Int J Data Sci Anal (2017) 3:297–307
DOI 10.1007/s41060-017-0049-y

REGULAR PAPER

Collaborative filtering-based recommender systems by effective trust

Vahid Faridani[1] · Mehrdad Jalali[1] · Majid Vafaei Jahan[1]

Received: 15 August 2016 / Accepted: 28 February 2017 / Published online: 15 March 2017

Abstract Collaborative filtering (CF) is one of the most well-known and commonly used techniques to build recommender systems and generate recommendations. However, it suffers from several inherent issues such as *data sparsity* and *cold start*. This paper tends to describe the steps based on which the ratings of an active users trusted neighbors are combined to complement and represent the preferences to the active user. First, by discriminating between different users, we calculate the significance of each user to make recommendations. Then, the trusted neighbors of the active user are identified and aggregated. Hence, a new rating profile can be established to represent the preferences of the active user. In the next step, similar users probed based on the new rating profile. Finally, recommendations are generated in the same way as the conventional CF with the difference that if a similar neighbor had not rated the target item, we will predict the value of the target item for this similar neighbor by using the ratings of her directly trusted neighbors and applying MoleTrust algorithm, to combine more similar users to generate a prediction for this target item. Experimental results demonstrate that our method outperforms other counterparts both in terms of accuracy and in terms of coverage.

Keywords Recommender systems · Collaborative filtering · Cold start · Data sparsity · Trusted neighbors

✉ Mehrdad Jalali
jalali@mshdiau.ac.ir

Vahid Faridani
vahid.faridani@mshdiau.ac.ir

Majid Vafaei Jahan
vafaeijahan@mshdiau.ac.ir

[1] Department of Computer, Mashhad Branch, Islamic Azad University, Mashhad, Iran

1 Introduction

With the rapidly growing amount of information available on the WWW, it becomes necessary to have tools to help users to select the relevant part of online information. Collaborative filtering, as the most popular approach in establishing recommender systems, has been successfully applied for a wide range of applications. However, it suffers from several inherent issues such as *data sparsity* and *cold start*. To resolve these issues and model user preferences more accurately, additional information from other sources is studied and combined into CF, including friendship [1], membership [2,3] and social trust [4,5]. In this paper, trust is defined as ones belief toward others in providing accurate ratings relative to the preferences of the active user. Both implicit trust (e.g., [6,7]) and explicit trust (e.g., [8–11]) have been investigated in the literature. The former trust is inferred from user behaviors such as ratings, whereas the latter is directly specified by users. By definition, the explicit trust tends to be more accurate and reliable than the implicit one. We focus on the explicit trust in this paper.

In this paper, we propose a novel trust-based approach called Effective Trust by incorporating the trusted neighbors explicitly specified by the active users in the systems, aiming to improve the overall performance of recommendations and to reduce the data sparsity and cold-start problems of CF., Specifically, we calculate the significance of each user to make recommendations. Then, we combine the ratings of trusted neighbors of an active user by averaging the ratings on the commonly rated items according to the extent to which the trusted neighbors are similar to the active user and also according to the extent to which the trusted neighbors are significant to make recommendations. The set of new ratings is subsequently used to represent the active users preferences and to find similar users based on user similarity. Finally, recommendations are generated in the same way as the conventional CF with the difference that if a similar neighbor had not rated the target item, we will predict the value of the target item for this similar neighbor by using the ratings of her directly trusted neighbors and applying MoleTrust algorithm, so as to combine more similar users to generate a prediction for this target item. Experiments on Flixster data set are conducted to demonstrate the effectiveness of our method in terms of accuracy and coverage. The results confirm that our method achieves promising recommendation performance, especially effective for the users comparing with the other counterparts.

The rest of the paper is organized as follows: Section 2 gives a brief overview of related research on trust-based CF from which the research gap is identified and motivating our present work. The proposed approach is then elaborated in Sect. 3 where we also highlight the advantages of our method in principle. Experiments on Flixster data set are conducted in Sect. 4 to verify the effectiveness of our method in predicting items ratings, especially for the users. Finally, Sect. 5 concludes our work and outlines potential future research.

2 Related work

Many CF approaches have been proposed in the literature to resolve the data sparsity and cold-start problems. Generally, they can be classified under two categories: memory based and model based. The most well-known models that are based on matrix factorization (MF) approaches are SVD [12], NNMF [13] and tensor factorization [14]. Model-based approaches usually can achieve better accuracy and coverage than memory-based approaches. This is because the former ones will train a prediction model using global rating data, whereas the latter concentrate on local rating information. However, model-based approaches cannot properly explain how the recommendations are generated and effectively adopt new ratings due to trained static models.

Trust information can be explicitly collected from users or implicitly inferred from users rating information. The former trust is specified directly by users themselves. Typical applications are FilmTrust[1] and Epinions.com where each user can specify others as trustworthy or untrustworthy. In contrast, implicit trust is usually inferred from user behaviors, such as ratings.

The closest approaches to ours are as follows. Massa et al. [10] analyze the drawbacks of conventional CF-based recommender systems and elaborate the rationale why incorporating trust can mitigate those problems. They propose the MoleTrust algorithm, which performs the depth-first search, to propagate and infer trust in the trust networks. Empirical results show that the coverage is significantly enlarged, but the accuracy remains comparable when propagating trust. Similarly, Golbeck et al. [9] proposed a breadth-first search method called TidalTrust to infer and compute trust value. Both approaches substitute similarity with trust to predict item ratings, and the performance of the two algorithms is close [15]. Hence, we will only compare our method with one of them, namely MoleTrust in this paper. In addition, Chowdhury et al. [8] proposed to enhance CF by predicting the ratings of similar users who did not rate the target items according to the ratings of their trusted neighbors, so as to combine more similar users for recommendation. However, it performs badly in conditions where only few ratings are available, which is the main concern of the present work. Another recent work using the explicit trust network is proposed by [11]. They improve the prediction accuracy by reconstructing the trust networks. More specifically, the trust links between two users will be removed if their similarity is lower than a threshold. Empirical results show that good performance was achieved in the cost of poor coverage, and it failed to perform its function in conditions in which the user similarity may not be computed.

