A Collaborative Filtering Recommender Approach by Investigating Interactions of Interest and Trust

Surong Yan

Abstract Collaborative filtering-based recommenders operate on the assumption that similar users share similar tastes; however, due to data sparsity of the input ratings matrix, traditional collaborative filtering methods suffer from low accuracy because of the difficulty in finding similar users and the lack of knowledge about the preference of new users. This paper proposes a recommender system based on interest and trust to provide an enhanced recommendations quality. The proposed method incorporates trust derived from both explicit and implicit feedback data to solve the problem of data sparsity. New users can highly benefit from aggregated trust and interest in the form of reputation and popularity of a user as a recommender. The performance is evaluated using two datasets of different sparsity levels, viz. Jester dataset and MovieLens dataset, and are compared with traditional collaborative filtering-based approaches for generating recommendations.

Keywords Collaborative filtering · Data sparsity · Cold start · Interest and trust

1 Introduction

Recommender systems have been developed as a solution to the well information overload problem [1, 2], by providing users with more proactive and personalized information services. The design of recommender systems that has seen wide use is collaborative filtering (CF). CF is based on the fact that word of mouth opinions

S. Yan (✉)
College of Computer Science and Technology, Zhejiang University,
Hangzhou 310027, China
e-mail: bkspace3000@gmail.compersephone@zufe.edu.cn

S. Yan
College of Information, Zhejiang University of Finance and Economics,
Hangzhou 310018, China

F. Sun et al. (eds.), *Knowledge Engineering and Management*,
Advances in Intelligent Systems and Computing 214,
DOI: 10.1007/978-3-642-37832-4_16, © Springer-Verlag Berlin Heidelberg 2014

of other people have considerable influence on the decision making of buyers [3]. If reviewers have similar preferences with the buyer, he/she is much more likely to be affected by their opinions.

In CF-based recommendation schemes, two approaches have mainly been developed: memory-based CF (also known as user-based CF) [1, 4] and item-based CF [5, 6]. User-based CF approaches have seen the widest use in recommendation systems. User-based CF uses a similarity measurement between neighbors and the target users to learn and predict preference towards new items or unrated products by a target user. An item-based CF reviews a set of items the target user has rated (or purchased) and selects the most similar items based on the similarities between items. Item-based CF has been applied to commercial recommender systems such as Amazon.com [7].

CF systems encounter two serious limitations with quality evaluation: the sparsity problem and the cold start problem. The sparsity problem occurs when an available data is insufficient for identifying similar users or items (neighbors) due to an immense amount of users and items [5]. The second problem, the so-called cold start problem, can be divided into cold-start items and cold-start users [8]. New items cannot be recommended until some user rate it, and new users are unlikely to receive good recommendations because of the lack of their rating or purchase history. A cold-start user is the focus of the present research. In this situation, the system is generally unable to make high quality recommendations [9]. A number of studies have attempted to resolve these problems in various applications, such as [10] and [11] tried to address the new user (or cold start) problem, [12] and [13] discussed the new item (or the first rater) problem, and [10], [11], [14], [15], and [16] studied the sparsity problem.

Approaches incorporating trust models into recommender systems are gaining momentum [17]. The emerging channel of SNS not only permits users to express opinions, but also enables users to build various social relationships. For example Epinions.com allows users to review various items (cars, books, music, etc.), and to assign a trust rating on reviewers as well as to build a trust relationship with another by adding him or her to a trust list, or block him or her with a block list. Some researchers [18–20] argued that this trust data can be extracted and used as part of the recommendation process, especially as a means to relieve the sparsity problem. Two computational models of trust namely profile-level trust and item-level trust have been developed and incorporated into standard CF frameworks [17]. The explicit trust-based recommender system models have high rating prediction. New users can also benefit from trust propagation as long as the users provide at least one trusted friend. However, these approaches rely on models of trust that are built from the direct feedback of users. Actually, in reality, the explicit trust statements may not always be available [21].

Also, there are needs for producing valuable recommendations when input data (trust data) is sparse and new user problem is present. These problems are intrinsic in the process of finding similar neighbors. Bedi and Sharma [22] tried to propose trust-based Ant recommender system (TARS) by incorporating a notion of dynamic trust between users to provide better recommendations. New users can benefit from pheromone updating strategy. But TARS still suffers from low accuracy due to the lack of the feedback rating of user.

To address the problems above, this paper proposes a novel approach, combination of interest and trust (CIT), to improve recommendation quality and value. Our wok focuses on sparseness in user similarity due to data sparsity of input matrix and difficulty in solving the new user's due to the lack of knowledge about preference of new user, and differs from the above works in a number of ways [17, 22]. First, the taste similarity measurement considers the number of overlapping items and distance between ratings of overlapping items. Second, as opposite to traditional predication methods, the prediction rating takes into account the difference of rating scale of user. Third, interest and trust score is generated based on user's explicit and implicit feedback behaviors. Fourth, definition of heavy users considers both virtual community-specific properties and trust-specific properties.

