A New Similarity Measure Based on Adjusted Euclidean Distance for Memory-based Collaborative Filtering

Huifeng Sun
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China
Email: huifengsun.zj@gmail.com

Yong Peng, Junliang Chen, Chuanchang Liu, Yuzhuo Sun
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China
Email: {pengyong, chjl, lcc3265 }@bupt.edu.cn, lock.key2012@gmail.com

***Abstract*—Memory-based collaborative filtering (CF) is applied to help users to find their favorite items in recommender systems. Up to now, this approach has been proven successful in recommender systems, such as e-commerce systems. The idea of this approach is that the interest of a particular user will be more consistent with those who share similar preference with him or her. Therefore, it is critical that an appropriate similarity measure should be selected for making recommendations. This paper proposes a new similarity measure named adjusted Euclidean distance (AED) method which unifies all Euclidean distances between vectors in different dimensional vector spaces. Our AED enjoy the advantages that it takes both the length of vectors and different dimension-numbers of vector spaces into consideration. Based on two datasets MovieLens and Book-Crossing, we conduct experiments comparing our AED with two notable existing methods. The experimental results demonstrate that our AED improves the accuracy of prediction and recommendation.**

***Index Terms*—collaborative filtering, recommender systems, adjusted Euclidean distance, similarity measure**

I. Introduction

Information on the Web has been growing sharply and explosively over the last decade. Facing huge amounts of information, it is difficult, expensive or even impossible for users to obtain useful parts. As a solution of information overload, recommender systems have been proven successful. Recommender systems is a kind of information filters which provide personalized recommendations. For example, recommender systems in B2C e-commerce field assist users to find their favorite books, CDs, clothes and so on.

Recommender systems are usually classified into the following categories, based on how recommendations are made [1], [2]: (1) content-based recommendations in which the user will be recommended items similar to the ones the user preferred in the past; (2) collaborative recommendations in which the user will be recommended items that people with similar tastes and preferences like at present; (3) hybrid approaches which combine collaborative and content-based methods. As a powerful method with the advantage that formalizes human preference, collaborative filtering does not depend on the content of items, but purely rely on preference of a set of users. These preferences include two categories: (1) explicit preferences which can be indicated by numeric ratings; (2) implicit preferences which can be described by user behavior, such as buying a book, seeing a film, or clicking on a hyperlink. The information on personal preferences is included in explicit or implicit user ratings. According to [3], algorithms for collaborative filtering can be group into two classes: memory-based and model-based [4], [5]. In this article, we focus on memory-based CF and will elaborate it Section 2.

The current memory-based collaborative filtering still requires further improvements to make recommender systems more effective. Making recommendations relies on similarities among users or items, so the mechanism of calculating similarities is critical. Differing from traditional similarity measures, a new similarity measure based on adjusted Euclidean distance is proposed by us.

The remainder of this paper is organized as follows. After we review previous work in Section 2, we present a more elaborate motivation and explanation of our approach in Section 3. In Section 4, we show experimental results that demonstrate the effectiveness of adjusted Euclidean distance (AED). Then, we end the paper with conclusions and perspectives.

II. Related Work

In this section, we review previous research work on memory-based collaborative filtering. After giving a brief review on memory-based CF methods, we review similarity measures about memory-based CF.

A. Memory-based collaborative filtering

Memory-based CF are motivated by the phenomena that people usually trust the recommendations from like-minded friends. These methods apply a collection of nearest neighbor to predict a user's rating on particular item based on the ratings given by like-mined users. According to rating predictions depends on other similar users' rated values on the same item or on the active user's previous rated values on other similar items or on

doi:10.4304/jsw.6.6.993-1000

combining the former two, these methods are classified into user-based [3], [6], [7], [8], item-based [9], [10], [11] and combined approaches [12], [13].

As to user-based CF, [2] gives the formal definition, that is, the value of the unknown rating $r_{c,s}$ for user c and item s is usually computed as an aggregate of the ratings of some other users (usually, the N most similar) for the same items s:

$$r_{c,s} = \underset{c' \in \hat{C}}{aggr}\, r_{c',s}, \quad (1)$$

where $\hat{C}$ denotes the set of N users that are the most similar to user c and who have rated item s (N can range from 1 to the number of all users). The most widely adopted and popular aggregation functions are [2]:

$$r_{c,s} = \frac{1}{N}\sum_{c' \in \hat{C}} r_{c',s}, \quad (2)$$

$$r_{c,s} = k\sum_{c' \in \hat{C}} sim(c,c') \times r_{c',s}, \quad (3)$$

$$r_{c,s} = \bar{r}_c + k\sum_{c' \in \hat{C}} sim(c,c') \times (r_{c',s} - \bar{r}_{c'}), \quad (4)$$

where the normalizing factor is usually given as $k = 1/\sum_{c' \in \hat{C}} |sim(c,c')|$,and $\bar{r}_c$, the average rating of user c, is defined as $\bar{r}_c = (1/|S_c|)\sum_{s \in S_c} r_{c,s}$, where $S_c = \{s \in S \mid r_{c,s} \neq \varnothing\}$ (S is the set of all items). The most common aggregation approach is shown as Eq. (3) which supposes every user take the same rating scale. Wheras, Eq. (4) considers the fact that different users may take the different rating scales.