The purpose of our work is to take a step further in addressing the cold-start and sparsity problems by proposing a novel approach to combine trusted neighbors in CF.

3 The effective trust method

In this section, we will present the proposed Effective Trust method the basic principle of which is to combine the ratings of trusted neighbors to complement and represent the preferences of active users. Four steps are taken to make recommendations. The architecture of the proposed method is shown in Fig. 1. First, by discriminating between different users, we calculate the significance of each user to make recommendations. Second, the ratings of trusted neighbors are combined into a single value for each item that is rated by at least one trusted neighbor. Hence, a new rating profile can be formed to represent the preferences of the active user. Trust propagation may be needed to combine more trusted neighbors, especially useful for the users that affected by the cold start problem. Third, similar users probed based on the new rating profile. Fourth, recommendations are generated in the same way as the conventional CF with difference that if a similar neighbor had not rated the target item, we will predict the value of the target item for this similar neighbor by using the ratings of her directly trusted neighbors and applying MoleTrust algorithm to combine more similar users to generate prediction for this target item. Detailed descriptions as well as the insights of the Effective Trust method are given in the subsequent sections.

[1] trust.mindswap.org/FilmTrust/.

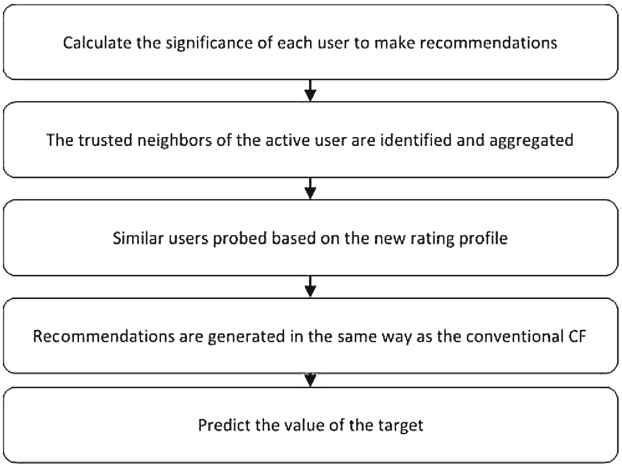
Calculate the significance of each user to make recommendations
The trusted neighbors of the active user are identified and aggregated
Similar users probed based on the new rating profile
Recommendations are generated in the same way as the conventional CF
Predict the value of the target

Fig. 1 Model proposed approach

3.1 Effective trust process

For clarity, we introduce a number of notations to model the recommendation problem. Specifically, we denote the sets of all users, all items and all ratings as U, UI and R, respectively. We keep the symbols u, v for the users and i, j for the items. Then, $r_{u,i}$ represents a rating given by user u on item i and takes a value in a certain rating scope, such as an integer from 1 to 5, predefined by a recommender system. Hence, the task of a recommender can be modeled as: given a set of user-item-rating (u,i,$r_{u,i}$) triplets, provide a best prediction (u,j,?) for user u on an unknown item j. The predicted rating is denoted as $\hat{r}_{u,j}$. In a trust-aware recommender system, the active user u may have identified a set of trusted neighbors TN_u. For each trusted neighbor $v \in \mathrm{TN}_u$, user u also specifies a trust value $t_{u,v} \in [0, 1]$ indicating the extent to which user u believes in user vs ability in giving accurate ratings. For simplicity, the total numbers of items in the catalog are denoted by $I_u = \{i|r_{u,i} \in R, i \in I\}$ and the set of users who rated item i is denoted by $U_i = \{u|r_{u,i} \in R, u \in U\}$. Hence, the recommendation problem can be re-described as: given a set of user ratings $(u, i, r_{u,i})$ and a set of user trust $(u, v, t_{u,v})$, predict a best prediction $(u, j, \hat{u}_j)$ for an active user u on a target item j. We are most concerned with the predictive accuracy of the predicted ratings (relative to the real preferences) and the percentage of target items that can be predicted.

3.1.1 *Calculation the significance of each user*

Guo et al. [16] have shown that some users are more significant than other users to make recommendations. Suppose $W = \{1, \ldots, 5\}$ be the set of possible values for ratings. Let $V = \{4, 5\}$ be the subset of W with elements that are regarded as relevant ratings. Let $V^c = 1, 2, 3$ be the subset of W with elements that are regarded as non-relevant ratings. Let $D_u = \{i \in I| \in r_{u,i} \in V\}$ be the set of items that user u has rated with a relevant value. Let $E_u = \{i \in I| \in r_{u,i} \in V^C\}$ be the set of items that user u has rated with a non-relevant value. To weight the importance of a rating, we use the following factors:

$$f_1 = 1 - \frac{\#D_u}{\#D_u + \#E_u} \tag{1}$$

$$f_2 = \frac{\#D_u + \#E_u}{\#I_u} \tag{2}$$

According to factor f_1, the lower the number of relevant ratings made by user u, the higher the significance of the users u relevant ratings. According to the factor f_2, the higher the number of ratings made by user u, the higher the significance of user u to make recommendations. Finally, we define the significance of a user to make recommendations as:

$$s_u = \left(1 - \frac{\#D_u}{\#D_u + \#E_u}\right)\left(\frac{\#D_u + \#E_u}{\#I_u}\right) \tag{3}$$

where $s_u \in [0, 1]$ is the significance of user u to make recommendations. Equation (3) provides a measure of the significance of the user u to make recommendations. This measure was calculated instead of the following:

- The ratio of the relevant ratings made by the user u to the number of ratings made by u.
- The ratio of the number of ratings made by u to the number of possible ratings that a user may make (that is to say, the number of items).

3.1.2 *Aggregating trusted neighbors*

The few rated users (rfu) are generally defined as the users who have rated less than five items [10]. Hence, to better model user preference, additional information is adopted. In our case, users social trust information is utilized since users in the systems can specify other users as trusted neighbors. Since few rated users usually are less active in the systems, they may not have a large number of trusted neighbors. We conduct experiments to show the statistics for these users in Flixster data set, the specifications of which will be presented in Sect. 4.1. Figure 2 shows the distribution of trusted neighbors for these users in Flixster data set. Specifically, most few rated users have only few trusted neighbors, and only few users have identified many trusted neighbors. Although social trust can be regarded as a (strongly and positively) additional information source to model user preference, the availability of trust information for few rated users is relatively limited.

