



Merging trust in collaborative filtering to alleviate data sparsity and cold start



Guibing Guo^{*}, Jie Zhang, Daniel Thalmann

School of Computer Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

Article history:

Received 29 May 2013

Received in revised form 17 November 2013

Accepted 7 December 2013

Available online 12 December 2013

Keywords:

Recommender systems

Collaborative filtering

Cold start

Data sparsity

Trusted neighbors

ABSTRACT

Providing high quality recommendations is important for e-commerce systems to assist users in making effective selection decisions from a plethora of choices. Collaborative filtering is a widely accepted technique to generate recommendations based on the ratings of like-minded users. However, it suffers from several inherent issues such as *data sparsity* and *cold start*. To address these problems, we propose a novel method called “Merge” to incorporate social trust information (i.e., trusted neighbors explicitly specified by users) in providing recommendations. Specifically, **ratings of a user's trusted neighbors are merged to complement and represent the preferences of the user and to find other users with similar preferences** (i.e., similar users). In addition, the quality of merged ratings is measured by the confidence considering the number of ratings and the ratio of conflicts between positive and negative opinions. Further, the rating confidence is incorporated into the computation of user similarity. The prediction for a given item is generated by aggregating the ratings of similar users. Experimental results based on three real-world data sets demonstrate that our method outperforms other counterparts both in terms of accuracy and coverage.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

The emergence of Web 2.0 applications has greatly changed users' styles of online activities from searching and browsing to interacting and sharing [6,40]. The available choices grow up exponentially, and make it challenge for users to find useful information which is well-known as the *information overload* problem. Recommender systems are designed and heavily used in modern e-commerce applications to cope with this problem, i.e., to provide users with high quality, personalized recommendations, and to help them find items (e.g., books, movies, news, music, etc.) of interest from a plethora of available choices.

Collaborative filtering (CF) is one of the most well-known and commonly used techniques to generate recommendations [1,17]. The heuristic is that the items appreciated by those who have similar taste will also be in favor of by the active users (who desire recommendations). However, CF suffers from several inherent issues such as *data sparsity* and *cold start*. The former issue refers to the difficulty in finding sufficient and reliable similar users due to the fact that users in general only rate a small portion of items, while the latter refers to the dilemma that accurate recommenda-

tions are expected for the cold users who rate only a few items and thus whose preferences are hard to be inferred.

To resolve these issues and model user preferences more accurately, additional information from other sources is studied and incorporated into CF including friendship [19], membership [38,12] and social trust [41,2], where trust is believed less ambiguously and more reliable than friendship and membership. In this paper, trust is defined as one's belief toward others in providing accurate ratings relative to the preferences of the active user. Both implicit trust (e.g., [26,28]) and explicit trust (e.g., [4,8,25,27]) 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. Although many trust-based approaches have been proposed and the improvements to some extent have been achieved, there is still much room left for a better trust-based approach as stressed by [32].

In this paper, we propose a novel trust-based approach called “Merge” by incorporating the trusted neighbors explicitly specified by the active users in the systems, aiming to improve the overall performance of recommendations and to ameliorate the data sparsity and cold-start problems of CF. Specifically, we merge 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. The

^{*} Corresponding author. Tel.: +65 84005322.

E-mail addresses: gguo1@ntu.edu.sg (G. Guo), zhangj@ntu.edu.sg (J. Zhang), danielthalmann@ntu.edu.sg (D. Thalmann).

quality of the merged rating is measured by the confidence considering the number of ratings and the ratio of conflicts between positive and negative opinions (ratings). The set of merged ratings is then used to represent the active user's preferences and to find similar users based on user similarity. Further, the rating confidence is also taken into account in the computation of user similarity. Finally, the Merge method is incorporated into a conventional CF to generate recommendations. Experiments on three real-world data sets 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 cold users comparing with the other counterparts. Although the idea of incorporating trust information into recommender systems is not new, our paper is the first work to effectively complement user rating profiles based on the ratings of trusted neighbors. Hence, our method shades light on a new way to build an effective trust-aware recommender system. A preliminary version of our work was published at the UMAP'12 conference [10].