As regards item-based CF, the method predicts rating on item s for user c relying on the ratings given by user c on the items which are similar to item s. The value of the unknown rating $r_{c,s}$ for user c and item s is usually predicted as an aggregate of the ratings of some other items (usually, the M most similar) for the same user c:

$$r_{c,s} = \underset{s' \in \hat{S}}{aggr}\, r_{c,s'}, \quad (5)$$

where $\hat{S}$ denotes the set of M items that are the most similar to item s and who have been rated by user c (M can range from 1 to the number of all items). Some examples of aggregation functions are given as:

$$r_{c,s} = \frac{1}{M}\sum_{s' \in \hat{S}} r_{c,s'}, \quad (6)$$

$$r_{c,s} = k\sum_{s' \in \hat{S}} sim(s,s') \times r_{c,s'}, \quad (7)$$

where k, which serves as the normalizing factor, is usually given as $k = 1/\sum_{s' \in \hat{S}} |sim(s,s')|$.

When it comes to combined memory-based CF, [13] reformulates the memory-based collaborative filtering problem in a generative probabilistic framework, treating individual user-item ratings as predictors of missing ratings, and [12] proposes a solution which gives the final prediction by linear weight sum of user-based CF prediction and item-based CF prediction.

Because of the same form of principle between user-based and item-based CF, we only discuss user-based CF in this literature.

B. Similarity measures

Kinds of approaches have been taken to compute the similarity which includes user similarity $sim(c,c')$ between two users and item similarity $sim(s,s')$ between two items. In general, user similarity is based on their ratings of items that both users have rated and item similarity is based on the ratings of the two items from the users who rated both the two items. Cosine-based and Correlation are the most two popular approaches to compute similarity. We present the two approaches as follows with user similarity for example and readers could infer item similarity because of the same form of principle between user-based and item-based CF. Let S_{uv} be the set of all items co-rated by both users u and v, i.e., $S_{uv} = \{s \in S \mid r_{u,s} \neq \varnothing \,\&\, r_{v,s} \neq \varnothing\}$ (S is the set of all items). In the cosine-based approach [3], [11], the two users u and v are treated as two vectors in m-dimensional space, where $m = |S_{uv}|$. Thus, the similarity between two vectors can be measured by computing the cosine of the angle between them:

$$sim(u,v) = \cos(\vec{u},\vec{v}) = \frac{\vec{u}\cdot\vec{v}}{\|\vec{u}\|_2 \times \|\vec{v}\|_2} = \frac{\sum_{s \in S_{uv}} r_{u,s} r_{v,s}}{\sqrt{\sum_{s \in S_{uv}} r_{u,s}^2}\sqrt{\sum_{s \in S_{uv}} r_{v,s}^2}}. \quad (8)$$

In the correlation-based approach, the Pearson correlation coefficient is adopted to measure the similarity [14], [15]:

$$sim(u,v) = \frac{\sum_{s \in S_{uv}} (r_{u,s} - \bar{r}_u)(r_{v,s} - \bar{r}_v)}{\sqrt{\sum_{s \in S_{uv}} (r_{u,s} - \bar{r}_u)^2 \sum_{s \in S_{uv}} (r_{v,s} - \bar{r}_v)^2}}. \quad (9)$$

In additional to these approaches, some researchers have proposed several novel similarity measure approaches such as the mean squared difference [15], adjusted item similarity [11], the PIP (Proximity-Impact-Popularity) measure [16], user-class similarity [17], random walk counting [18], UNION similarity [19] and RIPs(rated-item pools) user similarity [20]. Different approaches have different strengths and purposes for different situations (e.g., cold-starting problem, sparse rating, implicit ratings, and so on). Despite of all these similarity measure approaches, it still requires that we develop a new similarity measure approach to give memory-based CF further improvements and make recommender systems more effective.

III. THE ADJUSTED EUCLIDEAN DISTANCE SIMILARITY MEASURE

A. Motivation

We analyze the shortcomings of the cosine-based and the correlation-based similarity measure approaches shown as Eqs. (8), (9). All the two approaches take into consideration the direction of vectors, but not take into account the length of them. A fragment of a rating matrix, which includes ratings range from 1 to 5, is showed in Table 1.

Table 1. A fragment of a rating matrix for a recommender system

	Item1	Itme2	Item3	Item4	Item5
User1	1	1	1	1	1
User2	2	2	2	2	2
User3	5	5	5	5	5
User4	1	2	4	3	3
User5	1	2	3	4	5
User6	2	2	3	4	∅
User7	3	∅	∅	∅	∅

Quantitative analysis as follows: whether we use Eq. (8) or Eq. (9), we can discover two groups of arithmetic expression

Group 1

sim(user2, user1) = sim(user2, user3) ,

sim(user4, user1) = sim(user4, user3) ,

sim(user5, user1) = sim(user5, user3) .

Group 2

sim(user6, user5) < sim(user6, user7).

According to Group 1, we can infer that preferences of user1 and user3 are very similar. Actually, the rating vectors of user1 and user3, which are $(1,1,1,1,1)$ and $(5,5,5,5,5)$ respectively, show that their preferences are opposite because 1 is the lowest rating in the recommender system and 5 is the highest. In the light of Group 2, we infer that user7 has more similar preference with user6 than user5. In accordance with Table 1, it is evident that user5 is more similar with user 6 than user7. Hence, Calculation results and facts are contradictory. The contradiction is caused by ignoring the length of vectors when computing similarities and by neglecting different number of dimensions in different vector spaces when comparing similarities. It is overlooking vector length that results in the contradiction stems form Group 1. It is overlooking dimension number difference in different dimensional vector spaces that causes the contradiction results from Group 2. In order to overcome these shortcomings, we create adjusted Euclidean distance (AED) similarity measure.