Fortunately, trust can be propagated along with the web-of-trust. That is, if user A trusts B and B trusts C, it can

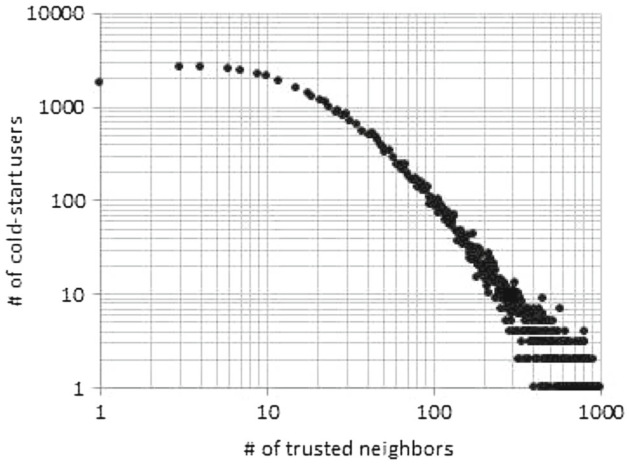

Fig. 2 Distributions of trusted neighbors for the few rated users on Flixster data set

be inferred that user A trusts C to some extent. MoleTrust [10] and TidalTrust [9] are two typical algorithms to infer trust value. To better use trust information, it is necessary to propagate trust in order to find more (indirectly) trusted neighbors. In this work, we adopt the MoleTrust to infer the trust value of indirectly connected users. Note that the trust value in the data set is binary, i.e., 0 or 1, where 0 means no direct trust connections, whereas 1 indicates that a user directly connects with and trusts another user. As a result, the inferred trust value by the MoleTrust will be also binary, and thus, we cannot distinguish trusted neighbors in a shorter distance with those in a longer distance. This issue may deteriorate the performance of trust-based approaches. Hence, like [16] we adopt a weighting factor to devalue trust in a long distance:

$$t_{u,v} = \frac{1}{d} \times t'_{u,v} \tag{4}$$

where $t'_{u,v}$ denotes the inferred trust value by the MoleTrust algorithm, d is the shortest distance between users u and v determined by a breath first search algorithm, and $t_{u,v} \in [0, 1]$ is the trust value that user u has toward another user v. In this way, directly specified trusted neighbors will be more trustworthy than the users in a long distance (but connected in the trust networks). Note that the greater d is, the more trusted neighbors will be inferred. However, the more cost will be taken and more noise is likely to be combined. In this work, we restrict $d \leq 3$[2] to prevent meaningless searching and save computational cost for large-scale data sets. In fact, as we will show later, the Effective Trust method works well enough when d is small.

Hence, a set of users can be identified as trusted neighborhood for user u if the trust value of a user v is greater than a trust threshold:

$$\mathrm{TN}_u = \{v | t_{u,v} > \theta_t \ , \ v \in U\} \tag{5}$$

where θ_t indicates the trust threshold. Since the distance is restricted by $d \leq 3$, we presume that the all connected trusted neighbors are useful and hence set $\theta_t = 0$ for simplicity. In addition, we presume that user u will always believe in her own ratings as they are accurately reflecting her real preferences.

3.1.3 Incorporating the ratings of trusted neighbors

After determining the trust neighborhood, a set of items can be identified as the candidate items for the Effective Trust process:

$$\tilde{I}_u = \{j | r_{v,j} \in R \ , \ v \in \mathrm{TN}_u \ , \ j \in I\} \tag{6}$$

That is, $\tilde{I}_u$ consists of items that have been rated by at least one trusted neighbor from the trust neighborhood. Then all the ratings of trusted neighbors on each item $j \in \tilde{I}_u$ will be combined into a single rating based on the weights of trusted neighbors:

$$\tilde{r}_{u,j} = \frac{\sum_{v \in \mathrm{TN}_u} w_{u,v} r_{v,j}}{\sum_{v \in \mathrm{TN}_u} |w_{u,v}|} \tag{7}$$

where $\tilde{r}_{u,j}$ indicates the combined value for user u on item $j \in \tilde{I}_u$ based on the ratings of all the trusted neighbors, and $w_{u,v}$ denotes the importance weight of user vs ratings relative to the active user u. We claim that the importance weight $w_{u,v}$ is composed of four parts: significance value s_v, trust value $t_{u,v}$, rating similarity $\mathrm{sim}_{u,v}$ and social similarity $j_{u,v}$. Hence, $w_{u,v}$ is computed as a linear combination of the four parts:

$$w_{u,v} = \alpha \mathrm{sim}_{u,v} + \beta t_{u,v} + \gamma s_v + (1 - \alpha - \beta - \gamma) j_{u,v} \tag{8}$$

where parameters α, β and γ indicate the extent to which the combination relies on rating similarity, trust value and significance value, respectively. The rationale behind this computation, i.e., incorporating four parts rather than trust value only, is that people trusting each other may not share similar preferences [17]. Specifically, it is possible that trusted neighbors have low similarity. According to the work conducted by [11], it is noted that trusted neighbors with high similarity have a positive influence on the predictive accuracy after eliminating those with low similarity. In addition, all users do not have equal significance to make recommendations [18]. Guo et al. [16] also noted that making

[2] The same setting is used in [8,10]. Better performance may be achieved by setting $d \leq 6$ and searching in a longer distance in the trust networks.

recommendations based on more significant users will result in good accuracy and good coverage. Therefore, it is necessary to consider both significance value, rating similarity and trust value.