The rest of this 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 Section 3 where we also highlight the advantages of our method in principle. Experiments on three real-world data sets are conducted in Section 4 to verify the effectiveness of our method in predicting items' ratings, especially for the cold users. Finally, Section 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 into two categories: memory-based and model-based. The most well-known model is matrix factorization (MF) based approaches, such as SVD [20], NNMF [39], tensor factorization [31]. 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. Although these problems might be mitigated to some extent such as [9,7], the real applications (e.g., citeulike.com and last.fm) usually adopt simpler memory-based approaches. Further, a lesson learned from the Netflix competition is that no single (memory- or model-based) approach can always achieve the best performance, and different methods generally reveal different patterns of rating data [3]. Hence, it is necessary to further improve the performance of memory-based approaches.

To better model user preferences for the cold users who only rated a few items, additional information is often adopted. For example, [19] take into consideration both the social annotation (tag) and friendships inherently established among users in a music track recommender system. Due to the ambiguity of friendship, friends may have different preferences in items. In contrast, users joining the same online community are more likely to have similar preferences [38]. Hence by leveraging data from multiple channels including memberships in a project wiki [5,12] build a system named SONAR for people recommendation. Comparing with friendship and membership, trust information is of less ambiguity and more relevant to similarity [2,33,41]. Till now many trust-based approaches have been proposed [13,15,16,24], and trust has been demonstrated to be able to decrease recommendation errors and also increase recommendable items.

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 (trust.mindswap.org/FilmTrust/) 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. For example, [26] define the *profile-level* and *item-level* trust as the percentage of correct predictions from the views of general profile and specific items, respectively. [14,22] compute the implicit trust by the aggregation of value of a user's rating which is defined as the differences between the predicted ratings (based on only one user) and the ground truth. The intuition is that the closer two ratings are, the more value of that rating, and the more valuable ratings a user has, the more trustworthy the user will be. Since explicit trust is directly specified by users themselves, it is believed more accurate and reliable than implicit trust in determining the social relationships among users. In addition, [18] also show that letting users specify the explicit level of their trust (i.e., explicit trust) on the friends can improve the quality of recommendations. Hence, the present work focuses on the explicit trust.

Many approaches have been proposed to make use of the explicit trust. For example, [15] design the *TrustWalker* approach to randomly select trusted neighbors in the trust networks, where users are represented as nodes and trusted neighbors are connected with each other by trust links (i.e., edges) the strength of which indicates the trustworthiness between two users. Trust information of the selected neighbors is combined with an item-based technique to predict item ratings. In contrast, our work focuses on generating predictions by combining trust information with a user-based technique. [23] report that more accurate prediction algorithms are possible by incorporating trust information into traditional collaborative filtering. They do not directly use trust to substitute similarity but rather amplify similarity measurement by taking into account the number of messages exchanged among users. Hence, this approach is message specific. Further, a number of hybrid approaches incorporating trust are also proposed, such as [29,30]. Good performance can be achieved by combining both user- and item-based CF approaches. However, in this paper we focus on how to further improve the user-based CF using explicit trust.

The closest approaches to ours are as follows. [25] 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 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, [8] propose 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 [34]. Hence, we will only compare our method with one of them, namely *MoleTrust* in this paper. In addition, [4] propose 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 incorporate more similar users for recommendation. However, it performs badly in cold 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 [27]. 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 is achieved at the cost of poor coverage, and it fails to function in cold conditions where user similarity may not be computable.

In addition, most previous works are only evaluated on a single data set [4,8,25,27]. Besides, the reported results often show that they are able to achieve improvements in either accuracy or coverage, but not in both. Further, the cold start problem has not been well addressed yet, and proposing better trust-aware recommender systems remains a big challenge [32]. The purpose of our work is to take a step further in addressing this challenge by proposing a novel approach to incorporate trusted neighbors in CF.

3. The merge method

In this section, we will present the proposed Merge method the basic principle of which is to incorporate the ratings of trusted neighbors to complement and represent the preferences of active users. Three steps are taken to make recommendations. First, the trusted neighbors of the active user are identified and aggregated. Trust propagation may be needed to incorporate more trusted neighbors, especially useful for the cold users. Second, the ratings of trusted neighbors are merged 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. Third, similar users will be probed based on the merged rating profile, and recommendations are then generated in the same way as the conventional CF. Detailed descriptions as well as the insights of the Merge method are given in the subsequent sections.