B. Rationale

Our AED similarity measure is based on normalizing distance between two vectors in multidimensional vector space. Euclidean (EU) distance can be used to estimate similarity as well, but AED is different from EU approach. In a recommender system, because different pairs of users co-rate different number of items, Euclidean distances among users usually compute in different dimensional spaces. Consequently, it makes little sense to put them together to measure similarity. For example, if user a and user b both rated the same 36 items, user c and user d both rated the same 183 items, there is no sense to mention the Euclidean distances $EUdist(a,b)$ and $EUdist(c,d)$ in the same breath because $EUdist(a,b)$ is computed in 36-dimensional space, $EUdist(c,d)$ in 183-dimensional space. Let $Eudist$ denote the Euclidean distance between an arbitrary pair of vectors in an arbitrary dimensional vector space, $Eudist_{\max}$ denote the maximal Euclidean distance in the same space. We propose the ratio of $Eudist$ and $Eudist_{\max}$ as a uniform method to unify Euclidean distances in different dimensional vector spaces. Then, the similarity between two vectors can be defined as $1-Eudist/Eudist_{\max}$.

We suppose that $V_{\min}$ and $V_{\max}$ are the lowest and highest rating value for a particular recommender system, and each user in the recommender system takes the same rating scale form $V_{\min}$ to $V_{\max}$. Let m-dimensional vector $\vec{u}$, $\vec{v}$ be the rating vector of user u and user v respectively; Let $dist(\vec{u},\vec{v})$ be the Euclidean distance between $\vec{u}$ and $\vec{v}$; Let $dist_{\max}$ be the maximal Euclidean distance in the m-dimensional vector space that each dimension ranges from $V_{\min}$ to $V_{\max}$. We propose the formula of AED as follow:

$$sim(u,v)=1-\frac{dist(\vec{u},\vec{v})}{dist_{\max}}=1-\frac{\sqrt{\sum_{s\in S_{uv}}(r_{u,s}-r_{v,s})^2}}{\sqrt{\sum_{k=1}^{m}(V_{\max}-V_{\min})^2}} \quad (10)$$

where S_{uv} denotes the set of all items co-rated by both users u and v, i.e., $S_{uv}=\{s\in S \mid r_{u,s}\neq\varnothing \,\&\, r_{v,s}\neq\varnothing\}$ (S is the set of all items) and $m=|S_{uv}|$ denotes the number of members in S_{uv}. In Eq. (10), $sim(u,v)\in[0,1]$, which represents the two users have completely opposite preference when it equal to 0 and the same preference when it equal to 1.

Accordingly, we suggest the aggregate function for predicting unknown rating value $r_{c,s}$ as follow:

$$r_{c,s}=\begin{cases} k\sum_{c'\in\hat{C}} sim(c,c')\times r_{c',s}, & \hat{C}_0=\varnothing \\ k_0\sum_{c'\in\hat{C}_0} m_{c'}\times r_{c',s}, & \hat{C}_0\neq\varnothing \end{cases} \quad (11)$$

where $\hat{C}$ denotes the set of N users that are the most similar to user c, $\hat{C}_0$ denotes the subset of $\hat{C}$ whose arbitrary member c' satisfies $dist(c,c')=0$ (i.e., Euclidean distance between arbitrary member c' and user c equal zero), $k=1/\sum_{c'\in\hat{C}}|sim(c,c')|$, $sim(c,c')$

is computed by Eq. (10), and $m_{c'}$ is the number of co-rated items for user c and c', $k_0 = 1/\sum_{c' \in \hat{C}_0} m_{c'}$. When $dist(c,c') = 0$, similarity measure as Eq. (10) loses the advantage of considering different dimension numbers of different dimensional spaces. In other words, similarity measure as Eq. (10) doesn't work for making prediction when $\hat{C}_0 \neq \varnothing$, so we propose a solution shown as the second section of Eq. (11) instead.

IV. Experiments and Evaluation

A. Datasets

The experiments are conducted based on the MovieLens[1] and Book-Crossing[2] [21]. The MovieLens (ML) data contains 100,000 ratings (1 to 5) of 1682 films from 943 users, where each user has rated at least 20 items. The Book-Crossing contains 1,149,780 ratings (433,681 explicit ratings on a scale form 1-10 and 716,109 implicit ratings expressed by 0) about 271,379 books from 278,858 users. We use a subset of Book-Crossing, BX1, which have 94,886 ratings. We select BX1 by the constraint that BX1 has ratings on books which have (explicitly or implicitly) rated by at least 20 users and from users who have (explicitly or implicitly) rated at least 200 books. Brief description of these datasets is showed in Table 2.

