User-Weight Model for Item-based Recommendation Systems

Min Gao, Yunqing Fu
College of Software Engineering, Chongqing University, Chongqing 400044, China
Email: {gaomin, yqfu}@cqu.edu.cn

Yixiong Chen
College of Computer Science, Chongqing University, Chongqing 400044, China
Email: chenyx@cqu.edu.cn

Feng Jiang
College of Civil Engineering, Chongqing University, Chongqing 400044, China
Construction Engineering Department, Chongqing Technology and Business Institute, Chongqing 400050, China
Email: jiangfeng@cqu.edu.cn

***Abstract*—Nowadays, item-based Collaborative Filtering (CF) has been widely used as an effective way to help people cope with information overload. It computes the item-item similarities/differentials and then selects the most similar items for prediction. A weakness of current typical item-based CF approaches is that all users have the same weight in computing the item relationships. In order to improve the recommendation quality, we incorporate users' weights based on a relationship model of users into item similarities and differentials computing. In this paper, a model of user relationship, a method for computing users' weights, and weight-based item-item similarities/differentials computing approaches are proposed for item-based CF recommendations. Finally, we experimentally evaluate our approach for recommendation and compare it to typical item-based CF approaches based on Adjusted Cosine and Slope One. The experiments show that our approaches can improve the recommendation results of them.**

***Index Terms*—personalized recommendation, collaboration filtering, item-based filtering, relationship model**

I. Introduction

Due to the explosive growth of the Web, recommendation systems have been widely accepted by users. Users offer feedback on purchased or consumed items, and recommendation systems use the information to predict their preferences for yet unseen items and subsequently recommend items with the highest predicted ratings for users [1, 2]. Personalized recommendation approaches have gained great momentum both in the commercial and research areas [3]. There have been several famous recommendation systems, such as Amazon [4, 5] and Netflix (http:// www.netflix.com/).

A problem of current item-based CF is that all users have the same weight when item-item similarities or differentials are computed. There is a common sense that some users' words are more important than others' in a social group. For item-based CF recommendation, that is, some users (and their ratings) will have higher weights than the others. In this paper a novel calculation approach is proposed to compute the weights of users to improve the recommendation results of typical CF approaches.

The contribution of the paper includes four points. Firstly, a novel user relationship graph model is presented. Secondly, an algorithm based on three rules is given for weighting users. Thirdly, we incorporate users' weights into computing item-item similarities and differentials. Fourthly, the weight-based approach is proved experimentally helpful to improve the recommendation results of item-based CF approaches.

In the next sections, the state of the art in item-based CF is reviewed first (Section 2). Then Section 3 proposes a model of users' relationship, a user weighting method, and weight-based item-item similarities/differentials computing approaches for item-based CF recommendations. And then we evaluate experimentally our approaches on a popular dataset MovieLens and compare them to typical item-based CF approaches (Section 4). At last, Section 5 draws conclusions.

II. Background and problem of item-based CF

CF is the most successful recommendation technique to date [6, 7]. In a typical CF scenario, there is a rating m*n matrix which includes a list of m users and a list of n items and lots of ratings. Items represented any kind of products. A rating $r_{u,i}$ means how the user u likes the item i. It is supposed that users mainly interested in high ratings. The key step of CF is to extrapolate unknown ratings [8].

This work was supported by the Fundamental Research Funds for the Central Universities (Project No.CDJZR10 18 00 23). This research is an extended version of the paper presented at ISECS 2011 [19].

doi:10.4304/jsw.7.9.2133-2140

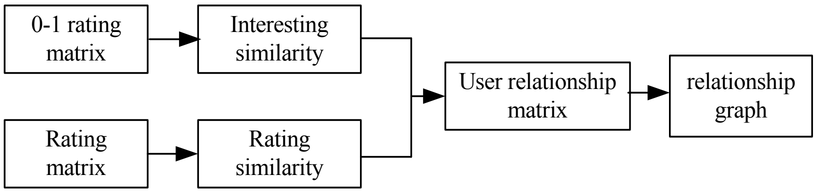

Fig. 2 User relationship model

The basic idea of traditional CF is to predict the rating of an item for a target user based on the opinions of other like-minded users. Item-based CF is the most popular approach in recommendation systems. It builds an item-item similarity/differential matrix for recommendations (See Fig. 1 [1]).