The rest of this paper is organized as follows: Section 2 describes the proposed CIT approach and provides a detailed description of how the system uses CIT for recommendation. Section 3 presents the effectiveness of CIT approach through experimental evaluations compared with existing work using the two datasets. Finally, in Sect. 4, the paper concludes with a discussion.

2 Proposed Recommender Approach by Investigating Interactions of Interest and Trust

In this section, the paper first derives ratings of users on items by identifying user's explicit and implicit feedback behavior; then estimates the interest similarity and trust intensity between users; third generates recommendations for target user by selecting small and best neighborhood; finally, discusses a solution to new user (cold start) problem by finding the migratory behavior of heavy users based on the Pareto principle rule.

2.1 Deriving Implicit Ratings of Users on Items

Feedback behaviors of a user not only show her/his personal interest in item recommendation list but also exhibit her/his degree of satisfaction on these items or products.

Deriving implicit item rating based on feedback behaviors: Target user can directly rate on recommended item by voting or clicking like/dislike button, or exhibit the degree of satisfaction on recommended item by indirect feedback behaviors. Such behaviors include buying the product, browsing the information of recommendation list, or adding the product into the favorites list or wish list. Leveraging implicit user feedback [23] has become more popular in CF systems.

In this paper, we leverage implicit user feedback behaviors, which can be viewed as a variant of user feedback. Therefore, rating of target user u on item j can be derived by direct or indirect behaviors of u.

If target user does not give a rating $r_{u,\ j}$ about recommended item j, recommender system will evaluate the feedback behaviors of target user based on association rule [14] and fuzzy relation [24] to generate the rating $r_{u,\ j}$ of target user on item j.

Deriving implicit feedback rating based on feedback behaviors: Referral feedback can be derived by directly feedback behavior of target user such as voting or clicking like/dislike button, or by indirectly feedback behavior of target user such as buying the product, browsing the information of recommendation list, or putting it into her/his favorites list or wish list. If target user does not give a vote/feedback rating $f_{u,\ v}(j)$, recommender system will evaluate the feedback behaviors of target user as described above and then generate the feedback rating.

Definition 1 Feedback rating: let $r_{u,\ i}$ and $p_{u,\ i}$ denote the real rating and predicted rating of target user u on item j respectively, the feedback rating of u on the item j recommended by v in time t based on fuzzy relation is

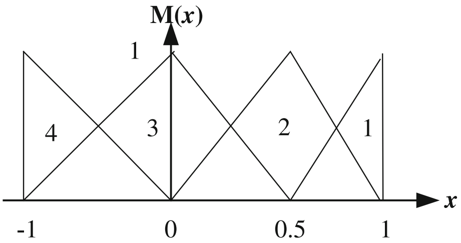
$$f^{t}_{u,v\in \mathrm{reco}(u,j)}(j) = M(|r_{u,j} - p_{u,j}|) \tag{1}$$

where reco(u, j) is the set of recommendation partners that have recommended target user u with item j in the latest recommendation round and M is the membership function that maps feedback rating of item j in different intervals. Figure 1 shows a membership function for the feedback rating.

2.2 Calculating Interest and Trust Scores

Calculating interest similarity score: There are a number of possible measures for computing the similarity, for example, the distance metric [11, 25], cosine/vector similarity [13, 15] and the Pearson correlation metric [2, 13]. The proposed approach defines a new taste similarity measurement for computing interest similarity between two users based on the overlapping part of their rating profiles, i.e., on the items that are rated by them in common:

Fig. 1 Membership function for feedback rating on rating scale (1–5)



Definition 2 Similarity rating: the similarity rating models the difference between ratings of two users on given common item. Let $r_{u,\ i}$ and $r_{v,\ i}$ denote the ratings of users u and v for item i respectively, the similarity rating of the item i between u and v based on fuzzy relation is

$$s_{u,v}(i) = M(|r_{u,i} - r_{v,i}|) \tag{2}$$

where M is the membership function that maps the absolute difference of rating between u and v on item i to the similarity rating of u and v on item i at different intervals.

The similarity function measures the taste similarity degree between the two users based on their ratings on common items and characterizes each participant's contextual view of the taste similarity. Let τ be the total number of items that u have rated, com(u, v) be the set of items rated by u, and v in common and q be the size of the set, $p = \sum_{i \in \mathrm{com}(u,v)} s_{u,v}(i)$ be the sum of the similarity rating between u and v on the ratings of the common items, and $\ell = \sqrt{\frac{1}{n}\sum_{i \in \mathrm{com}(u,v)} |r_{u,i} - r_{v,i}|^2}$ be the average of the absolute difference square of common rating between u and v. The taste similarity degree between u and v from the perspective of u is

$$\mathrm{Sim}(v/u) = \begin{cases} \frac{2p*q}{\tau(p+q)} * \exp(-\kappa\ell) & \text{if } p \neq 0 \text{ and } p + q \neq 0 \\ \eta\frac{q}{\tau} & \text{if } q \neq 0 \text{ and } p = 0 \\ 0 & \text{else} \end{cases} \tag{3}$$

where $0 < k$, $\eta < 1$ are adjust arguments. This taste similarity degree captures the intuition that user v should have rated many of the items that user u has rated and their ratings on these common items are similar. As seen from Eq. (3), sim (v/u) is large if rating overlaps between the two partners u and v is high and less otherwise.