Table 2. Brief description of the ML and BX1 datasets

Description	ML	BX1
Ratings number	100,000	94,866
Matrix size	943 × 1682	644 × 4793
Sparsity level (%)	93.69	96.93

Sparsity level = 100 × (total entries – total no. of ratings) / (total entries)

B. Experimental setup and metrics

To show experimental effect of similarity measures, we take user-based CF for example because of the same form of principle between user-based and item-based CF. We conduct the experiments by adopting cosine-based (COS) CF, Pearson correlation coefficient-based (PCC) CF and AED CF. For each dataset in Table 2, we carry out a group of experiments in following ways. In each group of experiments, we use 5-fold cross validation and separate the dataset into training set (80% of the original ratings) and test set (the remaining 20% of the ratings) in each fold data. For ML dataset, we do experiments based on the training sets and test sets published on the homepage of ML; and for BX1, we perform experiments based on the training sets and test sets separated by us through a random method. Each fold experiment is run to predict ratings or make recommendation in the test set based on the ratings in the corresponding training set. Performance on each fold is evaluated by comparing prediction with true ratings. The final performance is the average of performance in all 5 folds. In our experiments, all test performances are generated in this manner. In order to compute similarity, the number threshold of co-rated items between two users is set to 10 or 20[3] in all experiments.

In CF research, researchers are typically interested in two types of accuracy, the accuracy for prediction and the accuracy for recommendation. The first one measures the performance when predicting the unknown ratings on items for active user. The second one focuses on finding an accurate sequence of a set of unrated items, in order that it can recommend the top ranked items to the active user. The two scenarios require different experimental metrics and setups, which will be described as follows.

1. Accuracy of Predicting Ratings. To evaluate the accuracy when predicting unrated item for the active user, we use Mean Absolute Error (MAE). MAE is defined as

$$MAE = \frac{\sum | r_{u,i} - \hat{r}_{u,i} |}{N}, \quad (12)$$

where $r_{u,i}$ is the ground truth, $\hat{r}_{u,i}$ denotes the predicted ratings on items by user u, and N is the total number of ratings in the test set.

2. Accuracy of Recommendations. To evaluate the accuracy of recommendations, we use Mean Average Precision (MAP), which is defined as Average of the Average Precision (AP) value for a set of queries (a query could be considered as a user's asking for recommending items in recommender systems). AP emphasizes ranking relevant items higher. It is the average of precisions computed at the point of each of the relevant items in the ranked sequence. MAP and AP are classic evaluation metrics in information retrieval systems, here we introduce them into recommender systems. According to [22], we define AP of top-k recommendation system as:

$$AP = \frac{\sum_{r=1}^{k} (P(r) \times rel(r))}{number\ of\ relevant\ items}, \quad (13)$$

where r is the rank, k is the number recommended, $rel(r)$ is a binary function on the relevance of a given rank r, and $P(r)$ is precision at a given cut-off rank:

$$P(r) = \frac{| \{relevant\ recommended\ items\} |}{r}. \quad (14)$$

So, we define MAP of recommendation system as:

$$MAP = \frac{\sum AP(u)}{number\ of\ recommended\ users}, \quad (15)$$

where AP(u) denotes AP of query form user u. For ML data, we assume that users are interested in those movies which they had rated 4 or 5. For BX1, we assume that users are interested in those books that they had been given a rating of 6-10.

[1] http://www.grouplens.org
[2] http://www.informatik.uni-freiburg.de/~cziegler/BX/

[3] We also perform experiments setting number threshold of co-rated items to other value, e.g. 5, 30, 40, the results also achieve better performance as the value is 10 or 20.

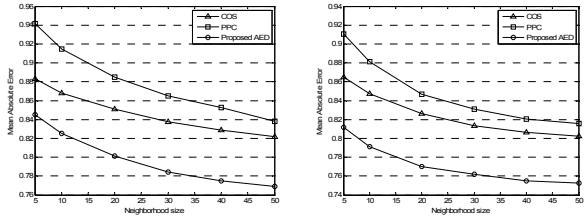
To calculate MAP, we use the following setup. For each active user from the test set (users who rated less than 30 or have no interesting items are ignored), we hide ratings of the user's in the test set. Then, the CF system predicts the ratings for these hidden-rating items. We recommend either the top 5 or the top 10 ranked items of these hidden-rating items to the user and then evaluate AP. MAP is the average of AP of all evaluated users.

C. Experimental results of MAE

Figs. 1, 2 show the performances of all evaluated CF methods in term of accuracy of prediction which is measured by MAE. Tables 3 and 4 describe these performances in detail. It can be seen that AED achieves MAE that is about 4.3-34.2 percent lower than those of the competing methods. Moreover, as shown in these Figs and Tables, AED enjoys stable performance improvement.

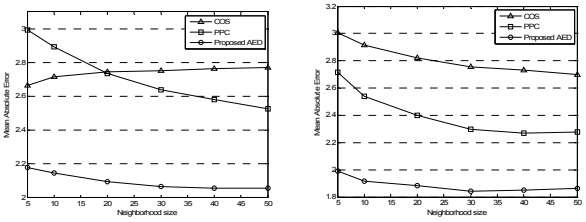
For one thing, taking account of different datasets, AED makes 4.3-10.8 percent fewer MAE than the competing methods for ML and 18.1-34.2 percent fewer for BX1, which is showed in Table 3 and 4 respectively. The results suggest that AED is particularly suitable for making predictions: AED achieves a particularly high improvement of accuracy measured by MAE. In accordance with Tables 3 and 4, AED gains MAE performance improvement on BX1 is 10.3-27.6 percent better than on ML.