Pearson correlation coefficient [19] is often used to compute user similarity based on ratings:

$$\mathrm{sim}_{u,v} = \frac{\sum_{i \in I_{u,v}} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{u,v}} (r_{u,i} - \bar{r}_u)^2}\sqrt{\sum_{i \in I_{u,v}} (r_{v,i} - \bar{r}_v)^2}} \tag{9}$$

where $\mathrm{sim}_{u,v} \in [0, 1]$ is the similarity between two users u and v, and $I_{u,v} = I_u \cap I_v$ denotes the set of items rated by both users u and v. In particular, $\mathrm{sim}_{u,v} > 0$ means positive correlation between users u and v, $\mathrm{sim}_{u,v} < 0$ indicates opposite correlation and $\mathrm{sim}_{u,v} = 0$ implies no correlation. Alterative similarity measures could be cosine similarity [19], Bayesian similarity [20], etc.

In addition, as indicated by [11] and as a general belief, even trusted users may not share similar preference and so does the social similarity. In other words, the trust and social similarity may be noisy and inaccurate. Considering the cases with positive trust, significance and social similarity but negative similarity may not make sense or be expected. Hence, we only consider the positively correlated users in this regard, i.e., $\mathrm{sim}_{u,v} > 0$,; another reason to this deals with its consistency with the value range of significance, trust and social similarity in (3), (4) and (10).

The fourth component is the ratio of commonly trusted neighbors between two users u and v; the intuition is that two users are socially close, if they share a number of trusted neighbors. Hence, a trusted neighbor who also shares some social friends will be regarded as more important than the user who has no friends in common with the active user. The social similarity is defined as the ratio of shared trusted neighbors over all the trusted neighbors and computed by the Jaccard index:

$$j_{u,v} = \frac{|\mathrm{TN}_u \cap \mathrm{TN}_v|}{|\mathrm{TN}_u \cup \mathrm{TN}_v|} \tag{10}$$

where $j_{u,v} \in [0, 1]$ indicates the social similarity of two users u and v based on their trusted neighbors. Hence, the importance weight $w_{u,v}$ can be computed using (8) since the four components are derived by (3), (4), (9) and (10), respectively. In this way, all the ratings of trusted neighbors on a certain item can be combined into a single value by (7).

Furthermore, since user u always gives accurate ratings from her own viewpoint, all her ratings will be retained and kept unchanged during the Effective Trust process as it is not necessary for them to be approximated (by the ratings of other trusted neighbors) in any way. Thus, we need to highlight that only the ratings of trusted neighbors on the other items that user u has not rated will be combined. To put it simply, the active user will keep all her own ratings, and the ratings of trusted neighbors will be used to complement her own preferences so that a new more complete and accurate rating profile can be formed and used to represent the preferences of the active user.

3.2 Incorporating with collaborative filtering

Given the new rating profile on the item set $\tilde{I_u}$ after the Effective Trust process in Sect. 3.1, which represents the preferences of the active user u, we then apply a conventional CF technique to predict the rating of a target item j that has not been rated by user u. More specifically, we first probe a set NN_u of similar users (i.e., nearest neighbors) for user u based on the similarity between user u and other users who either have rated item j or if they have not rated the target item j we can predict the rating of the target item j for them by using the ratings of their directly trusted neighbors and applying MoleTrust algorithm to combine more similar users to generate prediction for this target item j. Then the ratings of these nearest neighbors will be aggregated to produce a prediction for user u on item j.

In general, Pearson correlation coefficient (PCC) is often adopted to measure the similarity between two users u and v as $\mathrm{sim}'_{u,v}$ according to their ratings on the items that they commonly rated (see (9)):

$$\mathrm{sim}'_{u,v} = \frac{\sum_{i \in I_{u,v}} (\tilde{r}_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{u,v}} (\tilde{r}_{u,i} - \bar{r}_u)^2}\sqrt{\sum_{i \in I_{u,v}} (r_{v,i} - \bar{r}_v)^2}} \tag{11}$$

where $I_{u,v} = \tilde{I_u} \cap I_v$ denotes the set of items rated by both users u and v after Effective Trust process, $\bar{r}_u$ and $\bar{r}_v$ are the average ratings for users u and v, respectively.

After computing user similarity, a group of similar users are then selected into the nearest neighborhood NN_u of the active user u. Herein we use the thresholding method, i.e., adopting, the users whose similarity with the active user u, is greater than a predefined threshold:

$$\mathrm{NN}_u = \left\{ v \middle| \mathrm{sim}'_{u,v} > \theta_s\ ,\ v \in U \right\} \tag{12}$$

where θ_s indicates a predefined similarity threshold. An alternative method to determine the nearest neighborhood is well known as top K where the top K most similar users will be used. However, since in this work we focus on the performance of the few rated users, the top-K method is less effective to determine the nearest neighborhood than the thresholding method according to our experiments. Specifically, when we tune the values of K, no significant changes are observed in the performance of comparing methods. This may be due to the few similar users that can be identified based on the little rating information. Therefore, we use the thresholding rather than the top-K method to select nearest neighbors for the active users.

Finally, all the ratings of nearest neighbors are aggregated to produce a prediction on a target item j that the active user u has not rated [21,22]. We use the simple weighted average method, i.e., computing the average value of all ratings provided by the nearest neighbors v weighted by their similarity $\text{sim}'_{u,v}$ with the active user u. Formally, the prediction is computed by:

$$\hat{r}_{u,j} = \frac{\sum_{v \in \text{NN}_u} \text{sim}'_{u,v} r_{v,j}}{\sum_{v \in \text{NN}_u} |\text{sim}'_{u,v}|} \tag{13}$$

where $\hat{r}_{u,j}$ represents predicted value on item j. Hence, it ensures that the users with greater similarity will have more influence on predictions. We adopt the weighted average because the two most related works [10,11] also take the same equation.

3.3 An example

In this section, we intend to exemplify step-by-step application of the Effective Trust method to generate a prediction for a given item. Suppose there are nine users and nine items, denoted by u_k and i_j, respectively, where $k, j \in [1, 9]$ in a certain system. Each user may rate a few items by giving an integer rating ranged in [1,5] as given in Table 1. In addition, users may specify other users as trusted neighbors as given in Table 2, where an entry, for example, $(u_1, u_2, 1)$ indicates that user u_1 specifies user u_2 as a trusted neighbor. In this example, we are interested in generating a prediction on a target item i_5 (highlighted by the question mark) for an active user u_1. User u_1 has only reported a rating 5 on item i_3. She has indicated that users u_2 and u_3 as her trusted neighbors and both trusted users also pointed out others as trusted neighbors. By linking all the trusted neighbors together, we form a trust network for user u_1 as illustrated in Fig. 3. Specifically, users are represented as nodes, and the trust links are denoted as edges among users. Note that trust information is asymmetric, that is, users u_1 trusting u_2 does not imply users u_2 trusting u_1.

