International Journal of Innovative
Computing, Information and Control
Volume 11, Number 5, October 2015

ICIC International ©2015 ISSN 1349-4198
pp. 1629–1638

A SINGULARITY-BASED USER SIMILARITY MEASURE FOR RECOMMENDER SYSTEMS

Shunpan Liang[1], Lin Ma[1] and Fuyong Yuan[2,*]

[1]Institute of Information Science and Engineering
[2]Computer Teaching Experiment Center of Institute of Information Science and Engineering
Yanshan University
No. 438, West Hebei Avenue, Qinhuangdao 066004, P. R. China
liangshunpan@ysu.edu.cn; a_lin_ysu@163.com; *Corresponding author: fyyuan@ysu.edu.cn

Received March 2015; revised July 2015

Abstract. *Collaborative filtering is one of the most widely used methods in personalized recommender systems. The most critical part of collaborative filtering is to compute similarities among users using a user-item rating matrix based on which recommendations can be generated. The traditional methods used to calculate user similarity include Pearson correlation coefficient (PCC) and Jaccard Index. However, since PCC defines user similarity as the linear correlation and Jaccard Index defines user similarity as the proportion of common ratings, the accuracy is not ideal if we use these approaches directly. In this paper, we propose a singularity-based similarity measure to resolve this issue. Specifically, we first improve PCC by incorporating the number of common items rated by two users. We then come up with an improved Jaccard method by considering the rating values issued on the items. Finally, we combine the two improved approaches together in two different ways, aiming to further improve recommendation accuracy. Experimental results on two real-world data sets show that our method achieves superior accuracy.*

Keywords: Recommender systems, Collaborative filtering, Singularity, Pearson correlation coefficient, Jaccard, Accuracy

1. **Introduction.** Collaborative filtering (CF) has become one of the most widely used recommender methods to recommend items to users and it can be divided into two methods: memory-based approach [3] and model-based approach [8]. The hypothesis of CF is that if users had similar preference to some items in the past, they will have similar opinions towards other items. The process of CF includes three steps: searching neighborhoods of target user, then predicting ratings of target user to items according to ratings of neighborhoods and finally producing recommendation list. In order to find the neighborhoods of target user, we must measure the similarity between users and choose some users with the highest similarity as the nearest neighbors of target user. Therefore, similarity is important. The Pearson correlation coefficient (PCC) [15] and cosine similarity (COS) [4] are the most widely used similarity measures in CF.

However, there are some shortages existing in both PCC and COS. They can be summarized in four specific cases [5].

a) If the number of commonly rated items is 1, PCC cannot be calculated since the denominator part of the correlation formula becomes 0 and COS is always 1 regardless of differences in individual ratings.

b) If two users have only two commonly rated items, two situations will be resulted in about PCC. When the rating vectors are cross like $(4, 2)$ and $(1, 3)$, PCC is always -1; otherwise, PCC is 1 if computable, for example, $(4, 3)$ and $(3, 1)$.

c) If all the available ratings are flat, e.g., $(1, 1, 1)$, PCC cannot be calculated and COS results in 1 no matter what the rating values.

d) If two users show entirely opposite ratings on the commonly rated items, e.g., $(1, 5, 1)$ and $(5, 1, 5)$, PCC results in -1 all the time.

Besides the shortages mentioned above, both PCC and COS do not consider the proportion of commonly rated items. Considering that a, b and c have rated 6, 9 and 150 items respectively, a and b have 5 commonly rated items, while c and b also have 5 commonly rated items. Obviously, the similarity between b and a is more credible than the similarity between b and c if the similarity of the two pairs is the same. Therefore, the proportion of commonly rated items is important. Based on this hypothesis, we consider Jaccard method [6]. It is defined by the ratio of the number of intersection and union of the items rated by two users. However, the traditional Jaccard does not take rating values into consideration, which may lead to low precision in similarity calculation.