For another, consider different number thresholds of co-rated items. Comparing COS curve with PPC curve in Figs.1 and 2, the below curve are selected to serve as the optimal curve of the competing methods, which we name competing-optimal-curve. Competing-optimal-curve is COS curve in Fig. 1 (a), COS curve in Fig. 2 (b), the curve which is composed of COS curve when neighborhood size is smaller than the point of intersection (of COS curve and PPC curve) and PPC curve when bigger than the point in Fig. 2 (a), PPC curve in Fig. 2 (b). On basis of the definition of competing-optimal-curve, we can see two points as follows. First, it is shown in Fig. 1 and Table 3 that the MAE reduction, which compares AED curve with competing-optimal-curve, is bigger when number threshold of co-rated items is 10 than 20 for ML except when neighborhood size is 5, 10 and 20. Second, according to Fig. 2 and Table 4, MAE reduction of AED curve VS competing-optimal-curve is bigger when number threshold of co-rated items is 10 than 20 except when neighborhood size is 5 and 10.



(a) (b)

Figure 1. MAE comparison between COS, PPC and Proposed AED for ML: (a) when co-rated items number threshold is 10; (b) when co-rated items number threshold is 20.



(a) (b)

Figure 2. MAE comparison between COS, PPC and Proposed AED for BX1: (a) when co-rated items number threshold is 10; (b) when co-rated items number threshold is 20.

Table 3. Comparison of Accuracy of Predictions Measured by MAE, of Different CF, for ML

co-rated threshold	No. of neighbors	COS	PCC	AED	Improvement over COS	Improvement over PPC
10	5	0.8833	0.9416	0.8451	0.0382(+4.3%)	0.0965(+10.3%)
	10	0.8680	0.9147	0.8251	0.0429(+4.9%)	0.0896(+9.8%)
	20	0.8512	0.8847	0.8010	0.0502(+5.9%)	0.0837(+9.5%)
	30	0.8377	0.8651	0.7841	0.0536(+6.4%)	0.0810(+9.4%)
	40	0.8287	0.8528	0.7751	0.0536(+6.5%)	0.0777(+9.1%)
	50	0.8214	0.8382	0.7693	0.0521(+6.4%)	0.0689(+8.2%)
20	5	0.8646	0.9104	0.8118	0.0528(+6.1%)	0.0986(+10.8%)
	10	0.8474	0.8810	0.7910	0.0564(+6.7%)	0.0900(+10.2%)
	20	0.8260	0.8470	0.7703	0.0557(+6.7%)	0.0767(+9.1%)
	30	0.8136	0.8307	0.7618	0.0518(+6.4%)	0.0689(+8.3%)
	40	0.8062	0.8202	0.7550	0.0512(+6.4%)	0.0652(+7.9%)
	50	0.8022	0.8159	0.7525	0.0497(+6.2%)	0.0634(+7.8%)

co-rated threshold denotes the number threshold of items for two users both rated, i.e., similarities among users are computed with the precondition that they have co-rated items whose number no less than the *co-rated threshold*.

Table 4. Comparison of Accuracy of Predictions Measured by MAE, of Different CF, for BX1

co-rated threshold	No. of neighbors	COS	PCC	AED	Improvement over COS	Improvement over PPC
10	5	2.6650	2.9910	2.1788	0.4862(+18.2%)	0.8122(+27.2%)
	10	2.7142	2.8928	2.1471	0.5671(+20.9%)	0.7457(+25.8%)
	20	2.7460	2.7337	2.0952	0.6508(+23.7%)	0.6385(+23.4%)
	30	2.7517	2.6393	2.0672	0.6845(+24.9%)	0.5721(+21.7%)
	40	2.7651	2.5792	2.0565	0.7086(+25.6%)	0.5227(+20.3%)
	50	2.7709	2.5263	2.0557	0.7152(+25.8%)	0.4706(+18.6%)
20	5	3.0042	2.7161	1.9905	1.0137(+33.7%)	0.7256(+26.7%)
	10	2.9150	2.5405	1.9180	0.9970(+34.2%)	0.6225(+24.5%)
	20	2.8226	2.3977	1.8830	0.9396(+33.3%)	0.5147(+21.5%)
	30	2.7541	2.2953	1.8432	0.9109(+33.1%)	0.4521(+19.7%)
	40	2.7328	2.2699	1.8513	0.8815(+32.3%)	0.4186(+18.4%)
	50	2.7002	2.2755	1.8645	0.8357(+30.9%)	0.4110(+18.1%)

co-rated threshold denotes the number threshold of items for two users both rated, i.e., similarities among users are computed with the precondition that they have co-rated items whose number no less than the *co-rated threshold*.

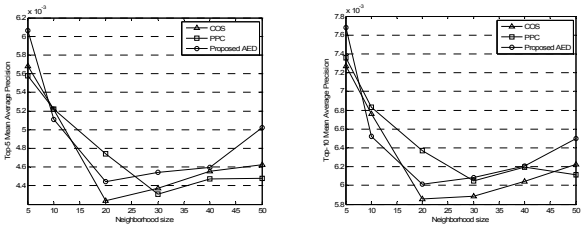
D. Experimental results of MAP

Figs. 3, 4 show the performances of all evaluated CF methods in term of accuracy of recommendations which is measured by MAP. Tables 5 and 6 describe them in detail. The big advantage of AED in terms of MAE does not fully carry over to MAP. Still, a significant gain in MAP performance could be achieved. MAP values of AED are typically about 2 to 165 percent better than those of the competing methods when the neighborhood size is big enough.