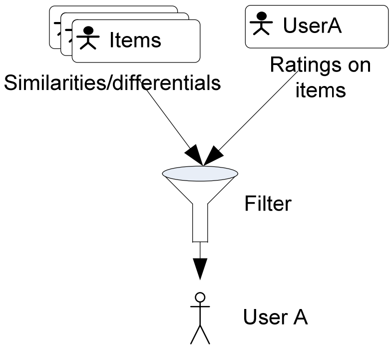

Fig. 1 Weighted Item-based Collaborative filtering

A. Item-based CF

Item-based CF, proposed by Sarwar et al. [7], is to compute the similarity between items and then to select the most similar items for prediction. Since it uses a pre-computed model, it recommends items quickly. There are several approaches to compute the similarities between items, such as Adjusted Cosine (See formula 1), and to compute the differences between items (See formula 3), e.g. Slope One [6].

$$\frac{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \overline{r_u}) \times (r_{u,j} - \overline{r_u})}{\sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \overline{r_u})^2} \sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,j} - \overline{r_u})^2}} \quad (1)$$

Here *U(i)* includes all users who have rated on item *i*. Formally, $U(i)=\{u|r_{u,i} \neq 0\}$. $\overline{r_u}$ is the average of user *u's* ratings. And also there are a number ways to estimate a rating, the most important step in a collaborative filtering system, such as weighted sum (See formula 2) and regression (See formula 4). Here *S(i)* includes all similar items of item *i*.

$$p_{u,i} = \frac{\sum_{j \in S(i)} sim(i,j) \times r_{u,j}}{\sum_{j \in S(i)} |sim(i,j)|} \quad (2)$$

The Slope One is a typical item-based collaborative filtering approach. It works on comparing the intuitive principle of popular differentials between items [6]. Note that it computes the differentials between items rather than similarities. The differential of item *i* and *j* $d_{i,j}$ is the average difference between item arrays of *i* and *j* (See formula 3). | | denotes the cardinality of a set.

$$d_{i,j} = \sum_{u \in U(i) \cap U(j)} \frac{(r_{u,i} - r_{u,j})}{|U(i) \cap U(j)|} \quad (3)$$

In turn, the deviations 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 4). Here $p_{u,j}$ is a prediction rating; $\overline{r_u}$ is the average of all known ratings of user *u*; and $\overline{d}$ is the average of all differentials $d_{i,j}$. $r_{u,i}$ - $d_{i,j}$ is the prediction for $r_{u,j}$ according to $r_{u,i}$.

$$p_{u,j} = \frac{\sum_{i \in r_u} (r_{u,i} - d_{i,j})}{|r_u|} = \overline{r_u} + \overline{d} \quad (4)$$

It has been proved that the Slope One scheme achieves accuracy comparable to the Adjusted Cosine and Pearson scheme. The Slope One has won the wide attention of researchers and companies due to its simplicity and efficiency [9, 10].

B. Weight Problem

The basis of the CF approaches is the relationships between items. Whether for computing similarities or differentials between items, the weights of all users are same in current item-based CF approaches. In other words, the weights of users are not taken into consideration in the current approaches. However, in any social group or network, some persons have high prestige because they have made great contribution to the group, and so on. So do the recommendation systems. If we take users' weights into consideration in item-base CF, the similarities of differentials between items will be more in line with the facts.

III METHODOLOGY FOR WEIGHTING USERS

In this section, we first propose a data model based on two kinds of users' relationship: interesting similarity and rating similarity, then present a weighting approach, and incorporate users' weights into item similarities/differentials computing approaches for further prediction.

A. Data Model for Weighting Users

There are various relationships between users in any social group; so are in a recommender system. Now we exploit this information for the calculation of users' weights.