Table 1 Data set consisting of rating information

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	s_u
u_1			5		?					0
u_2	5		4		3			2		0.222
u_3		4		3				1		0.222
u_4	3		5		2					0.222
u_5		4	4		3			3		0.222
u_6		3	3	5	5					0.222
u_7							5		4	0
u_8			4		2			1		0.222
u_9			4		5			5		0

Table 2 Data set consisting of trust information

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9
u_1		1	1						
u_2			1	1					
u_3		1		1					
u_4					1				
u_5				1		1			
u_6			1	1					
u_7									
u_8									
u_9									

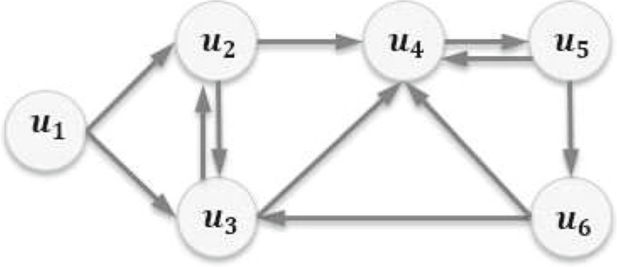

Fig. 3 Trust network

Table 3 Computed trust values between user u_1 and others

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9
d	1	1	1	2	3	4			
$t_{(u_1,u_k)}$	1.0	1.0	1.0	0.5	0.33	0.25			

Second, the trusted neighbors of the active user are identified by allowing trust propagation in the trust network. In Fig. 2, trust values between the active user u_1 and other users can be inferred by (4), and the results are presented in Table 3. In particular, as an active user, u_1 always trusts himself in giving accurate ratings and hence $t_{u_1,u_1} = 1$. Since users u_2 and u_3 are directly specified by user u_1, i.e., $d = 1$, their trust values will be 1.0. For user u_4, the minimum distance to user u_1 is 2, i.e., d=2. The shortest path of trust propagation is $u_1 \rightarrow u_2$ or $(u_3) \rightarrow u_4$, and the other path could be $u_1 \rightarrow u_2 \rightarrow u_3 \rightarrow u_4$. Hence, the trust value is computed by $t_{u_1,u_4} = 1/2 = 0.5$. The minimum distance from users u_1 to u_5 will be: $d(u_1, u_5) = d(u_1, u_4) + d(u_4, u_5) = 3$, and the distance to u_6 can be computed in the same manner. Note that this user will not be regarded as an inferred trusted neighbors due to the constraint $d \leq 3$, Hence, a set of users $TN_{u_1} = \{u_1, u_2, u_3, u_4, u_5\}$ is identified as trusted neighbors for active user u_1, although the trust value of user u_6 is computable.

Table 4 Combined rating profile for user u_1

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9
$\bar{r}(u_1, i_i)$	4.31	4	5	3	2.72			1.73	

Table 5 Computed similarity between user u_1 and others

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9
PCC	1.0	0.870	0.854	0.793	0.913	−0.887		0.975	−0.932

Third, the ratings of trusted neighbors will be combined using (7) and (8). For simplicity, in this example we set $\alpha = 0$, $\beta = 0.8$ and $\gamma = 0.2$ for (8), i.e., trust values and significance values are used as user weights. The resultant combined ratings are presented in Table 4. In particular, since user u_1 has rated item i_3, we presume the active user will always believe in her own ratings, and hence, there is no need to consider the ratings of trusted neighbors. Therefore, the combined rating on item i_3 is equal to $r_{u_1,i_3} = 5$ (i.e., 5). For other items that user u_1 has not rated, the ratings of trusted neighbors will be combined by (7). Take item i_1 as an instance. The ratings of users u_2 and u_4 will be averaged and weighted by their trust and significance values, i.e.,

$$\tilde{r}_{u_1,i_1} = \frac{5 \times (0.8 \times 1.0 + 0.2 \times 0.222) + 3 \times (0.8 \times 0.5 + 0.2 \times 0.222)}{(0.8 \times 1.0 + 0.2 \times 0.222) + (0.8 \times 0.5 + 0.2 \times 0.222)} = 4.31$$

This procedure continues until all the items rated by at least a trusted neighbor have been covered. A new rating profile is formed and given in Table 4. The combined rating profile is much more complete than the original. Fourth, user similarity is computed by (11) based on the formed rating profile (see Table 4). The results are given in Table 5. For consistency, the similarity between user u_1 and herself is 1.0. A set of users $\mathrm{NN}_{u_1} = \{u_2, u_3, u_4, u_5, u_8\}$ is selected as nearest neighbors, whose similarity is greater than the thresholds $\theta_s = 0$ and who either have rated the target item i_5 or if they have not rated the target item i_5, we can predict the value of the target item for them by using the ratings of their directly trusted neighbors and applying MoleTrust algorithm, so as to combine more similar users to generate prediction for this target item i_5. Note that user u_3 did not rate item i_5. The directly trusted neighbors of user u_3 are u_2 and u_4, so by applying MoleTrust algorithm we have:

$$r_{u_3,i_5} = \frac{1.0 \times 3 + 1.0 \times 2}{1.0 + 1.0} = 2.5$$

Finally, a prediction for item i_5 is generated by (13):

$$\tilde{r}_{u_1,i_5} = \frac{3 \times 0.870 + 2.5 \times 0.854 + 2 \times 0.793 + 3 \times 0.913 + 2 \times 0.975}{0.870 + 0.854 + 0.793 + 0.913 + 0.975} = 3.76$$

Compared with the values (2.72) shown in Table 4, the final prediction is different from the combined rating which is only based on trusted and significant neighbors since more ratings of similar users are used. In other words, generating a prediction only based on trusted and significant neighbors may not be reliable if only few trusted and significant neighbors can be identified. This is the situation for the few rated users. In contrast, by incorporating the ratings of trusted and significant neighbors, the ratings of similar users can be adopted to smooth the predictions.