First of all, let's see effect on different datasets. For ML, whether top-5 or top-10 recommendation, MAP of AED is 0.2 to 12.1 percent higher than the competing methods except the point at which neighborhood size is 10 or 20, which is shown in Fig. 3 and Table 5. For BX1, AED achieves a significant performance level of 12 to

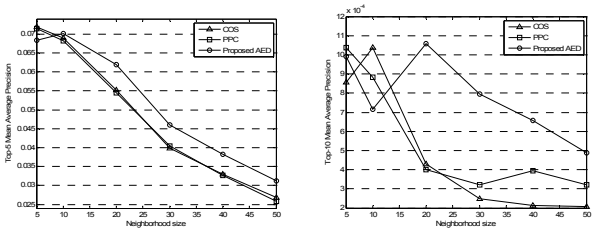
219.3 percent better than the competing methods in top-5 or top-10 recommendation when the neighborhood size is bigger than 10, which is shown in Fig. 4 and Table 6. In accordance with Tables 5 and 6, AED achieves MAP performance improvement on BX1 is typically 7.1-66.9 percent better than on ML. According to Figs. 3, 4, AED gains MAP performance improvement on BX1 is more stable than on ML.

Next, let's notice the impact of neighborhood size on performance. In accordance with Fig. 3, AED achieves the best MAP performance in all the three methods except when neighborhood size is 10 or 20. In the light of Fig. 4, AED outperforms the other two methods except for neighborhood size 5 or 10, with 12-219 percent better MAP performance. In a word, AED gains better MAP performance than the computing methods when neighborhood size is big enough.



(a) (b)

Figure 3. MAP comparison between COS, PPC and Proposed AED for ML: (a) Top-5; (b) Top-10.



(a) (b)

Figure 4. MAP comparison between COS, PPC and Proposed AED for BX1: (a) Top-5; (b) Top-10.

Table 5. Comparison of Accuracy of Recommendations Measured by MAP, of Different CF, for ML

	No. of neighbors	COS	PCC	AED	Improvement over COS	Improvement over PPC
TOP5	5	0.00568	0.00558	0.00606	0.00038 (+6.7%)	0.00048 (+ 8.7%)
	10	0.00521	0.00522	0.00511	-0.0001 (-2.0%)	-0.00011 (-2.1%)
	20	0.00424	0.00474	0.00444	0.0005 (+4.9%)	-0.0003 (-6.3%)
	30	0.00437	0.00431	0.00454	0.00017 (+3.9%)	0.00023 (+ 5.4%)
	40	0.00455	0.00447	0.00460	0.00005 (+ 0.9%)	0.00013 (+ 2.9%)
	50	0.00462	0.00448	0.00502	0.0004 (+ 8.7%)	0.00054 (+12.1%)
TOP10	5	0.00727	0.00736	0.00768	0.00041 (+5.6%)	0.00032 (+4.4%)
	10	0.00676	0.00683	0.00653	-0.00023 (-3.5%)	-0.0003 (-4.5%)
	20	0.00586	0.00637	0.00601	0.00015 (+2.7%)	-0.00036 (-5.6%)
	30	0.00588	0.00605	0.00608	0.0002 (+3.4%)	0.00003 (+0.6%)
	40	0.00605	0.00619	0.00620	0.00015 (+2.6%)	0.00001 (+0.2%)
	50	0.00623	0.00611	0.00650	0.00027 (+4.3%)	0.00039 (+6.3%)

Table 6. Comparison of Accuracy of Recommendations Measured by MAP, of Different CF, for BX1

	No. of neighbors	COS	PCC	AED	Improvement over COS	Improvement over PPC
TOP5	5	0.07183	0.07145	0.06840	-0.00343 (-4.8%)	-0.00305 (-4.3%)
	10	0.06896	0.06828	0.07018	0.00122 (+1.8%)	0.0019 (+2.8%)
	20	0.05525	0.05441	0.06191	0.00666 (+12.0%)	0.0075 (+13.8%)
	30	0.03983	0.04052	0.04598	0.00615 (+15.4%)	0.00546 (+13.5%)
	40	0.03306	0.03266	0.03829	0.00523 (+15.8%)	0.00563 (+17.2%)
	50	0.02686	0.02586	0.03127	0.00441 (+16.4%)	0.00541 (+20.9%)
TOP10	5	0.00087	0.00104	0.00099	0.00012 (+15.8%)	-0.00005 (-4.5%)
	10	0.00104	0.00088	0.00072	-0.00032 (-31.1%)	-0.00016 (-19.1%)
	20	0.00043	0.00040	0.00106	0.00063 (+146.5%)	0.00066 (+164.4%)
	30	0.00025	0.00032	0.00080	0.00055 (+219.3%)	0.00048 (+147.9%)
	40	0.00021	0.00039	0.00069	0.00048 (+207.8%)	0.0003 (+67.1%)
	50	0.00021	0.00032	0.00049	0.00028 (+133.6%)	0.00017 (+51.3%)

E. Discussion

AED achieves an accuracy that is superior to COS and PPC which are considered as the most popular methods for memory-based collaborative filtering. On one hand, AED achieves a high better MAE performance than the other two methods in all the experiments for each dataset selected. On the other hand, AED outperforms the other two methods in terms of MAP when neighborhood size is big enough. Here, we are interested in and focus on the following three observations in our experiments.