Traditionally, the rating similarity is seen as user similarity; however, the rating similarity is only one aspect of the user similarity. There are some other relationships behind the ratings. For example, there are many items are rated by both user u and user v; the ratings are very different though. In this case, the rating similarity of them is very low. However, there should be a relationship between them that is high since the rated rating sets of them are similar. In the paper, this relationship is called interesting similarity. Rating similarity is the most used relationship between users in recommendation systems [7]. So in our research, the interesting similarity and rating similarity are taken into consideration in data model for users' relationship (See Fig.2).

Usually used calculation algorithms of rating similarity are Adjusted Cosine (See Section II.A) and Pearson Correlation. The rating similarity is the most used relationship in recommendation systems [7], therefore, it is not introduced in detail in the paper. The interesting similarity is analyzed as below.

According to [11], More the items have been rated by both user u_i and u_j, closer the relationship between the users is. This is the first rule of computing correlation relationships between users.

We define $I(u_i)$ as the set of items which have been rated by the user u_i. $I(u_i,u_j)$ is the set of items which have been rated by both u_i and u_j.

$$I(u_i,u_j)=\begin{cases}\{i_k:(r_{u_i,i_k}\neq\varnothing)\wedge(r_{u_j,i_k}\neq\varnothing)\} & if(u_i\neq u_j)\\ 0 & if(u_i=u_j)\end{cases}$$

Definition 3.1 *ISM*

ISM is a ($|U|\times|U|$) interesting similarity matrix which records the number of items which have been rated by each pair of users in a rating matrix.

$|U|$ denotes the cardinality of the set of users. *ISM* is formed by all the $I(u_i,u_j)$.

ISM is a symmetric matrix. All $I(u_i,u_j)$ is the same as $I(u_j,u_i)$. However, a fact is that if one of the pair has rated plenty of items, and another only has rated on small quantity of items, the correlation value should be different between them. This is the second rule. According to this rule, we normalize matrix *ISM* to correlation matrix *IM* by formula (6). Without loss of generality, supposed that $|I(u_i,u_j)|\neq 0$. $|I(u_j)|$ denotes the cardinality of the set $I(u_j)$.

Definition 3.2 *IM*

IM is a matrix that records the relationships of interesting similarity between users according to the number of items which they have rated and the numbers of items which are rated by other users.

$$IM_{u_i,u_j}=\frac{ISM_{u_i,u_j}}{\sum_{u_j\in U} ISM_{u_i,u_j}}=\frac{|I(u_i,u_j)|}{\sum_{u_j\in U}|I(u_i,u_j)|}$$

Note that, IM_{u_i,u_j} can be different from IM_{u_j,u_i}, *IM* is an unsymmetrical matrix.

The two relationship values form two matrixes: interesting similarity matrix (*IM* for short) and rating similarity matrix (*RSM* for short). In this paper, the average of the two kinds of values is used to express the users' correlation because they have similar dimensions. Therefore, the matrix of users' relationship is

$$URM=\frac{IM+RSM}{2}.$$

URM can be regarded as a weighted connective matrix for a correlation graph G. Nodes in G correspond to users in U and there will be a link (u_i,u_j) from u_i to u_j if $URM_{u_i,u_j}\neq 0$. The weight of the link is URM_{u_i,u_j}. The graph G is valuable model to further exploit correlation between users.

For *example*, given a rating matrix *RM* is as Table I. There are 5 users, 6 items, and several ratings in the matrix. A rating is marked from 1 to 5 that represents how a user prefers an item. $r_{u,i}$=Ø means the item i is not rated by the user u.