Table 6 Specifications of Flixster data set

Dataset	#Users	#Items	#Rates	#Trust	Sparsity (%)
Flixster	2108	6356	50044	415883	99.63

4 Evaluation

In order to verify the effectiveness of the Effective Trust method, we conduct experiments on one real-world data set. Specifically, we aim to find out: (a) how the performance of our method in comparison with other counterparts; (b) what is the effect of trust propagation to our method and the others.

4.1 Data acquisition

One real-world data set is used in our experiments, namely Flixster[3], that contains both the data of explicit trust statements and user-item ratings. The specifications of data set are summarized in Table 6.

Flixster is a social movie site in which users are allowed to share their movie ratings, discover new movies and interact with others who have similar taste. We adopt the data set[4] collected by [23] which includes a large amount of data. The ratings are real values ranged from 0.5 to 5.0 with an interval 0.5, and the trust statements are scaled from 1 to 10 but not available. Hence, they are converted into binary values, that is, trust value 1 is assigned to a user who is identified as a trusted neighbor and 0 otherwise. Note that the trust statements in this data set are symmetric. We sample a subset by randomly choosing 2108 users who issued 50044

[3] http://www.flixster.com/.

[4] http://www.cs.sfu.ca/sja25/personal/datasets/.

item ratings and 415883 trust ratings. The rating sparsity is computed by:

$$\text{Sparsity} = \left(1 - \frac{\#\text{Ratings}}{\#\text{Users} \times \#\text{Items}}\right) \times 100\% \tag{14}$$

It is noted that the selected data set is highly sparse, i.e., users only rate a small portion of items in the system.

4.2 Experimental settings

In experiments, we compare the performance of our Effective Trust method with a number of trust-based state-of-the-art methods as well as a conventional user-based CF method.

- *CF* computes user similarity using the PCC measure, elects the users whose similarity is above the predefined similarity thresholds for (11), and uses their ratings to generate item predictions by (12). In this work, the threshold θ_S is set 0 for all methods.
- $MTx(x = 1, 2, 3)$ is the implementation of the MoleTrust algorithm [10] in which trust is propagated in the trust network with the length x. Only trusted neighbors are used to predict item ratings.
- *RN* denotes the approach proposed by [11] that predicts item ratings by reconstructing the trust networks. We adopt their best performance settings where the correlation threshold is 0.5, propagation length is 1, and the top 5 users with highest correlations are selected for rating predictions.
- $TCFx(x = 1, 2)$ denotes the approach proposed by [8] that enhances CF by predicting the ratings of the similar users who did not rate the items according to the ratings of the similar users trusted neighbors, so as to combine more users for recommendation. The best performance that they report is achieved when the prediction iteration x over trust network is 2. We adopt the same settings in our experiments.
- $UTx(x = 1, 2, 3)$ is our method with the trust propagation length x, tending to investigate the impact of trust propagation on the Effective Trust method.

In addition, we split each data set into two different views as defined in [10]: The view of **All Users** represents that all users and their ratings will be tested, whereas the view of **few rated Users** denotes that only these users who have rated less than five items, and their ratings will be tested in the experiments. In particular, we focus on the performance in the view of *few rated Users* which mostly indicates the effectiveness in mitigating the data sparsity and cold-start problems.

4.3 Evaluation metrics

The performance of all the methods is evaluated in terms of both accuracy and coverage. The evaluation is proceeding by applying the *leave-one-out* method on the two data views. In each data view, users ratings are hidden one by one in each iteration and then their values will be predicted by applying a certain method until all the testing ratings are covered. The errors between the predicated ratings and the ground truth are accumulated. The evaluation metrics are described as follows:

- Mean Absolute Error, or MAE, measures the degree to which a prediction is close to the ground truth:

$$\text{MAE} = \frac{\sum_u \sum_i |\hat{r}_{u,i} - r_{u,i}|}{N} \tag{15}$$

  where N indicates the number of testing ratings. Hence, the smaller the MAE value is, the closer a prediction is to the ground truth. Inspired by [24–26] who define a measure *precision* based on root-mean-square error (RMSE), we define the inverse MAE, or iMAE as the predictive accuracy normalized by the range of rating scales:

$$\text{iMAE} = 1 - \frac{\text{MAE}}{r_{\max} - r_{\min}} \tag{16}$$

  where $r_{\max}$ and $r_{\min}$ are the maximum and minimum rating scale defined by a recommender systems, respectively. Higher iMAE values indicate better predictive accuracy.
- Ratings Coverage, or RC, measures the degree to which the testing ratings can be predicted and covered relative to the whole testing ratings:

$$\text{RC} = \frac{M}{N} \tag{17}$$

  where M and N indicate the number of predictable and all the testing ratings, respectively.
- F-measure, or F1, measures the overall performance in considering both rating accuracy and coverage. Both accuracy and coverage are important measures for the predictive performance. According to [24], the F-measure is computed by:

$$\text{F1} = \frac{2.\text{iMAE.RC}}{\text{iMAE} + \text{RC}} \tag{18}$$

  Hence, the F-measure reflects the balance between accuracy and coverage.

Table 7 Predictive performance on the Flixster data set in the view of all users

View	Approaches measured by MAE, RC and F1									
	CF	MT1	MT2	MT3	RN	TCF1	TCF2	UT1	UT2	UT3
All	0.786	1.079	1.008	0.940	0.820	0.795	0.781	0.744	0.735	0.711
Users	35.53%	2.77%	20.54%	48.71%	0.30%	51.29%	62.59%	56.35%	64.84%	69.60%
	0.4967	0.0535	0.3284	0.6029	0.0061	0.6321	0.7122	0.6728	0.7266	0.7621

Table 8 Predictive performance on the Flixster data set in the view of few rated users

View	Approaches measured by MAE, RC and F1									
	CF	MT1	MT2	MT3	RN	TCF1	TCF2	UT1	UT2	UT3
Cold	0.813	1.162	0.998	0.964	0.890	0.822	0.829	0.815	0.813	0.808
Users	19.25%	2.29%	18.70%	46.59%	0.13%	29.08%	39.99%	33.67%	40.97%	47.63%
	0.3117	0.0445	0.3016	0.5850	0.0027	0.4290	0.5367	0.4772	0.5462	0.6027

Table 9 Improvements of all methods comparing with CF in F1

Dataset	Views	MTx (%)	RN (%)	TCFx (%)	UTx (%)
Flixster	All users	21.38	−98.77	43.39	53.43
	Few rated users	87.68	−99.13	72.18	93.36

4.4 Results and analysis

In this section, we conduct a series of experiments on one real-world data set to demonstrate the effectiveness of our approach relative to others and thus to answer the research questions proposed in Sect. 4. Both data set views, namely All Users, are tested. The results are presented in Tables 7 and 8 corresponding to the predictive performance on the Flixster.