First, AED gains a better MAE performance with the setting of co-rated items number threshold 10 than 20 when neighborhood size is big enough. The reason of this is that the degree of scatteration in different dimensional vector spaces increases when co-rated items number threshold decreases. The smaller the co-rated items number threshold, the more dimensional vector spaces for the user vectors are scattered into and so the higher degree of scatteration in different dimensional vector spaces. Because AED considers different number of dimensions in different dimensional vector spaces when measuring similarities and the other two methods ignore this, the higher degree of scatteration in different dimensional vector spaces, the better performance AED reaches.

Second, AED outperforms COS and PPC when neighborhood size is big enough. This is because that the bigger the neighborhood size is, the more neighbors can depend on when predicting unknown ratings, and that leads to the bigger probability that neighbor vectors scatter into more vector spaces with different dimensions.. In other words, prediction relies on more scattered neighbor vectors from different dimensional vector spaces. With the increase of scatteration degree in different dimensional neighbor vector spaces, there is superiority for AED relative to the other two methods in prediction because that AED takes different number of dimensions into consideration when scaling similarities and the other two methods neglect. Therefore, if the

neighborhood size is big enough, a large better performance gain will be always achieved for AED.

Third, performance improvement on BX1 is better than on ML, whether measured by MAE or MAP. We infer that this is caused by that BX1 has a higher sparsity level than ML (shown as Table 2). With the growth of sparsity level, there is a bigger probability that user vectors scatter into more vector spaces with different dimensions. Actually, it is certain that the sparsity difference between BX1 and ML leads to difference of scatteration in different dimensional spaces between them, because of two points as follows: (1) BX1 has 4793 items that is more than ML who has 1682 items; (2) dimension bound for user vectors of BX1 is larger than ML, since dimension of user vectors for BX1 ranges from 1 to 2694 (40 user vectors of the total 644 vectors has dimension less than 20), and for ML ranges from 20 to 737. As AED enjoys the advantage of taking into account different dimensions factor that aims at the scatteration of different dimensional spaces to measure similarity, it has superiority relative to the other two methods when the experimental dataset is sparse.

In a word, AED enjoys two advantages: (1) taking into consideration the length of vectors to scale similarity; (2) taking into account different number of dimensions for different dimensional vector spaces to measure similarity. The two advantages not only consider the distribution in the same vector space, but also consider the distribution in the different vector spaces. Consequently, AED is a comprehensive solution.

V. CONCLUSIONS

In this paper, we propose the adjusted Euclidean distance (AED) similarity measure for memory-based collaborative filtering. AED is based on Euclidean distance between two vectors in multidimensional vector space, but overcomes shortcoming of Euclidean distance method. When it comes to similarity measure, whether by Euclidean distance, by cosine of the angle or by Pearson correlation coefficient approach, the length of vectors or different number of dimensions of vector spaces are ignored, but these are all considered by AED. With AED, it is meaningful and effective for comparing similarities of different dimensional vector spaces. An experimental comparison with other similarity measure methods (cosine-based approach and correlation-based approach) shows that AED outperforms the competing methods both in terms of accuracy for prediction and recommendation.

We believe that AED approach will contribute to further improvements of recommender systems: (1) AED approach will be a promising solution to one of CF methods' limitation known as sparsity problem; (2) AED approach will be a promising basis for hybrid systems. On one hand, in recent years, with the rapid growth of magnitudes of users and items in e-commence field, extreme sparsity of users' rating data appears, resulted in poor performance of traditional CF and decreased quality for making recommendations. On the other hand, A popular research direction is the combination of CF methods with content-based filtering into hybrid systems [23], [24]. So, in the future, we plan to study on solving sparsity problem with the consideration of AED and hybrid systems combine AED CF with content-based filtering.

ACKNOWLEDGMENT

Thanks for the providers of MovieLens and Book-Crossing for their datasets. The work described in this paper was supported by the National Grand Fundamental Research 973 Program of China (No. 2011CB302506), the Novel Mobile Service Control Network Architecture and Key Technologies (No.2010ZX03004-001), the Fundamental Research Funds for the Central Universities (No.2009RC0507), and the National Natural Science Foundation of China (No. 61001118).

REFERENCES

[1] M. Balabanovic and Y. Shoham, "Fab: content-based, collaborative recommendation," Comm. ACM, vol. 40, March 1997, pp. 66-72.

[2] Adomavicius G. and Tuzhilin A., "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," IEEE Trans. on Knowledge and Data Engineering, vol. 17, June 2005, pp. 734-749.

[3] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative ltering," Proc. the 14th Conference on Uncertainty in Artificial Intelligence (UAI 98), Morgan Kaufmann Publisher, May 1998, pp. 43-52.

[4] Xin Xin, Lrwin King, Hongbo Deng, and Michael R. Lyu, "A social recommendation framework based on multi-scale continuous conditional random fields," Proc. the 18th ACM Conference on Information and Knowledge Management (CIKM 09), ACM Press, Nov. 2009, pp. 1247-1256.