Calculating trust scores: Trust intensity being dynamic information, increases or decreases with the involvement (positive experience or negative experience) or non-involvement (over time without any experience) of user as a recommender. Initially at time $t = 0$, dynamic trust intensity depends on the existed trust or distrust relationships of target user such as friend list or follow list or block list. A default trust value is preset. If there are any trust relationships absent, the paper will discuss this problem later.

At time $t > 0$, dynamic trust intensity is calculated as follows: Trust intensity determines the amount of confidence a user should have in (on) other user and characterizes each participant's contextual view of the trust intensity. Let $f^t_{u,v}(i) \in [-1, +1]$, denote the feedback rating of target user u on item i recommended by v in the time t, n denote the total number of feedback ratings of u on v, $m = \sum_{t=1}^{n} f^t_{u,v}(i)$ denote the sum of feedback ratings, and Γ denote the total number of recommendations that u have received from v. The amount of confidence a user u should have on user v is given as:

$$\mathrm{Trust}(v/u) = \begin{cases} \frac{2m*n}{\Gamma(m+n)} * \exp(-\kappa\ell) & \text{if } m \neq 0 \text{ and } m+n \neq 0 \\ \eta \frac{n}{\Gamma} & \text{if } n \neq 0 \text{ and } m = 0 \\ 0 & \text{else} \end{cases} \quad (4)$$

Combination of taste similarity and trust intensity (CIT*)*: Let $C(v/u)^t$ denote the value of CIT in time t, it is a weighted sum of taste similarity and trust intensity between u and v from the perspective of u:

$$C(v/u)^t = \chi \mathrm{sim}(v/u)^t + (1-\chi)\mathrm{trust}(v/u)^t \quad (5)$$

where $0 \leq \chi \leq 1$ is a small constant value. The value of $C(v/u)^t$ lies in between >-1 to 1 indicating the degree of confidence ranging in between 0 to 1. Since $C(v/u)^t \leq 0$ indicates that u and v are not correlated, therefore zero and negative correlations are not considered and all such outcomes are set to zero.

2.3 Generating Prediction Scores

Scale and translation invariance states that each user may have its own scale of rating items and is a stronger condition than normalization invariance [4]. For example, some users might rate most items as roughly similar while others tend to use more often extreme ratings. For compensating the difference in rating scales between users, the paper adopts the scale and translation invariance to produce predictions, which depends on the following factors:

Definition 3 Complete factor: Complete factor w is a weighted sum of difference between ratings $r_{v,\ j}$ $(v = 1, 2 \ldots)$ of selected best neighborhood ner(u) of u on given item j and their average rating $\overline{r_v}$ on all rated items respectively. The complete rating is defined as

$$w_{u,j}^t = \frac{\sum_{v \in \mathrm{ner}(u)} C(v/u)^t (r_{v,j} - \overline{r_v})}{\sum_{v \in \mathrm{ner}(u)} |C(v/u)^t|} \quad (6)$$

where $c(v/u)^t$ denotes the value of CIT that user u holds about user v at time t.

Definition 4 Adjust scale factor: Adjust scale factor $\Im$ is an adjusting about scales differences between ratings of target user and ratings of selected neighborhood on common items.

$$\Im^t = \frac{\sum_{i \in \mathrm{com}(u,v)} (r_{u,i} - \overline{r_u})(w_{u,i} - \overline{w_u})}{\sum_{i \in \mathrm{com}(u,v)} (w_{u,i} - \overline{w_u})(w_{u,i} - \overline{w_u})} \quad (7)$$

where $r_{u,\ i}$ and $w_{u,\ i}$ denote the actual ratings and the predicted complete ratings of user u on item i respectively, $\overline{r_u}$ and $\overline{w_u}$ denote the average actual ratings and the average predicted complete ratings of users u respectively.

Combining the above two weighting factors, the final prediction for the target user u on item j at time t ($t > 0$) is produced as follows:

$$p_{u,j}^{t} = \overline{r_u} + \Im^{t}(w_{u,j}^{t} - \overline{w_u}) \tag{8}$$

where r_u denotes the average ratings of users u on all rated items.

2.4 Addressing to New User (Cold Start) Problem

In a user's interest network, local similarity is aggregated on each edge connecting similar users and the user. Summing up the aggregated local similarity corresponding to all incoming links of the user and number of incoming links to the user, we get the global taste similarity. Global taste similarity acts as a positive indicator reflecting the user's taste that is popular or unpopular, and the popular users are defined as candidate heavy users over a period of time.

In a user's trust network, local trust is aggregated on each edge connecting trustworthy users and the user. Summing up the aggregated local trust corresponding to all incoming links of the user and number of incoming links to the user, we get the global trust. Users benefit from this wisdom of crowds because global trust acts as positive feedback reflecting the user's behavior that is universally considered good or bad, and high reputation users are defined as candidate heavy users over a period of time.