3.1. Merging 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 , I 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 $v \in TN_u$, user u also specifies a trust value $t_{u,v} \in [0, 1]$ indicating the extent to which user u believes in user v 's ability in giving accurate ratings. We presume that user u will always trust herself in giving accurate ratings. Thus, user u herself is also included in the trust neighborhood TN_u , i.e. $u \in TN_u$ and $t_{u,u} = 1$. Besides, for simplicity, the set of items rated by user u is 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 $(u, j, \hat{r}_{u,j})$ 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. Aggregating trusted neighbors

The cold users are generally defined as the users who have rated less than five items [25]. Hence, to better model user preference, additional information is adopted. In our case, user's social trust information is utilized since users in the systems can specify other users as trusted neighbors. Since cold 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 cold users in real-world data sets, the specifications of which will be presented in Section 4.1. Fig. 1 shows the distribution of trusted neighbors for the cold users in three different real-world data sets.

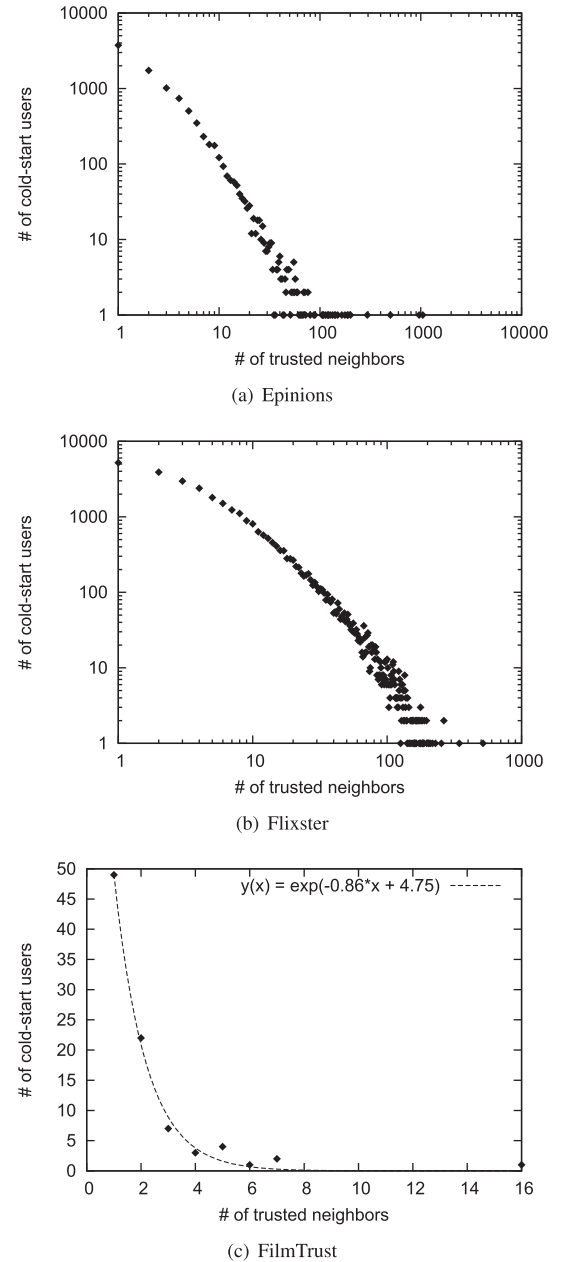


Fig. 1. The distributions of trusted neighbors for the cold users.

Both (a) and (b) in Fig. 1 show the well-known *power law* property of social network. Specifically, most cold users have only few trusted neighbors and only few cold users have identified many trusted neighbors. From Fig. 1(c), it is observed that FilmTrust has much fewer amount of trusted neighbors than Epinions and Flixster. Besides, the described trending line of data points follows an exponential function. Nevertheless, the distribution of trusted neighbors is quite close: only few cold users have many trusted neighbors whereas most cold users have only few ones. Therefore, although social trust can be regarded as a (strongly and positively) additional information source to model user preference, the availability of trust information for cold-users is relatively limited.