TABLE I
AN EXAMPLE OF RATING MATRIX

	i_1	i_2	i_3	i_4	i_5	i_6
u_1	4	Ø	5	3	3	Ø
u_2	3	0	4	0	5	4
u_3	Ø	4	Ø	2	Ø	Ø
u_4	4	3	3	4	5	3
u_5	Ø	Ø	3	Ø	4	Ø

Table II shows the *ISM* of the *RM*. Every element in the matrix is the number of items which have been rated by each pair of users (see formula 6). *ISM* is a symmetrical matrix. For each user pair (u_i,u_j), $ISM_{u_i,u_j}=ISM_{u_j,u_i}$, e.g., $ISM_{u_1,u_2}=ISM_{u_2,u_1}=3$

TABLE II
ISM OF THE RM

	u_1	u_2	u_3	u_4	u_5
u_1	0	3	1	4	2
u_2	3	0	0	4	2
u_3	1	0	0	2	0
u_4	4	4	2	0	2
u_5	2	2	0	2	0

Table III shows the correlation matrix *IM* of the *RM*. *IM* is an unsymmetrical matrix. For each user pair (u_i,u_j), where $i \neq j$, IM_{u_i,u_j} can be not equal to IM_{u_j,u_i}, e.g., IM_{u_1,u_2} =3/10=0.3, IM_{u_2,u_1} =3/9≅0.333, $IM_{u_1,u_2} \neq IM_{u_2,u_1}$. The sum of each row in the *IM* is 1, e.g.,

$$\sum_{u_j \in U} IM_{u_1,u_j}$$
$$= IM_{u_1,u_1} + IM_{u_1,u_2} + IM_{u_1,u_3} + IM_{u_1,u_4} + IM_{u_1,u_5}$$
=0.3+0.1+0.4+0.2 =1.

TABLE III
IM OF THE RM

	u_1	u_2	u_3	u_4	u_5
u_1	0	3/10	1/10	4/10	2/10
u_2	3/9	0	0	4/9	2/9
u_3	1/3	0	0	2/3	0
u_4	4/12	4/12	2/12	Ø	2/12
u_5	2/6	2/6	0	2/6	0

Then the formula

$$Sim_{u_1,u_2} = \frac{\sum_{i \in I(u_1) \cap I(u_2)} (r_{u_1,i} - \overline{r_{u_1}}) \times (r_{u_2,i} - \overline{r_{u_2}})}{\sqrt{\sum_{i \in I(u_1) \cap I(u_2)} (r_{u_1,i} - \overline{r_{u_1}})^2} \sqrt{\sum_{i \in I(u_1) \cap I(u_2)} (r_{u_2,i} - \overline{r_{u_2}})^2}}$$

is used to compute the rating similarity matrix *RSM* of the *RM*.

B. Weighting Algorithm

The purpose of the algorithm is to forecast users' weights. It is important to properly control the weight flow in order to transfer high score to the users that are strongly related to users with high weights.

The spreading algorithm follows three rules. Firstly, if a user u_i is linked by high ranked users with high weights, then u_i will also have high weight. Secondly, users have to transfer their positive influence through the graph, but this effect decreases its power if it spreads further and further away. Thirdly, if the user u_i is connected to two or more nodes, these nodes share the boosting effect according to the weights of the connections as computed in *URM*.

The rules are just similar to the propagation and attenuation of *PageRank*. We can compute users' weights in a very efficient way based on the method for *PageRank* computing [12-14].

The *PageRank* score for the node *n* is defined as formula (5), where *O(q)* is the out-degree of the node *q*, α is a decay factor, a common choice of α is 0.85 [15].

$$PR(n) = (1-\alpha) \cdot \frac{1}{|V|} + \alpha \cdot \sum_{q:(q,n) \in E} \frac{PR(q)}{O(q)} \tag{5}$$

So the weight value for a node u_k can be computed by Formula (6).

C. Computing Methods for Item Similarities and Differentials

As can be see from Fig. 3, we combine users' weights and the procedures of item similarities and differentials algorithms, such as Adjusted Cosine and Slope One (See Formulas 7 and 8). Then we apply formula (2) and (4) to predict ratings.