Definition 5 Heavy user: User with a long-term recommender history, high global taste similarity, and high trust intensity is defined as a heavy user.

List of heavy user can act as default friends list at time t and be added as friends of new user at initial time $t = 0$. Combining ratings of the top_n heavy users on item j, the final prediction for the new user u on item j at time t ($t = 0$) is produced as follows:

$$p_{u,j}^{0} = \frac{\sum_{v=1}^{\mathrm{top\_}n} \overline{r_v}}{\mathrm{top\_}n} + \frac{\sum_{v=1}^{\mathrm{top\_}n} G(v)(r_{v,j} - \overline{r_v})}{\sum_{v=1}^{\mathrm{top\_}n} G(v)} \tag{9}$$

where $G(v)$ is a weight sum of global taste similarity and global trust intensity of v.

3 Experimental Study

In this section, the paper describes the experimental methodology and metrics that are used to compare different prediction algorithms; and presents the results of the experiments.

3.1 Experimental Configuration

The experimental study was conducted on two different types of datasets. The Jester dataset and MovieLens dataset available online on websites www.ieor.berkeley.edu/~goldberg/jester-data and www.grouplens.org/node/73, respectively. Jester dataset has 73,421 users who entered numeric ratings for 100 jokes ranging on real value scale from −10 to +10. The MovieLens dataset consists of 100,000 ratings ranging in the discrete scale 1 to 5 from 943 users on 1,682 movies.

Benchmark algorithms: The experiments compare the proposed method CIT to some typical alternative prediction algorithms: pure Pearson predictor [1]; Pearson-like predictor (scale and translation invariant Pearson (STIPearson_2.0) [4], Non-personalized (NP) [26], Slope one-like predictor (Bi-Polar Slope One) [27], and compositions of adjust scale [4] + approach (Non personalized/Bi-Polar Slope One). The average predictor simulates the absence of a recommender and uses the target user's average rating as the predicting rating. It is only for providing a baseline for comparison purposes.

Methodology: For the purposes of comparison, a subset of the datasets is adopted.

MovieLens: The full u data set, 100,000 ratings by 943 users on 1,682 items.

MovieLens dataset 1: The data sets u1.base and u1.test through u5.base and u5.test are 80 %/20 % splits of the u data into training and test data. Each of u1, …, u5 has disjoint test sets. MovieLens dataset 2: The data sets ua.base, ua.test, ub.base, and ub.test split the u data into a training set and a test set with exactly 10 ratings per user in the test set. The sets ua.test and ub.test are disjoint. MovieLens 2 represents the sparse user-item rating matrix.

Jester dataset:

Jester dataset 1: Data from 24,983 users who have rated 36 or more jokes, a matrix with dimensions 24,983 × 101. Jester dataset 2: Data from 24,938 users who have rated between 15 and 35 jokes, a matrix with dimensions 24,938 × 101. Jester dataset 2 represents the sparse user-item rating matrix.

In every test round, for each user in the test sets, ratings for one item were withheld. Predictions will be recomputed for the withheld items using each of the different predictors. The quality of the various prediction predictors is measured by comparing the predicted values for the withheld ratings to the actual ratings.

Metrics: The type of metrics used depend on the type of CF applications. To measure predictive accuracy, we use the All But One MAE (AllBut1MAE) [2, 28] by withholding a single element *j* from ratings of target user to compute the MAE on the test set for each user and then average over the set of test users.

$$\text{AllBut1MAE} = \frac{1}{n}\sum\nolimits_{u,j} |p_{u,j} - r_{u,j}| \tag{10}$$

where n is the total number of ratings over all users, $p_{u,\,j}$ and $r_{u,\,j}$ are the predicted rating and actual rating of u on withheld item j respectively. The lower the AllBut1MAE, the better the prediction.

To measure decision-support accuracy, the experiments use receiver operating characteristic (ROC) sensitivity and specificity. Sensitivity [true positive rate (TPR)] is defined as the probability that a good item is accepted by the filter; and specificity (SPC) (1 − false positive rate) is defined as the probability that a bad item is rejected by the filter. For the confusion matrix [29], it has

$$\begin{aligned} \mathrm{TPR} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \\ \mathrm{SPC} &= 1-\frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}=\frac{\mathrm{TN}}{\mathrm{FP}+\mathrm{TN}} \end{aligned} \tag{11}$$

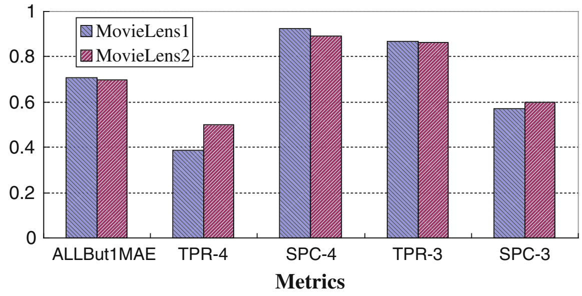
The higher the TPR and the SPC, the better the prediction. By tuning a threshold value, all the items ranked above it are deemed observed by the user, and below unobserved, thus the system will get different prediction values for different threshold values. In Jester dataset, the experiments consider an item good if the user gives it a rating greater than or equal to 4, 6, 8, or above, otherwise they consider the item bad. In MovieLens dataset, the experiments consider an item good if the user gave it a rating greater than or equal to 3, 4, or above, otherwise they consider the item bad. ROC Sensitivity and specificity ranges from 0 to 1.