[5] Nathan N. Liu, Min Zhao, and Qiang Yang, "Probabilistic latent preference analysis for collaborative filtering," Proc. the 18th ACM Conference on Information and Knowledge Management (CIKM 09) , ACM Press, Nov. 2009, pp. 759-766.

[6] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative ltering," Proc. the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 99), ACM Press, Aug. 1999, pp. 230–237.

[7] R. Jin, J. Y. Chai, and L. Si, "An automatic weighting scheme for collaborative ltering," Proc. the 27nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 04), ACM Press, July 2004, pp. 337–344.

[8] Zhao ZD and Shang MS, "User-based collaborative-filtering recommendation algorithms on hadoop," Proc. the 3rd International Conference on Knowledge Discovery and Data Mining (WKDD 10), IEEE Computer Society, Jan. 2010, pp. 478-481.

[9] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms", ACM Transactions on Information Systems, vol. 22, Jan. 2004, pp. 143–177.

[10] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative ltering," IEEE Internet computing, vol. 7, Jan. 2003, pp. 76–80.
[11] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. "Item-based collaborative ltering recommendation algorithms", Proc. the 10th International World Wide Web Conference (WWW 01), ACM Press, May 2001, pp. 285–295.
[12] H. Ma, I. King, and M. Lyu, "Effective missing data prediction for collaborative ltering," Proc. the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 07), ACM Press, July 2007, pp. 39–46.
[13] J. Wang, A. De Vries, and M. Reinders, "Unifying user-based and item-based collaborative ltering approaches by similarity fusion," Proc. the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 06), ACM Press, Aug 2006, pp. 501–508.
[14] P. Resnick, N. Iakovou, M. Sushak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," Proc. the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW 94), ACM Press, Oct. 1994, pp. 175-186.
[15] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating 'word of mouth'," Proc. the ACM CHI 95 Human Factors in Computing Systems Conference, ACM Press, May 1995, pp. 210-217.
[16] H.J. Ahn, "A new similarity measure for collaborative ltering to alleviate the new user cold-starting problem," Information Sciences , vol. 178, Jan. 2008, pp. 37–51 .
[17] C. Zeng, C.-X. Xing, L.-Z. Zhou, and X-H. Zheng, "Similarity measure and instance selection for collaborative ltering," International Journal of Electronic Commerce, vol. 8, Summer 2004, pp. 115–129.
[18] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," IEEE Transactions on Knowledge and Data Engineering , vol. 19, Jan. 2007, pp. 355–369.
[19] P. Symeonidis, A. Nanopoulos, A.N. Papadopoulos, and Y. Manolopoulos, "Collaborative ltering: fallacies and insights in measuring similarity," Proc. the 10th PKDD Workshop on Web Mining (WEBMine 06), Sep. 2006, pp. 56–67.
[20] Yue Shi, Martha Larson, and Alan Hanjalic, "Exploiting user similarity based on rated-item pools for improved user-based collaborative filtering," Proc. the 3rd ACM conference on Recommender systems (RecSys 09), ACM Press, Oct. 2009, pp. 125-132.
[21] C.-N. Ziegler, S.M. McNee, J.A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversi cation," Proc. the 14th International World Wide Web Conference (WWW 05), ACM Press, May 2005, pp. 22–32.
[22] http://en.wikipedia.org/wiki/Information_retrieval#Mean_ Average_precision.
[23] Asela Gunawardana and Christopher Meek, "A unified approach to building hybrid recommender systems," Proc. the 3rd ACM conference on Recommender systems (RecSys 09), ACM Press, Oct. 2009, pp. 117–124.
[24] Ghazanfar MA and Prugel-Bennett A, "A scalable, accurate hybrid recommender system," Proc. 3th International Conference on Knowledge Discovery and Data Mining (WKDD 10), IEEE Computer Society, Jan. 2010, pp. 94-98.

Huifeng Sun was born in Zhejiang Province, China in 1982. He received his B.S. from Zhejiang University in 2006.

He is currently a Ph.D. Candidate in the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. His research interests include machine learning, data mining, cloud computing and service computing.

Yong Peng was born in Hubei Province, China in 1978. He received his Ph.D. from Beijing University of Posts and Telecommunications in 2004.

He is currently an associate professor in the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. His research interests include service computing and wireless networks.

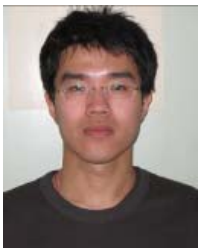
Junliang Chen was born in Zhejiang Province, China in 1933. He received his associate Ph.D. from Moscow Institute of Telecommunications Engineering in 1961.

He is currently a professor in the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. His research domains cover communication network and communication software.

Prof. Chen is a member of China Academy of Sciences(CAS) , and China Academy of Engineering(CAE). He is an IEEE fellow.

Chuanchang Liu received his B.S. in computer science from Harbin Institute of Technology, and his M.S. and Ph.D. in computer science from Beijing University of Posts and Telecommunications.

He is currently an assistant professor in the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. His research interests include network service & intelligence, service computing and semantic Web.



Yuzhuo Sun was born in Liaoning Province, China in 1985. He received his B.S. from Beijing University of Posts and Telecommunications in 2008.

He is pursuing his M.S. in the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. His research interests is data mining and service computing.