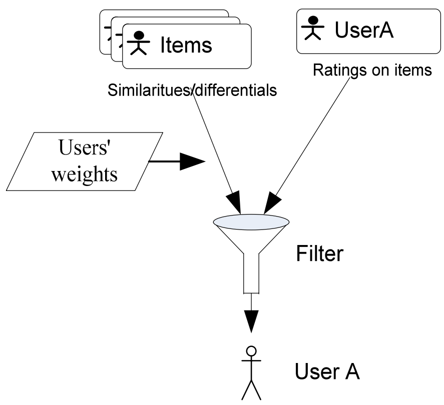

Fig. 3 Weighted Item-based Collaborative filtering

$$w_{u_n} = (1-\alpha) \cdot \frac{1}{|V|_w} \cdot w_{u_n} + \alpha \cdot \sum_{u_k:(u_k,u_n) \in E} \frac{w_{u_k} \times URM_{u_k,u_n}}{\sum_{u_m:(u_k,u_m) \in E} URM_{u_k,u_m}} \tag{6}$$

$$Sim_{i,j} = \frac{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \overline{r_u}) \times (r_{u,j} - \overline{r_u}) \times w_u^{\ 2}}{\sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - \overline{r_u})^2 \times w_u^{\ 2}} \sqrt{\sum_{u \in U(i) \cap U(j)} (r_{u,j} - \overline{r_u})^2 \times w_u^{\ 2}}} \tag{7}$$

$$d_{i,j} = \frac{\sum_{u \in U(i) \cap U(j)} (r_{u,i} - r_{u,j}) \times w_u}{\sum_{u \in U(i) \cap U(j)} w_u} \tag{8}$$

IV. Experimental evaluation

A. Data Set

In the experiments, we used MovieLens dataset from the well-known MovieLens project (http://MovieLens.umn.edu) to evaluate our approach. MovieLens is a widely used benchmark to evaluate scoring algorithms applied to recommender systems. The data consists of 100,000 ratings (1-5) from 943 users on 1682 movies. Each user has rated at least 20 movies. The data set was divided into training set (80% of the data) and test set (20% of the data) five times.

Each training and test sets are named *Unbase* and *Untest* (n=1,…,5). Every *Unbase* was used to compute user ranks, and the ratings in each *Untest* were the target ones to predict.

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 precision of our algorithm and the real ratings, and then compare it with Adjusted Cosine and Slope One algorithms. For all predictions $\{p_1, p_2, \ldots, p_n\}$ and their real ratings $\{r_1, r_2, \ldots, r_n\}$. MAE is the average of absolute error between all $\{p_i, r_i\}$ pairs (See formula 8 [16]). The lower the MAE, the more accurately the predictions are, and the better the recommendation approach is.

$$MAE = \frac{\sum_{i=1}^{N} | p_i - r_i |}{N} \tag{9}$$

The precision is the percentage of truly "high" ratings (B) among those (A) that were predicted to be "high" by a recommender system (See formula 10). That is the number of correct results divided by the number of all returned results.

$$\frac{A \cap B}{A} \tag{10}$$

And the recall is the percentage of correctly predicted "high" ratings among all the ratings known to be "high" (see formula 11). That is the number of correct results divided by the number of results that should have been returned.

$$\frac{A \cap B}{B} \tag{11}$$

F-measure is a measure of a test's accuracy. It considers both the precision and the recall of the test to compute the score (See formula 12). The parameter β has regular certain values of 0.5,1, and 2.

$$F_\beta = \frac{(1+\beta^2) \cdot (precision \cdot recall)}{\beta^2 \cdot precision + recall} \tag{12}$$

C. Procedure and Results

(A). Comparison of prediction results

To compare weight-based algorithms (weight-based Adjusted Cosine and weight-based Slope One) with typical algorithms (Adjusted Cosine and Slope One), we performed the experiment where we computed MAE, precision, recall, and f-measure ($F_{0.5}$, F_1, and F_2) for all of them. Our results are shown in Fig.4 and Fig.5. The blue columns (the left columns) are the metric values for one of weight-based algorithms; the purple ones (the right columns) are for one of typical algorithms. It can be observed from the charts that our weight-based algorithms out performs typical algorithms at prediction error (MAE) and recommendation accuracy (F-measure).