This paper presents a singularity-based algorithm to improve the accuracy of similarity between users. The basis of our hypothesis is that a very singular similarity should be awarded a higher value than a very normal similarity. For example, if only two users have voted positively while the rest have voted negatively for one item, this represents a very great singularity which should be considered as a very great similarity for this item. Since it is reported that PCC works better than COS in CF [13], we adopt PCC to improve in order to achieve a better result. Besides, PCC defines user similarity as the linear correlation and it does not consider the proportion of common ratings, so Jaccard is considered in our new method. Accordingly, the method proposed in our paper is considered from two aspects. First, considering the ratio of the number that item rated, we propose an improved PCC. Second, based on the proportion of common rating items, the improved Jaccard is put forward. At last, the two parts are combined in two different ways. In order to test and verify the new similarity measure, experiments are implemented on two most used real data sets, and the results show that our method is efficient and accurate.

The rest of the paper is organized as follows. Section 2 gives a brief overview of related research on how to improve predict accuracy of CF and motivate our present work. The proposed approach is then elaborated in Section 3 where we also highlight the advantages of our method in principle. Experiments on two real-world data sets are conducted in Section 4 to verify the effectiveness of our method in predicting items' ratings. Finally, Section 5 concludes our work.

2. **Related Work.** Collaborative filtering (CF), as a kind of the most successful technique in personalized recommendation, has been widely used in many domains. However, CF also suffers from a few of issues, for instance, cold start problem, and data sparsity. These problems seriously reduce the quality of recommendation. This paper aims at improving the prediction accuracy. Up to now, many researchers have focused on the prediction and proposed some solutions.

The researches can be classified into two categories broadly. First, some researchers attempt to modify one method in some way. Ortega et al. proposed a method using Pareto dominance [2] to perform a pre-filtering process eliminating less representative users from the k-neighbor selection process while retaining the most promising ones. Nevertheless, this measure will be effective when the value of k is large. Patra et al. presented a similarity measure for neighborhood based CF [19], which found importance of each pair of rated items by exploiting Bhattacharyya similarity. Guo et al. taking into consideration both the direction and length of rating vectors proposed a novel Bayesian similarity measure based on the Dirichlet distribution [5]. It overcomes the shortages of traditional similarity measure and achieves good results. Lathia et al. developed a concordance-based measure [14], which estimates the correlation based on the number of concordant, discordant and tied pairs of common ratings. It considers users' privacy. However, since it depends on the mean of ratings to determine the concordance, this approach also suffers from sparsity problem. Ahn proposed a new similarity for collaborative filtering that is called PIP [7]. It calculates similarity from three aspects: proximity, impact and popularity. However, this measure considers only the local information of the ratings and does not consider the global preference of user ratings. Besides, the calculation method is too complex and will produce non-bounded numerical value, which is less meaningful in user correlation. Bobadilla et al. proposed a significance based similarity measure (S-Measure) [9]. It is used to improve prediction/recommendation quality by weighting the ratings of the items according to their importance. This measure first calculates the significance of an item, the significance of a user and the significance of an item for a user. Then the traditional PCC or COS will be used to calculate the similarities according to significance. Recently, Bobadilla et al. introduced a singularity based similarity measure (SM) [10]. This paper hypothesized that the results obtained by applying traditional similarity measures could be improved by taking contextual information. It categorized the ratings as positive and non-positive firstly. Then it computed the singularity values of each item and was combined with mean squared difference. However, the accuracy may be not exact when the number of items rated by users is small. Our work is also based on the idea of singularity, but it is different from SM, and the specific method will be introduced in Section 3.