3.2 *Experimental Results for Data Sparsity in User-Item Rating Matrix*

The data sparsity in user-item rating matrix results in sparseness of user similarity matrix. CIT and all compared prediction algorithms are implemented in java on Microsoft windows operating system by modifying the Cofi: A Java-Based Collaborative Filtering Library.

The performance of our methods: The test results (Figs. 2, 3) indicate that CIT (our method) performs well when facing two kinds of dataset under three kinds of metrics. The AllBut1MAE of CIT slightly decreases while the TPR (ROC sensitivity) clearly increases when facing the Jester dataset 2 and MovieLens dataset 2 that represent the sparse user-item rating matrix. The metrics scoring of CIT shows

Fig. 2 Metrics scoring of CIT for MovieLens dataset



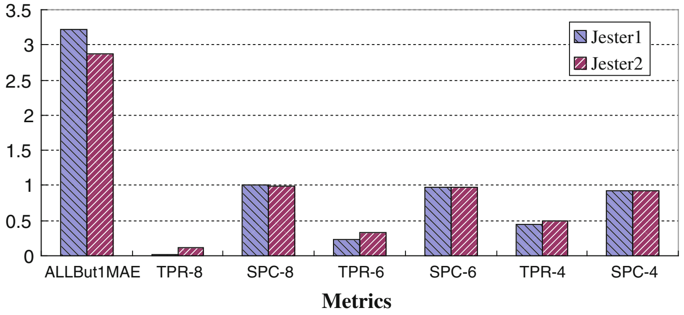


Fig. 3 Metrics scoring of CIT for Jester dataset

Table 1 Metrics scoring of all predictor compared for Jester dataset 1

Predictors	MAE	TPR-8	SPC-8	TPR-6	SPC-6	TPR-4	SPC-4
CIT	3.2177	0.0116	0.9993	0.2263	0.9783	0.4517	0.9251
Pearson	3.4138	0.0077	0.9993	0.1735	0.9800	0.3744	0.9344
STIPearson_2.0	3.3362	0.0886	0.9927	0.2972	0.9650	0.4635	0.9152
STINP_2.0	3.4625	0.0077	0.9989	0.1496	0.9803	0.3608	0.9356
Bi-PSO	3.4481	0.0039	0.9989	0.2196	0.9745	0.4790	0.9015
AS_Bi-PSO	3.4586	0.0039	0.9993	0.1323	0.9838	0.3620	0.9365
NP	3.5306	0.0058	0.9978	0.1275	0.9833	0.3558	0.9298
AS_NP	3.4614	0.0116	0.9987	0.1477	0.9795	0.3608	0.9339
Average	3.7785	0.0000	1.0000	0.0364	0.9935	0.2779	0.9266

that it is insensitive to dataset and is adaptive to sparse user-item rating matrix. The test results also show that TPR increases and SPC (ROC specificity) declines as the threshold of ROC become smaller. Meanwhile, the smaller the threshold of ROC, the bigger the TPR; the smaller the threshold of ROC, the smaller the SPC.

Compare with other methods: Tables 1, 2 and Fig. 4 show experimental results for Jester dataset while Tables 3, 4 and Fig. 5 show experimental results for MovieLens dataset. Overall, any compared predictors with metrics scoring outperform base line average. CIT is very successful and outperforms all other predictors compared. Especially CIT outperforms significantly all other compared predictors when meeting with Jester dataset 2 and MovieLens dataset 2 that represent the sparse user-item rating matrix.

For Jester dataset, the Pearson, STIPearson_2.0, NP, (Bi-Polar Slope One (Bi-PSO), and adjust scale + Bi-PSO (AS_Bi-PSO), adjust scale + NP(AS_NP) perform near-similarly, among which STIPearson_2.0 and AS_NP are more impressive than others. Furthermore, the performance of these compared predictors becomes slightly worse for Jester dataset 2.