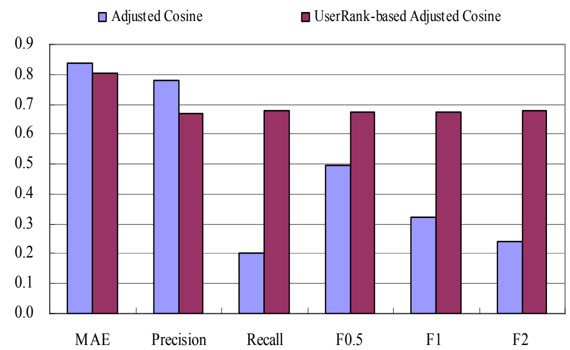

Fig.4 The experiment results of Adjusted Cosine and weight-based Adjusted Cosine recommendation algorithms

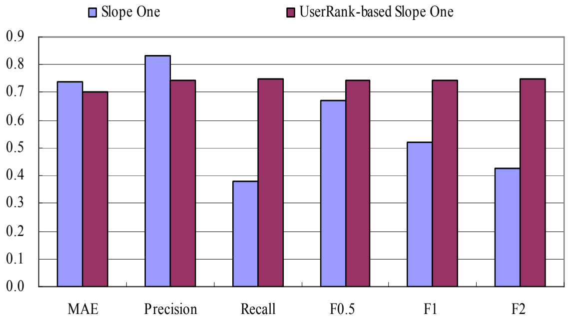

Fig.5 The experiment results of Slope One and weight-based Slope One recommendation algorithms

- ✧ Table IV shows the MAE values of the algorithms. The lower the MAE, the more accurately the predictions are.

TABLE IV
THE MAE OF EXPERIMENTS

Algorithms	MAE
Adjusted Cosine	0.837
Weight-based Adjusted Cosine	0. 806
Slope One	0. 740
Weight-based Slope One	0. 702

- ✧ Although the values of precision of the weight-based algorithms are somewhat lower than the traditional algorithms, the results are high significant for the recall of the weight-based algorithms. So that the F-measure values are higher than the typical algorithms, that means the accuracy of the algorithms is improved. The F-measure and recall results are shown in Table V.

TABLE V
RESULTS OF F-MEASURE AND RECALL

	Adjusted Cosine	Weight-based Adjusted Cosine	Slope One	Weight-based Slope One
$F_{0.5}$	49.7%	67.2%	67.3%	74.4%
F_1	32.2%	67.4%	52.2%	74.5%
F_2	23.9%	67.7%	42.6%	74.7%
Recall	20.3%	67.9%	38%	74.8%

(B). Comparison of the prediction ability for user relevant items

To compare the prediction ability for user relevant items of the algorithms, we performed the experiment where we computed MAE results for the user relevant items in the test set. The user relevant items are those items with 4 or 5 score rated by the user. The table VI shows the MAE results. As can be seen from the table, the prediction abilities of weight-based algorithms are better than the traditional algorithms.

TABLE VI
MAE RESULTS OF PREDITIONS FOR USERS' RELEVENT ITEMS

Algorithms	MAE
Adjusted Cosine	0.75
Weight-based Adjusted Cosine	0.71
Slope One	0.66
Weight-based Slope One	0.62

(C). Comparisons of algorithm stability

To compare the stability of the algorithms, we then performed the experiment where we computed MAE, precision, recall, and f-measure of the prediction results for the ratings more than 3 (R3 for short) and more than 4 (R4 for short) in the test set respectively. Fig. 6 to Fig. 9 shows the deviations of the MAE, precision, recall, and f-measure results of Adjusted Cosine, Weight-based Adjusted Cosine, Slope One, and Weight-based Slope One respectively. The metric values of the prediction for the items in R3 and the items in R4 are shown by the left columns ■ and the right columns ■ in the figures. As can be seen from the figures, the prediction stability of weight-based algorithms is better than the typical algorithms.