Second, other researchers try to combine two or three kinds of methods to get a better one. Bobadilla et al. proposed a new metric that combined Jaccard measure and mean squared difference, which is called JMSD [12]. It assumed that the two measures could complete each other and increased the quality of recommendation. Another new method, using optimization based on neural learning, which is called MJD (Mean-Jaccard-Difference) [11], was also presented by Bobadilla et al. It obtained important improvement when applied to new user cold start situations. Choi and Suh proposed a new similarity function in order to select different neighbors for each different target item [16]. In this way, the rating of a user on an item is weighted by the item similarity between the item and the target item. They combined three different kinds of methods to calculate the item and user similarity respectively, which are Pearson, Cosine and Distance. Although the experimental results displayed that the new method is better than the traditional one, it was only a combination of traditional methods. In addition, some researchers combine trust into CF to improve the performance of recommendation. For example, Guo et al. presented a merged method called Merge$x$ [17], which merged the ratings of trusted neighbors in order to form a new and more complete rating profile for the active users based on which recommendations can be generated by integrating similarity and trust into CF. Deng et al. proposed a social network based service recommendation method with trust enhancement known as RelevantTrustWalker [18]. First, a matrix factorization method is utilized to assess the degree of trust between users in social network. Next, an extended random walk algorithm is proposed to obtain recommendation results. However, trust networks are binary and there are no specific trust ratings in real datasets.

Our work focuses on a singularity-based algorithm to improve the accuracy of similarity between users. The approach we present is different from the above mentioned metrics which are combined directly by traditional methods. In order to achieve a better result, we first propose an improved PCC by considering the ratio of the number that item rated. Then we take Jaccard into consideration as the proportion of common ratings is also important, and we improve the traditional Jaccard method further. At last, the two parts are combined in two different ways, aiming to further enhance recommendation accuracy.

3. The Singularity-based Similarity Measure.

This section introduces the specific method. PCC defines user similarity as the linear correlation and it works better than COS [13] in CF recommender systems. However, because of various problems, such as data sparsity, it will lead to inaccurate prediction in practical application. Accordingly, based on the idea of singularity, we improve PCC by considering the ratio of the number that item rated firstly. On the other hand, the proportion of common ratings is also important and it will affect the quality of recommendation. Thus, Jaccard is considered next. We calculate positive singularity, negative singularity and empty singularity for each item to improve Jaccard. In the end, we combine the two methods in two different ways in order to achieve a better effect.

3.1. The improved Pearson correlation coefficient method.

In a recommender system, when two users both rate the items which are rated by only a few users, the similarity between the two can be regarded as showing stronger evidence compared with when they rate the items that are rated by the majority of users. For example, a high-budget film premiere or the latest product of Apple (non-singularity item) will be considered much more popular than a little-known older film or a non-intelligent electronic product (singularity item), and a large number of users will watch the popular film or buy the new Apple product, but this behavior cannot reflect that the users are similar. Conversely, if two users both watch the religious film (singularity item), we know they have the same preference and their similarity is high. So the more singularity the item is, the higher the similarity is.

The traditional PCC is defined as (1):

$$sim(x,y)^{PCC} = \frac{\sum\limits_i (r_{x,i} - \overline{r_x})(r_{y,i} - \overline{r_y})}{\sqrt{\sum\limits_i (r_{x,i} - \overline{r_x})^2} \cdot \sqrt{\sum\limits_i (r_{y,i} - \overline{r_y})^2}} \tag{1}$$

where i represents the set of commonly rated items by users x and y. $\overline{r_x}$ and $\overline{r_y}$ are the average rating value of users x and y respectively. $r_{x,i}$ and $r_{y,i}$ denote the rating of item by users x and y respectively.

We define U as the set of the users and I as the set of the items. Let $|I|$ be the number of items and $|U|$ be the number of users. The similarity using the improved Pearson correlation coefficient (IP) which is based on singularity between users x and y is defined as (2)-(4):

$$sim(x,y)^{IP} = \frac{\sum\limits_i \left(r'_{x,i} - \overline{r'_x}\right)\left(r'_{y,i} - \overline{r'_y}\right)}{\sqrt{\sum\limits_i \left(r'_{x,i} - \overline{r'_x}\right)} \cdot \sqrt{\sum\limits_i \left(r'_{y,i} - \overline{r'_y}\right)}} \tag{2}$$

$$r'_{x,i} = \frac{|U_i|}{|U|} \cdot r_{x,i} \tag{3}$$

$$\overline{r'_x} = \frac{\sum\limits_i r'_x}{|i|} \tag{4}$$

where i represents the set of common rating items of users x and y, $|i|$ is the number of i, the calculation methods of x and y are the same and we do not repeat here. As we know, the real data set is comparatively sparse and $\frac{|U_i|}{|U|}$ may be too small to affect the result. Therefore, in the experiments of Section 4 we define $r'_{x,i}$ as (5) and $r'_{y,i}$ likewise.