Fortunately, trust can be propagated along with the web-of-trust. That is, if users A trusts B and B trusts C , it can be inferred that users A trusts C to some extent. MoleTrust [25] and TidalTrust [8] 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 sets 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, we adopt a weighting factor to devalue the inferred trust in a long distance:

$$t_{u,v} = \frac{1}{d} * t'_{u,v}, \quad (1)$$

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 towards 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 incorporated. According to the theory of six-degree separation [36], that is, any two users in the social network can be connected (if possible) within small (less than six) steps. In this work, we restrict $d \leq 3$ ¹ to prevent meaningless searching and save computational cost for large-scale data sets. In fact, as we will show later, the Merge 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:

$$TN_u = \{v | t_{u,v} > \theta_t, v \in U\}, \quad (2)$$

where θ_t is 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. Although it is flexible to tune the trust threshold θ_t , it is not necessary to do so in practice. We defer the explanation till Section 3.1.3. In addition, the active user u herself is also regarded as a trusted neighbor in her trust neighborhood, i.e., $u \in TN_u$ and $t_{u,u} = 1$. In other words, we presume that user u will always believe in her own ratings as they are accurately reflecting her real preferences.

3.1.2. Merging the ratings of trusted neighbors

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

$$\tilde{I}_u = \{i | r_{v,i} \in R, \exists v \in TN_u, i \in I\}. \quad (3)$$

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 merged into a single rating based on the weights of trusted neighbors:

$$\tilde{r}_{u,j} = \frac{\sum_{v \in TN_u} w_{u,v} \cdot r_{v,j}}{\sum_{v \in TN_u} |w_{u,v}|}, \quad (4)$$

where $\tilde{r}_{u,j}$ is the merged 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 v 's ratings relative to the active user u . We claim that the importance weight $w_{u,v}$ is composed of three parts: trust value $t_{u,v}$, rating similarity $s_{u,v}$ and social similarity $j_{u,v}$. Hence, $w_{u,v}$ is computed as a linear combination of the three parts:

$$w_{u,v} = \alpha \cdot s_{u,v} + \beta \cdot t_{u,v} + (1 - \alpha - \beta) \cdot j_{u,v}, \quad (5)$$

where parameters α and β indicate the extent to which the combination relies on rating similarity and trust value, respectively. The rationale behind this computation, i.e., incorporating three parts rather than trust value only, is that people trusting each other may not share similar preferences [33]. Specifically, it is possible that trusted neighbors have low similarity. According to the work conducted by [27], it is noted that trusted neighbors with high similarity have a positive influence on the predictive accuracy after eliminating those with low similarity. Therefore, it is necessary to consider both rating similarity and trust value.

Pearson correlation coefficient [1] is often used to compute user similarity based on ratings:

$$s_{u,v} = \frac{\sum_{i \in I_{u,v}} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{u,v}} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{u,v}} (r_{v,i} - \bar{r}_v)^2}}, \quad (6)$$

where $s_{u,v} \in [-1, 1]$ 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 . Since the active user $u \in TN_u$, we denote $s_{u,u} = 1$ for the purpose of consistency. In particular, $s_{u,v} > 0$ means positive correlation between users u and v , $s_{u,v} < 0$ indicates opposite correlation and $s_{u,v} = 0$ implies no correlation. Alternative similarity measures could be cosine similarity [1], Bayesian similarity [11], etc.

In addition, as indicated by [27] 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 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., $s_{u,v} > 0$. Another reason is to be consistent with the value range of trust and social similarity in Eqs. (1) and (7).

The third 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:

$$j_{u,v} = \frac{|TN_u \cap TN_v|}{|TN_u \cup TN_v|}, \quad (7)$$

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 Eq. (5) since the three components are derived by Eqs. (1), (6) and (7), respectively. In this way, all the ratings of trusted neighbors on a certain item can be merged into a single value by Eq. (4).

Furthermore, since user u always gives accurate ratings from her own viewpoint, all her ratings will be retained and kept unchanged during the merging 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 merged. 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.