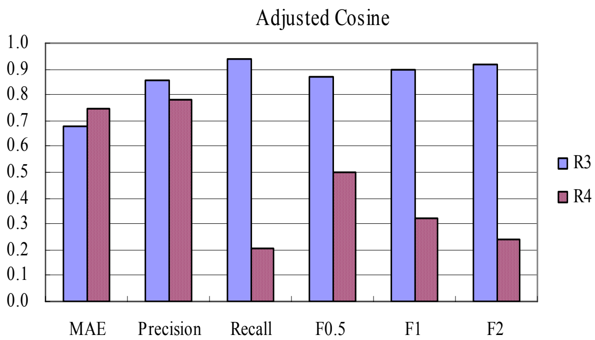

Fig. 6 The results of Adjusted Cosine for R3 and R4

- ✧ As can be seen from Fig.5, all matric values of the predictions for R3 are better than the predictions for R4. In these metric values, the change of recall is most significant, from 93.6% to 20.3%; secondly, the change of f-measure is great as well, the values of F2, F1, and F0.5 vary from 91.9%, 89.5%, and 87.1% to 23.9%, 32.2%, and 49.7%; the changements of MAE and precision are the least, from 0.676 and 85.5% to 0.745 and 77.8%.

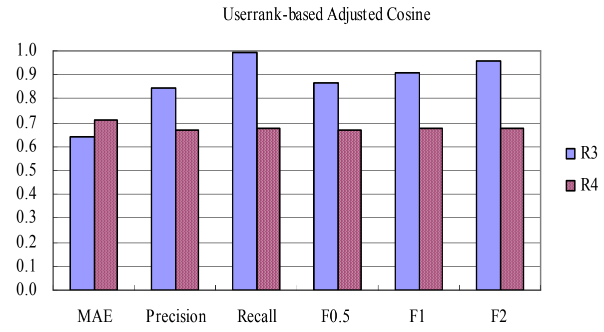

Fig. 7 The results of weight-based Adjusted Cosine for R3 and R4

- ✧ As can be seen from Fig.7, the changement trend of all matric values of weight-based Adjust Cosine is similar with Adjust Cosine (Fig.4), but the changement rangeability is lower. The maximum of the change in Fig.5 is only more than 30%, but it is more than 70% in Fig.4.
- ✧ The Fig.8 and Fig.9 are similar with Fig.6 and Fig.7. Compared with weight-based Adjusted Cosine, the changement rangeability of weight-based Slope One are lower.

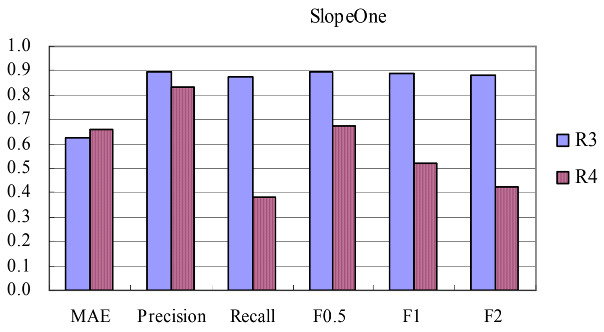

Fig. 8 The results of Slope One for R3 and R4

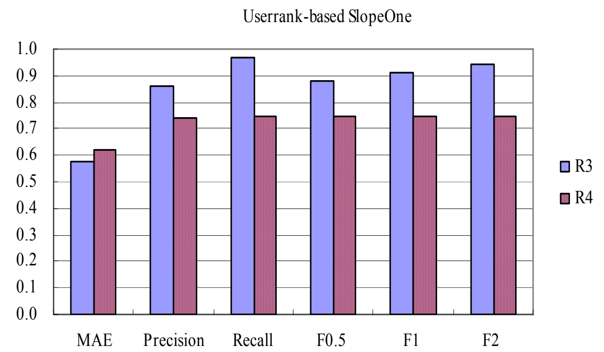

Fig. 9 The results of weight-based Slope One for R3 and R4

In summary, 1) except MAE values, all the other metrics including prediction precision, prediction accuracy, prediction ability for user relevant items, and the stability of the weight-based algorithms are better than the typical algorithms, and 2) the weight-based Slope One are better then user-based Adjusted Cosine at these metrics.