4.4.1 Importance weights with parameters

An important step for the Effective Trust method is to compute the importance weights of trusted neighbors which is a linear combination of rating similarity, trust value, significance value and social similarity with parameters α, β and γ [see (8)]. The experiments show that the settings of (α, β, γ) are (0.4, 0.3, 0.2) on Flixster. As the best parameters are set by different value combinations across Flixster data set, we may conclude that similarity (0.4) is more important than trust value (0.3) which is superior to significance value (0.2) which is superior to social similarity in determining user preferences. Furthermore, it shows that rating, trust and significance information are useful and should be integrated to improve the recommendation performance.

4.4.2 Trust propagation in different lengths

An important factor for trust-based approaches is the use of trust transitivity. By propagating trust values through trust networks, more trusted neighbors can be identified, and hence, the performance of CF can be further improved. We investigate the influence of trust propagation on the performance of the Effective Trust method. Compared with UT1, UT2 and UT3 have a better accuracy and coverage. This may be explained by that the combined ratings will be more accurate. We may conclude that trust propagation is helpful to improve recommendation performance, and for our method, it shows that a short propagation length (i.e., 2) will be good enough to achieve a satisfying performance. This is because although more trusted neighbors can be identified via trust propagation, it does not guarantee that the combined rating profile will cover a lot more items and hence increase accuracy greatly. Rather, it is possibly that adding few trusted neighbors may result in some noisy combined ratings (due to few ratings) and hence harm the predictive performance.

To have a better view of the overall performance that each method achieves, we further compute the percentage of improvements that each method obtains comparing with the CF in terms of F1. Formally, it is computed by[5]:

$$\text{Improvement} = \frac{\text{Method.F1} - \text{CF.F1}}{\text{CF.F1}} \times 100\% \qquad (19)$$

where Method refers to any one of the methods tested in our experiments except the CF approach, whose F1 performance is regarded as a reference. Hence, the greater positive

[5] The formula can be referred to as the relative change defined in http://en.wikipedia.org/wiki/Relative_change_and_difference.

changes between Method and CF, the more improvements we obtain. The results are given in Table 4 both in the view of All Users and in the view of few rated Users. To explain, we take one value in Table 9 as an example, namely 53.43% for our method UTx. In the view of All Users of Flixster, the best Effective Trust method given in Table 7 is UT3 with F1 value 0.7621, while the F1 of CF is 0.4967. Hence, the improvement is $(0.7621-0.4967)/0.4967\times 100\% = 53.43\%$,; other values can be explained and verified as well. A conclusion that can be drawn from the results in Table 9 is that our method consistently outperforms the others (in term of improvement) and significantly improve the performance of traditional collaborative filtering.

4.4.3 Comparison with other methods

For other methods, we obtain different results on Flixster as given in Tables 7 and 8. More specifically, CF cannot achieve large portion of predictable items. It confirms that CF suffers from cold start severely. Unlike our imagination, the RN method accomplishes bad accuracy and also covers the smallest portion of items, since only the ratings of the users who have a large number of trusted neighbors and high rating correlations are possible to be predicted. Hence, RN is not comparative with others. Comparing with CF, all other methods except of MT1 and MT2 achieve better performance for few rated users. When only direct trusted neighbors are used (MT1, UT1), our method achieves better accuracy and coverage. When trust is propagated in longer length, both accuracy and coverage are increased. Nevertheless, our method outperforms MTx in all propagation lengths. TCF methods generally obtain better coverage in the view of All Users. However, for few rated users, TCF functions badly due to the limitation that it relies on CF to find similar users before it can apply trust information on them. As aforementioned, CF is not effective in few rated conditions. This fact leads to bad performance of TCF methods. In contrast, our method is not subject to the ratings of few rated users themselves. Instead, trust and significance information are combined to form a more concrete rating profile for the few rated users based on which CF is applied to find similar users and hence generate recommendations. Consistently, we come to a conclusion that the Effective Trust method outperforms the other approaches both in accuracy and in coverage as well as a better balance between them.

5 Conclusion and future work

This paper proposed a novel method to combine trusted neighbors into traditional collaborative filtering techniques, aiming to resolve the data sparsity and cold-start problems from which traditional recommender systems suffer. Specifically, the ratings of trusted neighbors were combined to complement and represent the preferences of the active users, based on which similar users can be identified and recommendations are generated. The prediction of a given item is generated by averaging the ratings of similar users weighted by their importance. Experiments on one real-world data set were conducted, and the results showed that significant improvements against other methods were obtained both in accuracy and in coverage as well as the overall performance. Further, by propagating trust in the trust networks, even better predictive performance can be achieved. In conclusion, we proposed a new way to better integrate trust, similarity and significance to improve the performance of collaborative filtering.

The present work depends on the explicit trust during the Effective Trust process. However, users may not be willing to share or expose such information due to the concerns of, for example, privacy. For future work, we intend to infer implicit trust from user behaviors and enhance the generality of the Effective Trust method.