3.1.3. Determining the confidence of merged ratings

A merged rating for an active user on a certain item can be computed using Eq. (4) based on the ratings of trusted neighbors. However, the quality or usefulness of the merged ratings is unknown. We term it as the *confident* of the merged ratings, or

¹ The same setting is used in [25,4]. Better performance may be achieved by setting $d \leq 6$ and searching in a longer distance in the trust networks.

rating confidence for short, which reflect the usefulness of the merged ratings and to what extent the merged ratings are reliable. Intuitively, two factors may have important influence: the number of ratings involved and the conflicts between positive and negative opinions among all these ratings.

More specifically, if an item receives many ratings from the trusted neighbors, the merged value is likely to be correct and reliable. In contrast, if an item only receives few ratings, the merged value tends to be noisy and unreliable. In fact, as shown in Fig. 1, most cold users do not specify many other users as trusted neighbors, and by definition cold users rate only small number of items. In this paper, we regard the rating whose value is greater than the median rating scale as a positive opinion and otherwise as a negative opinion:

$$\begin{cases} r_{v,i} \text{ is positive :} & \text{if } r_{v,i} > r_{med}; \\ r_{v,i} \text{ is negative :} & \text{otherwise;} \end{cases} \quad (8)$$

where r_{med} is the median rating scale in the range from the minimum rating scale r_{min} to the maximum rating scale r_{max} predefined by a recommender system. The more consistent (i.e., less conflicts) between positive and negative opinions, the more reliable the merged rating will be. Therefore, only adopting the merged ratings may ignore the significant differences among different items and raise much noise in the merged rating profile, especially for those who have already rated many items, i.e., the *heavy users*. It is necessary to take into account the rating confidence for later rating predictions.

In conclusion, the measure of rating confidence should manage to reflect the differences in the number of ratings of trusted neighbors, and the differences in the conflicts between positive and negative opinions. Formally, the confidence $c_{u,j}$ of a merged rating $\tilde{r}_{u,j}$ is defined in the evidence space $\langle p_{u,j}, n_{u,j} \rangle$ (refers to [35]):

$$c_{u,j} = c(p_{u,j}, n_{u,j}) = \frac{1}{2} \int_0^1 \left| \frac{x^{p_{u,j}}(1-x)^{n_{u,j}}}{\int_0^1 x^{p_{u,j}}(1-x)^{n_{u,j}} dx} - 1 \right| dx, \quad (9)$$

where $c_{u,j} \in (0, 1]$ is the rating confidence of merged rating $\tilde{r}_{u,j}$ as a function of $p_{u,j}$ and $n_{u,j}$, referring to the number of positive, negative opinions (ratings) provided by the trusted neighbors on item $j \in \tilde{I}_u$, respectively. Hence, the greater the amount $p_{u,j} + n_{u,j}$ of ratings is, and the less conflicts between r and s will lead to greater confidence $c_{u,j}$. For consistency, the rating confidence of the ratings rated by the active users is always believed to be the highest, i.e., $c_{u,i} = 1$, for any item $i \in I_u$.

With the concept of rating confidence, we can now explain why it is not necessary to set or tune a proper trust threshold θ_t during the formation of trust neighborhood in Eq. (2). The reason is straightforward. Although less trustworthy users may be involved in the merging process, their influence to the merged rating is less than those with greater trust values, and the confidence measure can also mitigate their influence.

In summary, the merging process for each item $j \in \tilde{I}_u$ will produce two outputs: the merged rating $\tilde{r}_{u,j}$ and the corresponding rating confidence $c_{u,j}$. All the pairs of $(\tilde{r}_{u,j}, c_{u,j})$ will form a new rating profile to represent the preferences of the active users, based on which item predictions can be generated.