V. Conclusion and Future Work

The Recommendation systems help users find items they would be interested in. Currently, item-based collaborative filtering approaches are most popular in recommender systems. The typical Adjusted Cosine and Slope One are well-known algorithms of them. In this paper we analyzed how to weight users predict ratings for items based on user ranks. Experimental results show that the information of users' weights is helpful to improve the prediction results, the prediction ability for user relevant items, and the stability of typical algorithms.

Acknowledgment

We gratefully acknowledge valuable input from Tao Zhou and Peidong Li, MSc students, from the School of Computer Science, Chongqing University.

References

[1] M. Gao, K. Liu, and Z. Wu, "Personalisation in web computing and informatics: theories, techniques, applications, and future research," *Information Systems Frontiers. DOI:10.1007/s10796-009-9199-3,* 2009.

[2] F. Jiang and M. Gao, "Collaborative filtering approach based on item and personalized contextual information," in *Proceedings of the International Symposium on Intelligent Information Systems and Applications* Qingdao, P. R. China, 2009, pp. 63-66.

[3] M. Eirinaki and M. Vazirgiannis, "Web mining for web personalization," *ACM Transactions on Internet Technology,* vol. 3, pp. 1-27, 2003.

[4] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet computing,* vol. 7, pp. 76-80, 2003.

[5] A. S. Das, M. Datar, A. Garg, and S. Rajaram, "Google news personalization: scalable online collaborative filtering," *Proceedings of the 16th international conference on World Wide Web,* pp. 271-280, 2007.

[6] D. Lemire and A. Maclachlan, "Slope One Predictors for Online Rating-Based Collaborative Filtering," 2005.

[7] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web* Hongkong, 2001, pp. 285-295.

[8] M. Gao and Z. Wu, "Personalized context-aware collaborative filtering based on neural network and slope one," *Lecture Notes in Computer Science,* vol. 5738, pp. 109-116, 2009.

[9] G. Louppe, "Collaborative filtering: Scalable approaches using restricted Boltzmann machines," in *Department of Electrical Engineering and Computer Science*. vol. Master: University of Li`ege, 2010.

[10] D. Jannach and K. Hegelich, "A case study on the effectiveness of recommendations in the mobile internet," in *Proceedings of the third ACM conference on Recommender systems (RecSys '09)*, 2009, pp. 205-208.

[11] F. Zhang, "Research on trust based collaborative filtering algorithm for user's multiple interests," *Journal of Chinese Computer Systems,* vol. 29, pp. 1415-1419, 2008.

[12] A. N. Langville and C. D. Meyer, "Deeper inside pagerank," *Internet Mathematics,* vol. 1, pp. 335-380, 2004.

[13] T. Haveliwala, "Efficient computation of PageRank," 1999.

[14] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub, "Extrapolation methods for accelerating PageRank computations," 2003, pp. 261-270.

[15] M. Gori, A. Pucci, V. Roma, and I. Siena, "Itemrank: A random-walk based scoring algorithm for recommender engines," in *Proceedings of the 20th International Joint*

Conference on Artificial Intelligence (IJCAI) Hyderabad, India, 2007, pp. 778-781.

[16] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems,* vol. 22, pp. 5-53, 2004.

[17] M., Gao. Z. Wu, and F. Jiang, "Userrank for Item-based Collaborative Filtering Recommendation". *Information Processing Letters*. DOI: 10.1016/j.ipl.2011.02.003, 2011(111) 440-446

[18] M. Gao and Z. Wu, "Incorporating Personalized Contextual Information in Item-based Collaborative Filtering Recommendation". *Journal of Software*, 2010. 5（7）29-736.

[19] F. Jiang and Z. Wang, "A Novel Weighting Model for Recommendation systems," in ISECS 2011 Shanghai, China, 2011 (In press).