For MovieLens dataset, Pearson, STIPearson_2.0, NP, Bi-PSO, AS_Bi-PSO, and AS_NP perform similarly, among which Bi-PSO and NP are more impressive than others. Furthermore, the performance of these compared predictors becomes

Table 2 Metrics scoring of all predictor compared for Jester dataset 2

Predictors	MAE	TPR-8	SPC-8	TPR-6	SPC-6	TPR-4	SPC-4
CIT	2.8674	0.1234	0.9951	0.3329	0.9754	0.5022	0.9311
Pearson	3.5507	0.0127	1.0000	0.1163	0.9878	0.3345	0.9416
STIPearson_2.0	3.4586	0.0823	0.9964	0.2475	0.9744	0.4396	0.9185
STINP_2.0	3.4996	0.0538	0.9959	0.1696	0.9801	0.3612	0.9405
Bi-PSO	3.5271	0.0000	0.9998	0.1003	0.9919	0.3367	0.9463
AS_Bi-PSO	3.5464	0.0506	0.9979	0.1498	0.9830	0.3568	0.9458
NP	3.5652	0.0095	0.9991	0.0891	0.9926	0.2928	0.9475
AS_NP	3.5000	0.0538	0.9955	0.1696	0.9809	0.3648	0.9408
Average	3.9317	0.0000	1.0000	0.0198	0.9981	0.1115	0.9694

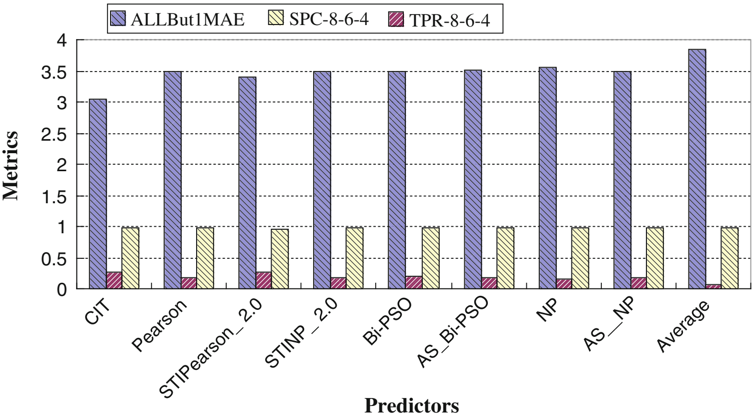


Fig. 4 Average metrics scoring of all schemes compared for Jester dataset: The lower the AllBut1MAE, the better the prediction accuracy is; while the higher the TPR and the SPC, the better the decision-support accuracy is

Table 3 Metrics scoring of all predictor compared for MovieLens dataset 1

Predictors	MAE	TPR-4	SPC-4	TPR-3	SPC-3
CIT	0.7077	0.3867	0.9227	0.8670	0.5722
Pearson	0.7687	0.3218	0.9125	0.8707	0.4783
STIPearson_2.0	0.7448	0.4042	0.8937	0.8517	0.5485
STINP_2.0	0.7418	0.3457	0.9190	0.8722	0.5206
Bi-PSO	0.7490	0.5585	0.8109	0.9241	0.3958
AS_Bi-PSO	0.7363	0.3437	0.9215	0.8780	0.5204
NP	0.7561	0.3580	0.9067	0.8572	0.5219
AS_NP	0.7417	0.3446	0.9175	0.8729	0.5175
Average	0.8409	0.1884	0.9314	0.9196	0.2509

worse obviously for MovieLens dataset 2. In addition, TPR of Bi-PSO is above all other predictors but SPC of Bi-PSO is below all other predictors. All methods except CIT are sensitive to the density of dataset.

Table 4 Metrics scoring of all predictor compared for MovieLens dataset 2

Predictors	MAE	TPR-4	SPC-4	TPR-3	SPC-3
CIT	0.6969	0.5014	0.8913	0.8620	0.5997
Pearson	0.8407	0.3649	0.8398	0.8713	0.3615
STIPearson_2.0	0.8370	0.3911	0.8332	0.8675	0.3810
STINP_2.0	0.8246	0.4136	0.8435	0.8495	0.4454
Bi-PSO	0.8125	0.5742	0.7327	0.9063	0.3510
AS_Bi-PSO	0.8769	0.3647	0.8203	0.8514	0.3651
NP	0.8112	0.4213	0.8421	0.8542	0.4449
AS_NP	0.8215	0.4169	0.8411	0.8517	0.4448
Average	0.8736	0.3326	0.8288	0.9079	0.2504

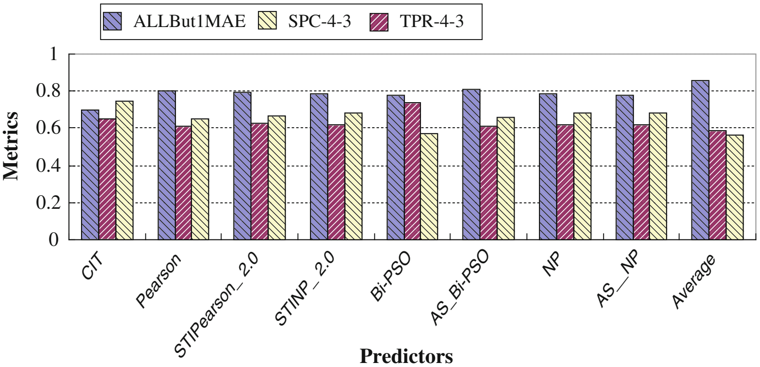


Fig. 5 Average metrics scoring of all schemes compared for MovieLens dataset: The lower the AllBut1MAE, the better the prediction accuracy is; while the higher the TPR and SPC, the better the decision-support accuracy is

As a result, we have evidence that CIT is insensitive to dataset and more adaptive to sparse user-item rating matrix than all other compared predictors. CIT can effectively relieve the data sparsity problem. The reason is that CIT combines the interest and trust to create the user similarity.