$$r'_{x,i} = \begin{cases} \frac{|U_i|}{H} \cdot r_{x,i}, & |U_i| \leq H \\ r_{x,i}, & \text{otherwise} \end{cases} \tag{5}$$

where H is an experimental value and it is set 100 in our experiment.

TABLE 1. Ratings of 5 users to 8 items

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
U_1			*5*					*3*
U_2	*2*	*5*					*3*	
U_3			*4*					*4*
$\boldsymbol{U_4}$		***1***	***3***	***2***	***5***			***4***
$\boldsymbol{U_5}$	***4***	***5***		***4***	***5***	***5***		***5***

Use Table 1 as an example. Table 1 represents a hypothetical portion of a recommender system, where possible votes are in the range of 1-5. We are trying to find the similarity between U_4 and U_5, which are in bold. The set of common rating items of U_4 and U_5 is (I_2, I_4, I_5, I_8) and the ratings on the set are $(1, 2, 5, 4)$ and $(5, 4, 5, 5)$ respectively. Then we can get the PCC similarity of U_4 and U_5 by Equation (1). It is approximately equal to 0.3651. The similarity is small, which is because that the ratings in common are not very similar. Next, we calculate the similarity by IP method. The number of each item rated is 3, 2, 2, 4 respectively. Considering the number of each item rated, the ratings of U_4 and U_5 are $(0.6, 0.8, 2, 3.2)$ and $(3, 1.6, 2, 4)$ respectively. Then we can get the similarity between U_4 and U_5 using Equation (2), and the result is approximately 0.6152. The IP similarity is larger than the traditional PCC similarity, which is because that IP method considers the number of each item rated while PCC measure does not. For the commonly rated items of two users, the more singularity the items are, the greater the similarity between the two users is.

3.2. **The improved Jaccard method.** Although the IP measure can get better result than PCC, the number of common ratings is not taken into account. However, the proportion of common ratings is also important. Therefore, we adopt the idea of Jaccard measure. It is defined as (6):

$$sim(x, y)^{Jaccard} = \frac{|I_x| \cap |I_y|}{|I_x| \cup |I_y|} \tag{6}$$

where $|I_x|$ and $|I_y|$ represent the number of items rated by users x and y respectively.

However, Jaccard measure does not consider the value of ratings. Accordingly, based on the idea of singularity we proposed the improved Jaccard (IJ), which presents positive singularity, negative singularity and empty singularity. Next, we will introduce the three kinds of singularities specifically. In a recommender system, where possible ratings are in the range of 1-5, 1-3 indicates negative evaluations and 4-5 indicates positive evaluations.

For an item, the ratings of users x and y can be divided into three situations.

a) The ratings are agreement. They are positive or negative, which is defined as PA or NA.

b) The ratings are disagreement. One is positive and the other is negative. This condition is defined as D.

c) One user rates and the other one does not. If the user gives positive rating, it is defined as PO. Conversely, if the rating is negative, it is defined as NO.