References

1. Konstas, I., Stathopoulos, V., Jose, J.M.: In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval—SIGIR ’09, p. 195. ACM Press, New York (2009). doi:10.1145/1571941.1571977. http://portal.acm.org/citation.cfm?doid=1571941.1571977
2. Yuan, Q., Zhao, S., Chen, L., Liu, Y., Ding, S., Zhang, X., Zheng, W.: Augmenting Collaborative Recommender by Fusing Explicit Social Relationships. In: Workshop on Recommender Systems and the Social Web, Recsys, pp. 49–56 (2009)
3. Guy, I., Ronen, I., Wilcox, E.: In: Proceedings of the 13th International Conference on Intelligent User Interfaces—IUI ’09, p. 77. ACM Press, New York (2008). doi:10.1145/1502650.1502664. http://portal.acm.org/citation.cfm?doid=1502650.1502664
4. Ziegler, C.N., Lausen, G.: Analyzing Correlation Between Trust and User Similarity in Online Communities. Springer Berlin Heidelberg, pp. 251–265. (2004). doi:10.1007/978-3-540-24747-0_19
5. Jøsang, A., Quattrociocchi, W., Karabeg, D.: Taste and Trust, pp. 312–322. Springer, Berlin (2011). doi:10.1007/978-3-642-22200-9_25
6. O’Donovan, J., Smyth, B.: In: Proceedings of the 10th International Conference on Intelligent User Interfaces—IUI ’05, p. 167. ACM Press, New York (2005). doi:10.1145/1040830.1040870. http://portal.acm.org/citation.cfm?doid=1040830.1040870
7. Seth, A., Zhang, J., Cohen, R.: In: Proceedings of the 18th International Conference on User Modeling, Adaptation, and Personalization, pp. 279–290. Springer, Berlin (2010). doi:10.1007/978-3-642-13470-8_26
8. Chowdhury, M., Thomo, A., Wadge, W.W.: In: Chawla, S., Karlapalem, K., Pudi, V. (eds.) COMAD, Computer Society of India. (2009). http://dblp.uni-trier.de/db/conf/comad/comad2009.html#ChowdhuryTW09
9. Golbeck, J.A.: Computing and Applying Trust in Web-Based Social Networks. University of Maryland at College Park, College Park, USA (2005)

10. Massa, P., Avesani, P.: In: Proceedings of the 2007 ACM Conference on Recommender Systems—RecSys '07, p. 17. ACM Press, New York. (2007). doi:10.1145/1297231.1297235. http://portal.acm.org/citation.cfm?doid=1297231.1297235
11. Ray, S., Mahanti, A.: In: 2010 43rd Hawaii International Conference on System Sciences, pp. 1–9. IEEE (2010). doi:10.1109/HICSS.2010.225. http://ieeexplore.ieee.org/document/5428690/
12. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8):30–37 (2009). doi:10.1109/MC.2009.263. http://ieeexplore.ieee.org/document/5197422/
13. Zhang, S., Zhang, S., Wang, W., Ford, J., Makedon, F.: In: Proceedings of the 6th Siam Conference on Data Miningpp, pp. 549—-553. SDM (2006). http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.108.415
14. Shi, Y., Karatzoglou, A., Baltrunas, L., Larson, M., Hanjalic, A., Oliver, N.: In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval—SIGIR '12, p. 155. ACM Press, New York (2012). doi:10.1145/2348283.2348308. http://dl.acm.org/citation.cfm?doid=2348283.2348308
15. Victor, P., Cornelis, C., Cock, M.D., Teredesai, A.M.: Trust- and distrust-based recommendations for controversial reviews. IEEE Intell. Syst. **26**(1), 48–55 (2011). doi:10.1109/MIS.2011.22. http://ieeexplore.ieee.org/document/5714203/
16. Guo, G., Zhang, J., Thalmann, D.: Merging trust in collaborative filtering to alleviate data sparsity and cold start. Knowl. Based Syst. **57**, 57–68 (2014). doi:10.1016/j.knosys.2013.12.007
17. Singla, P., Richardson, M.: In: Proceeding of the 17th International Conference on World Wide Web—WWW '08, p. 655. ACM Press, New York (2008). doi:10.1145/1367497.1367586. http://portal.acm.org/citation.cfm?doid=1367497.1367586
18. Bobadilla, J., Hernando, A., Ortega, F., Gutiérrez, A.: Collaborative filtering based on significances. Inf. Sci. **185**(1), 1–17 (2012). doi:10.1016/j.ins.2011.09.014. http://linkinghub.elsevier.com/retrieve/pii/S0020025511004713
19. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005). doi:10.1109/TKDE.2005.99. http://ieeexplore.ieee.org/document/1423975/
20. Guo, G., Zhang, J., Yorke-Smith, N.: A Novel Bayesian Similarity Measure for Recommender Systems. In: 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013). August 3–9, 2013, Beijing, China (2013)
21. Jahan, M.V., Akbarzadeh-Totonchi, M.R.: From local search to global conclusions: migrating spin glass-based distributed portfolio selection. IEEE Trans. Evolut. Comput. **14**(4), 591–601 (2010). doi:10.1109/TEVC.2009.2034646. http://ieeexplore.ieee.org/document/5398877/
22. Mahboob, V.A., Jalali, M., Jahan, M.V., Barekati, P.: Swallow: resource and tag recommender system based on heat diffusion algorithm in social annotation systems. Comput. Intell. (2016). doi:10.1111/coin.12086
23. Jamali, M., Ester, M.: In: Proceedings of the Fourth ACM Conference on Recommender Systems—RecSys '10, p. 135. ACM Press, New York (2010). doi:10.1145/1864708.1864736. http://portal.acm.org/citation.cfm?doid=1864708.1864736
24. Jamali, M., Ester, M.: In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, pp. 397–406. ACM, New York (2009). doi:10.1145/1557019.1557067
25. Alizadeh, S.H., Arghami, N.R., Jahan, M.V.: In: 2015 International Congress on Technology, Communication and Knowledge (ICTCK), pp. 530–535. IEEE (2015). doi:10.1109/ICTCK.2015.7582724. http://ieeexplore.ieee.org/document/7582724/
26. Fahimi, M., Jahan, M.V., Torshiz, M.N.: In: 2014 International Congress on Technology, Communication and Knowledge (ICTCK), pp. 1–5. IEEE (2014). doi:10.1109/ICTCK.2014.7033517. http://ieeexplore.ieee.org/document/7033517/