3.3 Experimental Results for New User

New user (cold start) problem occurs when recommendations are to be generated for user who has not rated any item or rated very few items since it is difficult for the system to compare her/him with other users and find possible neighborhood. This problem can be solved through “heavy” users. The list of default trusted friends can be captured and automatically added to the new user’s friend list at initial stage.

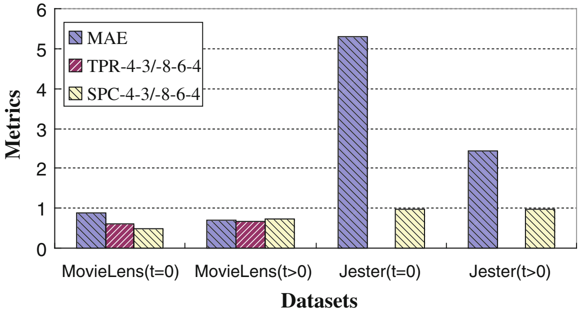


Fig. 6 Average metrics scoring of the CIT approach on MovieLens and Jester datasets

Figure 6 shows that results of CIT approach on MovieLens and Jester Datasets at time $t = 0$ and $t > 0$. Figures 7 and 8 show experimental results of all approaches compared on MovieLens and Jester datasets. It can be observed that the predication on MovieLens dataset is more accurate than on Jester Dataset. This is because the rating scale of MovieLens dataset is smaller than that of Jester datasets.

As a result, we have evidence that CIT is slightly sensitive to dataset for new user. The reason is that the larger the ranging rating scale, the more difficult the prediction. CIT combines the taste similarity and referral trust to compute the user similarity and adopts adjust scale modified Resnicks prediction formula to predict rating. On one hand, CIT incorporates the advantages of STI Pearson and Bi-PSO. On the other hand, feedback-based referral trust considers that target user's feedback is valuable to continually update the similarity between users. Thus, the proposed approach can be used as part of a more broad recommendation explanation which not only improves user satisfaction but also helps users make better decisions.

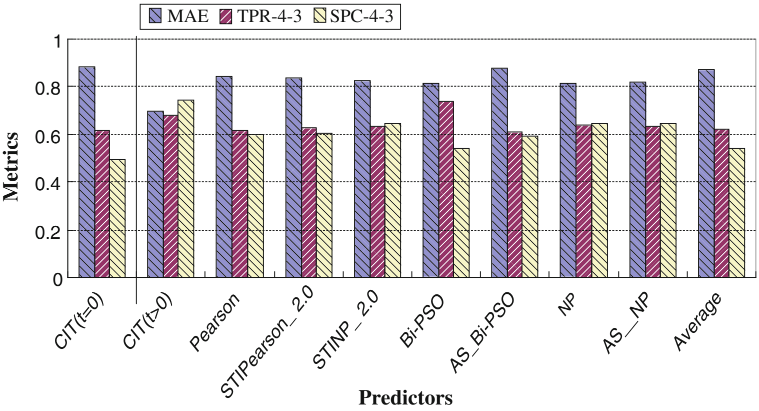


Fig. 7 Average metrics scoring of all schemes compared for MovieLens dataset

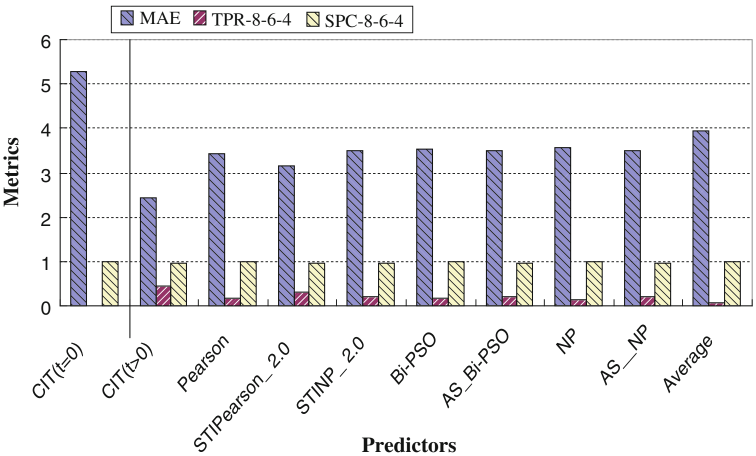


Fig. 8 Average metrics scoring of all schemes compared for Jester dataset

4 Conclusion

Traditionally, CF systems have relied heavily on similarities between the ratings profiles of users as a way to differentially rate the prediction contributions of different profiles. In the parse data environment, where the predicted recommendations often depend on a small portion of the selected neighborhood, ratings similarity on its own may not be sufficient, therefore other factors might also have an important role to play. In the proposed system, sparseness in user similarity due to data sparsity of input matrix is reduced by incorporating trust. Also, new users highly benefit from this wisdom of crowds as positive feedback in the form of aggregated global trust and global interest, and defines heavy user as default recommender over a period of time. The proposed approach is used as an explanation for recommendation by providing additional information along with the predicted rating regarding the trust intensity between target user and recommendation partiers. This not only improves user's satisfaction degree, but also helps users make better decisions as it acts as an indication of confidence of the predicted rating.