We define P_i as the set of users who have assigned to the item i a positive value. Use the example in Table 1:

$$P_1 = \{U_5\}, P_2 = \{U_2, U_5\}, P_3 = \{U_2, U_3\}, P_4 = \{U_5\}, P_5 = \{U_4, U_5\},$$
$$P_6 = \{U_5\}, P_7 = \emptyset, P_8 = \{U_3, U_{4,} U_5\}$$

We define N_i as the set of users who have assigned to the item i a negative value. Use the example in Table 1:

$$N_1 = \{U_2\}, N_2 = \{U_4\}, N_3 = \{U_4\}, N_4 = \{U_4\}, N_5 = \emptyset, N_6 = \emptyset, N_7 = \{U_2\}, N_8 = \{U_1\}$$

We define E_i as the set of users who have not assigned to the item i. Use the example in Table 1:

$$E_1 = \{U_1, U_3, U_4\}, E_2 = \{U_1, U_3\}, E_3 = \{U_2, U_5\}, E_4 = E_5 = \{U_1, U_2, U_3\},$$
$$E_6 = \{U_1, U_2, U_3, U_4\}, E_7 = \{U_1, U_3, U_4, U_5\}, E_8 = \{U_2\}$$

Definition 3.1. *Positive singularity. The more users who have rated the item i with positive ratings, the lower the singularity related to this item will be. The positive singularity is denoted by S_P^i and defined as (7):*

$$S_P^i = 1 - \frac{|P_i|}{|U|} \tag{7}$$

where $|P_i|$ is the number of the set P_i.

Definition 3.2. *Negative singularity. The more users who have rated the item i with negative ratings, the lower the singularity related to this item will be. The negative singularity is denoted by S_N^i and defined as (8):*

$$S_N^i = 1 - \frac{|N_i|}{|U|} \tag{8}$$

where $|N_i|$ is the number of the set N_i.

Definition 3.3. *Empty singularity. The more users who have not rated the item i, the lower the singularity related to this item will be. The empty singularity is denoted by S_E^i and defined as (9):*

$$S_E^i = 1 - \frac{|E_i|}{|U|} \tag{9}$$

where $|E_i|$ is the number of the set E_i.

With the above explanation, the improved Jaccard is defined as (10):

$$sim(x, y)^{IJ} = \frac{\sum\limits_{i \in PA} S_P^i + \sum\limits_{i \in NA} S_N^i + \sum\limits_{i \in D} \sqrt{S_P^i \cdot S_N^i}}{\sum\limits_{i \in PA} S_P^i + \sum\limits_{i \in NA} S_N^i + \sum\limits_{i \in D} \sqrt{S_P^i \cdot S_N^i} + \sum\limits_{i \in PO} \sqrt{S_P^i \cdot S_E^i} + \sum\limits_{i \in NO} \sqrt{S_N^i \cdot S_E^i}} \tag{10}$$

where $|PA| + |NA| + |D| = ||I_x| \cap |I_y||$, and $|PA| + |NA| + |D| + |PO| + |NO| = ||I_x| \cup |I_y||$.

The improved Jaccard is unique. Jaccard has never been improved ever before and it is used by combining with other methods merely. The Jaccard similarity between U_4 and U_5 is approximately 0.5714, while the IJ similarity is approximately 0.6645, which is closer to the IP similarity.

3.3. **The final method.** Since IP does not consider the number of common ratings and IJ does not consider the rating values, we combine the IPCC and IJ in two different ways to get the final method. One way is a simple multiplication of the two methods, and it shows that the two methods are equally important. This method is expressed as IPIJ. Another way is a linear combination of the two, and it can highlight which method is more significant. This method is named IPAIJ. They are defined as (11) and (12):

$$sim(x,y)^{IPIJ} = sim(x,y)^{IP} \cdot sim(x,y)^{IJ} \tag{11}$$

$$sim(x,y)^{IPAIJ} = \partial \cdot sim(x,y)^{IP} + (1-\partial) \cdot sim(x,y)^{IJ} \tag{12}$$

The effects of these methods are validated in the fourth section.

4. **Experiments.** This section explains the experimental design for evaluating our ideas proposed in Section 3 as well as how the ideas affect the quality of recommendation.

