Where, When, and How of Context-Awareness (2000)
- [24] GAO, M., ZHONGFU, W., Kecheng, Liu., Personalisation in Web Computing and Informatics: Theories, Techniques, Applications, and Future Research. Information Systms Frontiers. DOI: 10.1007/s10796-009-9199-3.
- [25] GAO, M., ZHONGFU, W., Personalized Context-aware Collaborative Filtering based on Neural Network and Slope One. The 6th International Conference on Cooperative Design, Visualization and engineering (CDVE 2009). Lecture Notes in Computer Science, Springer, 2009 (Accepted).
- [26] ADOMAVICIUS, G., KWON, Y. O. New recommendation techniques for multicriteria rating systems. IEEE Intelligent Systems, (2007) 48-55.
- [27] ADOMAVICIUS, G., TUZHILIN, A. Multidimensional recommender systems: a data warehousing approach. Proceedings of the Second International Workshop on Electronic Commerce. Lecture Notes in Computer Science, (2001) 2232.

Min Gao is currently a Ph.D. candidate at Chongqing University, China. She received her B.Sc. degree in Computer Science from Qingdao Technological University in 2002, and M.Sc. degree in Computer Theory and Application from Chongqing University in 2005. She was an Academic Visitor in the Informatics Research Centre at the University of Reading, UK from 2007-2008. Her recent research is primarily in personalization and recommendation in web computing and informatics.

Zhongfu Wu is a senior professor, and Director of Network and Grid Research Institute, Chongqing University, China. He has been active in the research of e-learning, information systems and security, and high-performance computing for more than forty years and been broad known at research domain of Computer Science. His publication list contains more than 200 publications, articles and books. He has supervised about 40 candidates for PhD to completion.