Acknowledgments This work is supported in part by the National Key Technology R&D Program (No. 2012BAH16F02), the Natural Science Foundation of China (Grant No. 61003254 and No. 60903038), and the Fundamental Research Funds for the Central Universities.

References

1. Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J (1994) GroupLens: an open architecture for collaborative filtering of netnews. In: ACM CSCW'94 conference on computer-supported cooperative work, sharing information and creating meaning, pp 175–186

2. Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering. In: 14th conference on uncertainty in artificial intelligence (UAI'98). Morgan Kaufmann, San Francisco, pp 43–52
3. Shardanand U, Maes P (1995) Social information filtering: algorithms for automating word of mouth. In: Proceedings of the SIGCHI conference on human factors in computing systems, Denver, CO, USA, pp 210–217
4. Lemire D (2005) Scale and translation in variant collaborative filtering systems. Inf Retr 8(1):129–150
5. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: 10th international conference on World Wide Web, pp 285–295
6. Deshpande M, Karypis G (2004) Item-based top-N recommendation algorithms. ACM Trans Inf Syst 22(1):143–177
7. Linden G, Smith B, York J (2003) Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput 7(1):76–80
8. Schein AI, Popescul A, Ungar LH (2002) Methods and metrics for cold-start recommendations. In: the 25th annual international ACM SIGIR conference on research and development in information retrieval, pp 253–260
9. Massa P, Avesani P (2004) Trust-aware collaborative filtering for recommender systems. In: International conference on cooperative information systems, Agia Napa, Cyprus, pp 492—508
10. Kim HN, Ji AT, Ha I, Jo GS (2010) Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation. Electron Commerce Res Appl 9(1):73–83
11. Park YJ, Chang KN (2009) Individual and group behavior-based customer role model for personalized product recommendation. Expert Syst Appl 36(2):1932–1939
12. Balabanovic M, Shoham Y (1998) Content-based, collaborative recommendation. Commun ACM 40(3):66–72
13. Lee TQ, Park Y, Park YT (2008) A time-based approach to effective recommender systems using implicit feedback. Expert Syst Appl 34(4):3055–3062
14. Huang CL, Huang WL (2009) Handling sequential pattern decay: developing a two stage collaborative recommendation system. Electron Commerce Res Appl 8(3):117–129
15. Jeong B, Lee J, Cho H (2009) An iterative semi-explicit rating method for building collaborative recommender systems. Expert Syst Appl 36(3):6181–6186
16. Lee JS, Olafsson S (2009) Two-way cooperative prediction for collaborative filtering recommendations. Expert Syst Appl 36(3):5353–5361
17. O'Donovan J, Smyth B (2005) Trust in recommender systems. In: IUI'05, SanDiego, CA, USA, pp 167–174
18. Massa P, Bhattacharjee B (2004) Using trust in recommender systems: an experiment analysis. In: 2nd international conference on trust management, Oxford, England, pp 221–235
19. Massa P, Avesani P (2007) Trust-aware recommender systems. In: RecSys. ACM, New York, NY, USA, pp 17–24
20. Massa P, Avesani P (2009) Trust metrics in recommender systems. In: Golbeck J (ed) Computing with social trust. Springer, London, pp 259–285
21. Yuan W, Guan D, Lee YK, Lee S, Hur SJ (2010) Improved trust-aware recommender system using small-worldness of trust networks. Knowl-Based Syst 23(3):232–238
22. Bedi P, Sharma R (2012) Trust based recommender system using ant colony for trust computation. Expert Syst Appl 39(1):1183–1190
23. Claypool M, Le P, Wased M, Brown D (2001) Implicit interest indicators. In: IUI '01, pp 33–40
24. Kuo YL, Yeh CH, Chau R (2003) A validation procedure for fuzzy multi-attribute decision making. In: The 12th IEEE international conference on fuzzy systems, vol 2, pp 1080–1085
25. Kim HK, Kim JK, Ryu YU (2009) Personalized recommendation over a customer network for ubiquitous shopping. IEEE Trans Serv Comput 2(2):140–151

26. Herlocker J, Konstan J, Borchers A, Riedl J (1999) An algorithmic framework for performing collaborative filtering. In: Research and development in information retrieval
27. Lemire D, Maclachlan A (2005) Slope one predictors for online rating-based collaborative filtering. In: SIAM data mining (SDM'05), Newport Beach, CA
28. Canny J (2002) Collaborative filtering with privacy via factor analysis. In: the Special inspector general for Iraq reconstruction
29. Receiver Operating Characteristic (2012). http://en.wikipedia.org/wiki/Receiver_operating_characteristic