4.1. **Experimental design.** In order to evaluate the effectiveness of our approach, two real-world data sets are used in our experiments. One is MovieLens 100K(ML-100K) which includes 100,000 ratings with 943 users and 1682 items. The other is MovieLens 1M(ML-1M) and there are 6040 users and 3952 items with 1,000,209 ratings. In both data sets, each person has rated at least 20 items. The user profile includes age, sex and profession. The items include 19 types of movies. The density of the user-item matrix is 6.3% in ML-100K and 4.1% in ML-1M. The two data sets are the most commonly used.

We evaluate recommendation performance using the 5-fold cross validation method. The data set is split into five disjoint sets; for each iteration, four folds are used as training set and one as testing set. We apply the KNN approach to selecting a group of similar users whose ranking is in the top KNN according to similarity; we vary KNN from 5 to 50 with step 5. The ratings of selected similar users are aggregated to predict items' ratings by a mean-centering approach [1]. Accuracy is measured by mean absolute error (MAE) and root mean square error (RMSE). The lower MAE and RMSE indicate better accuracy. They are defined as follows:

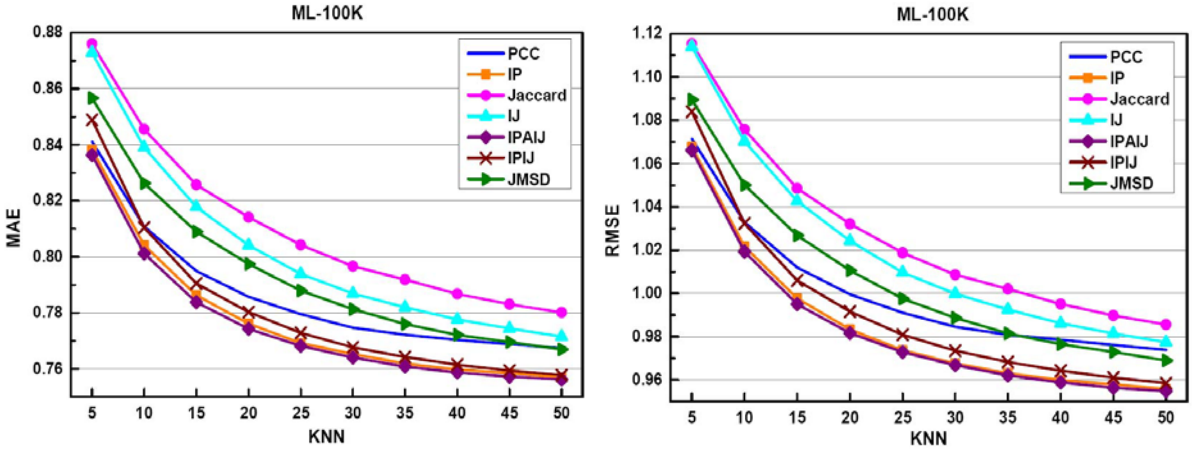
$$MAE = \sum_{(u,i)\in T} |r_{ui} - \widehat{r}_{ui}| \Big/ |T| \tag{13}$$

$$RMSE = \sqrt{\sum_{(u,i)\in T} (r_{ui} - \widehat{r}_{ui})^2 \Big/ |T|} \tag{14}$$

where T represents the set of prediction results and $|T|$ is the number of the set and $\widehat{r}_{ui}$ is the prediction rating of user u to item i.

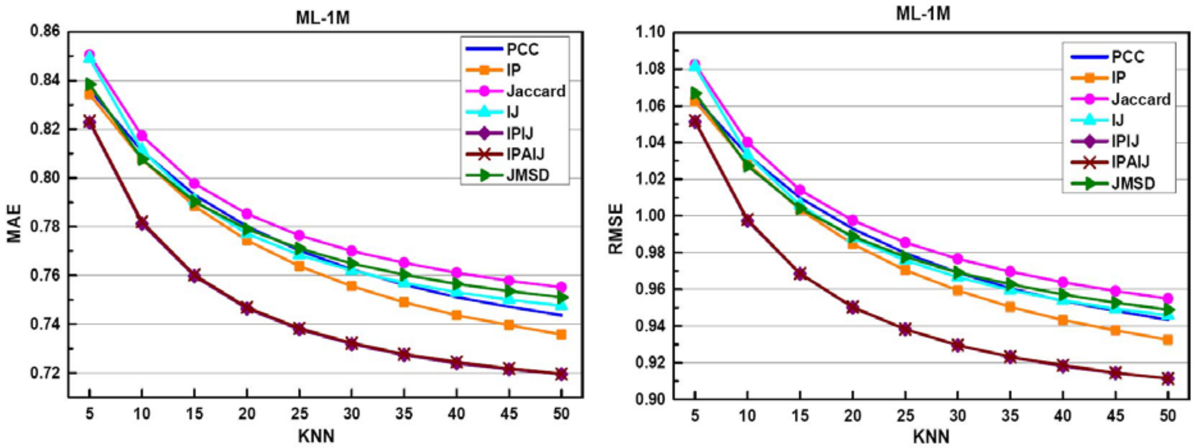
4.2. **Experimental results and analysis.** In this section, we compare the methods introduced above. The baseline approaches are PCC and Jaccard, and then the improved PCC (IP) and improved Jaccard (IJ). At last, we compare the final methods. One is IPIJ, which is a simple multiplication of the two improved methods; the other is IPAIJ, which is a linear combination of the two improved methods. In our experiment, ∂ is 0.7 on ML-100K and ∂ is 0.6 on ML-1M respectively. Besides, we also compare our methods with JMSD [3], which is the combination of Jaccard and MSD.

Figures 1(a) and 1(b) are the performances of these approaches on ML-100K in terms of MAE and RMSE respectively. The results show that IP is better than PCC while IJ is better than Jaccard. When KNN = 5, the result of PCC has little difference with IP, and Jaccard and IJ are by the same token. With the numbers of neighbors increasing, the differences are bigger and the effects of the improved methods are more obvious. Besides,



(a) (b)

Figure 1. The performances on MovieLens 100K in terms of MAE and RMSE



(a) (b)

Figure 2. The performances on MovieLens 1M in terms of MAE and RMSE

IPIJ, which is directly multiplied by IP and IJ where IP plays an equally important role with IJ, is not very ideal, especially when KNN = 5, the performance of IPIJ is worse than PCC. However, IJ is necessary because IPAIJ, which is a linear combination of IP and IJ, is the best method of all. JMSD is better than Jaccard and IJ, and its performance exceeds PCC with the increasing of KNN, but it is worse than IPIJ, IP and IPAIJ throughout. However, from Figures 1(a) and 1(b) we can see that the effect of IPAIJ is only a little better than IP. When KNN = 50, the results of IP, IPIJ and IPAIJ are almost the same. In our experiment ∂ is 0.7 in the method of IPAIJ, which implies that IJ plays a certain role on some degree and IP plays a more important role than IJ on ML-100K.

Figures 2(a) and 2(b) are the performances of these approaches on ML-1M in terms of MAE and RMSE respectively. We can also find that the improved methods are better than the traditional ones. When KNN is in 10-35, IJ is better than PCC. However, in the whole range, IJ is worse than IP. When KNN = 5, the results of IP and JMSD are almost the same, but with the increasing of KNN, IP is better than JMSD. From the two figures we can see that IPIJ and IPAIJ are obviously better than other methods and IPIJ is almost in the same level with IPAIJ. Since ∂ is 0.6 in IPAIJ and the effects of the two approaches are very close, we can see that IP plays an equally important role with IJ on ML-1M.

5. **Conclusions.** This paper presents a singularity-based method to improve the prediction accuracy of collaborative filtering recommender systems. First, considering the ratio of the number that item rated, we propose the improved Pearson correlation coefficient method. Besides, since the proportion of commonly rated items is also important and it will affect the quality of recommendation, we take Jaccard into consideration. Based on the idea of singularity, we propose positive singularity, negative singularity and empty singularity of each item, and then improved Jaccard method. At last, we combined the two methods in two different ways to make the two methods complement. Two real data sets which are the most commonly used are selected to validate our approach, and the experimental results demonstrate the effectiveness of our methods in improving the prediction accuracy of recommender systems.

REFERENCES

[1] C. Desrosiers and G. Karypis, A comprehensive survey of neighborhood-based recommendation methods, *Recommender Systems Handbook*, pp.107-144, 2011.

[2] F. Ortega, J. Sanchez, J. Bobadilla et al., Improving collaborative filtering-based recommender systems results using Pareto dominance, *Information Sciences*, vol.239, pp.50-61, 2013.

[3] G. Guo, J. Zhang and D. Thalmann, A simple but effect method to incorporate trusted neighbors in recommender systems, *Proc. of the 20th International Conference on User Modeling, Adaptation and Personalization (UMAP'12)*, vol.7379, pp.114-125, 2012.

[4] G. Adomavicius and A. Tuzhilin, Toward the next generation of recommender systems, *IEEE Trans. Knowl. Data Eng.*, vol.17, pp.734-749, 2005.

[5] G. Guo, J. Zhang and N. Smith, A novel Bayesian similarity measure for recommender systems, *Proc. of the 23rd International Joint Conference on Artificial Intelligence*, pp.2619-2625, 2013.

[6] G. Koutrica, B. Bercovitz and H. Garcia, Flexrecs: Expressing and combining flexible recommendations, *Proc. of the ACM SIGMOD International Conference on Management of Data*, pp.745-758, 2009.

[7] H. J. Ahn, A new similarity for collaborative filtering to alleviate the new user cold-starting problem, *Information Science*, vol.178, pp.37-51, 2008.

[8] H. Ma, D. Zhou, C. Liu et al., Recommender systems with social regularization, *Proc. of the 4th ACM International Conference on Web Search and Data Mining (WSDM'11)*, pp.287-296, 2011.

[9] J. Bobadilla, A. Hemando, F. Ortega et al., Collaborative filtering based on significances, *Information Sciences*, vol.185, pp.1-17, 2012.

[10] J. Bobadilla, F. Ortega and A. Hemando, A collaborative filtering similarity measure based on singularities, *Information Proceeding & Management*, vol.48, pp.201-217, 2012.

[11] J. Bobadilla, F. Ortega, A. Hemando et al., A collaborative filtering approach to mitigate the new user cold start problem, *Knowledge-Based Systems*, vol.26, pp.225-238, 2011.

[12] J. Bobadilla, F. Ortega and J. Bemal, A new collaborative filtering metric that improves the behavior of recommender systems, *Knowledge-Based Systems*, vol.23, pp.520-528, 2010.

[13] J. S. Breese, D. Heckerman, C. Kadie et al., Empirical analysis of predictive algorithms for collaborative filtering, *Proc. of the 14th Conference on University in Artificial Intelligence (UAI98)*, pp.43-52, 1998.

[14] N. Lathia, S. Hailes and L. Capra, Private distributed collaborative filtering using estimated concordance measures, *Proc. of the 2007 ACM Conference on Recommender Systems (RecSys07)*, pp.1-8, 2007.

[15] P. Resnick, N. Iacovou, M. Suchak et al., An open architecture for collaborative filtering of netnews, *Proc. of the ACM Conference on Computer Supported Cooperative Work*, pp.175-186, 1994.

[16] K. Choi and Y. Suh, A new similarity function for selecting neighbors for target item in collaborative filtering, *Knowledge-Based Systems*, vol.37, pp.146-153, 2013.

[17] G. Guo, J. Zhang and D. Thalmann, Merging trust in collaborative filtering to alleviate data sparsity and cold start, *Knowledge-Based Systems*, vol.57, pp.57-68, 2014.

[18] S. Deng, L. Huang and G. Xu, Social network-based service recommendation with trust enhancement, *Expert Systems with Applications*, vol.41, pp.8075-8084, 2014.

[19] B. Patra, R. Launonen, V. Ollikainen et al., A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data, *Knowledge-Based Systems*, 2015.