3.2. Incorporating with collaborative filtering

Given the new rating profile on the item set \tilde{I}_u after the merging process in Section 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 have rated 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 according to their ratings on the items that they commonly rated (see Eq. (6)). In our case, other than the merged ratings, the confidence is also important to indicate the quality of the merged ratings. Since Eq. (6) does not consider the rating confidence, we introduce a confidence-aware PCC to compute user similarity, denoted by CPCC:

$$s'_{u,v} = \frac{\sum_{i \in I_{u,v}} c_{u,i} (\tilde{r}_{u,i} - \bar{r}_u) (r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{u,v}} c_{u,i}^2 (\tilde{r}_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{u,v}} (r_{v,i} - \bar{r}_v)^2}}, \quad (10)$$

where $I_{u,v} = \tilde{I}_u \cap I_v$ is the set of items rated by both users u and v after the merging process, \bar{r}_u and \bar{r}_v are the average ratings for users u and v respectively, and $c_{u,i}$ is the confidence measurement regarding the merged rating $\tilde{r}_{u,i}$. The CPCC measure is inspired by the work of [37] in which the confidential weight of an item rated by the active user in their Eq. (7) plays the same role as the rating confidence in our work, i.e., to discount the values of their ratings. For a real rating $r_{v,j}$ provided by a similar user v , we consider that its rating confidence is $c_{v,j} = 1$ and hence omitted in Eq. (10).

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:

$$NN_u = \{v \mid s'_{u,v} > \theta_s, v \in U\}, \quad (11)$$

where θ_s is 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 cold 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. We will investigate the effect of similarity threshold for our method in the experiments (see Section 4.4.1).

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. We use the simple weighted average method, i.e., to compute the average value of all ratings provided by the nearest neighbors v weighted by their similarity $s_{u,v}$ with the active user u . Formally, the prediction is computed by:

$$\hat{r}_{u,j} = \frac{\sum_{v \in NN_u} s'_{u,v} \cdot r_{v,j}}{\sum_{v \in NN_u} |s'_{u,v}|}, \quad (12)$$

where $\hat{r}_{u,j}$ represents the predicted value on item j . Hence it ensures that the users with greater similarity will have more influence on the predictions. An alternative prediction method is Resnick's formula [1] which in addition considers user bias in giving ratings. Nevertheless, we adopt the weighted average because the two most related works [25,27] also take the same equation.

3.3. An example

In this section, we intend to exemplify step by step the use of the Merge 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 shown in Table 1(a). In addition, users may specify other users as trusted neighbors as shown in Table 1(b), 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. 2. 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 .

The first step of the Merge method is to identify the trusted neighbors of the active user by allowing trust propagation in the trust network. According to Fig. 2, trust values between the active user u_1 and other users can be inferred by Eq. (1) and the results are presented in Table 2. In particular, as an active user, u_1 always trusts herself in giving accurate ratings and hence $t_{u_1, u_1} = 1.0$. 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 although the trust value of user u_6 is computable, this user will not be regarded as an inferred trusted neighbors due to the constraint $d \leq 3$. Hence, a set of users $TN_{u_1} = \{u_1, u_2, u_3, u_4, u_5\}$ are identified as trusted neighbors for active user u_1 .

Second, the ratings of trusted neighbors will be merged using Eqs. (4), (5) and (9). For simplicity, in this example we set $\alpha = 0, \beta = 1$ for Eq. (5), i.e., trust values are used as user weights. The resultant merged ratings and confidences are presented in Table 3. In particular, since user u_1 has rated item i_3 , and we presume the active user will always believe in her own ratings, hence there is no need to consider the ratings of trusted neighbors. Therefore, the merged rating on item i_3 is equal to r_{u_1, i_3} (i.e., 5), and the confidence is the highest (i.e., 1.0). For other items that user u_1 has not rated, the ratings of trusted neighbors will be merged by Eq. (4) as well as the rating confidence by Eq. (9). Take item i_1 as

Table 1
The synthetic data set consisting of both (a) rating and (b) trust information.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9
(a) User-item rating matrix									
u_1			5		?				
u_2	5		4		3			2	
u_3		4		3				1	
u_4	3		5		2				
u_5		4	4		3			3	
u_6		3	3	5	5				
u_7							5		4
u_8			4		2			1	
u_9			4		5			5	
	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9
(b) User-user trust matrix									
u_1	1		1						
u_2		1		1					
u_3		1		1					
u_4				1					
u_5					1				
u_6			1	1					
u_7							1		
u_8								1	
u_9									1

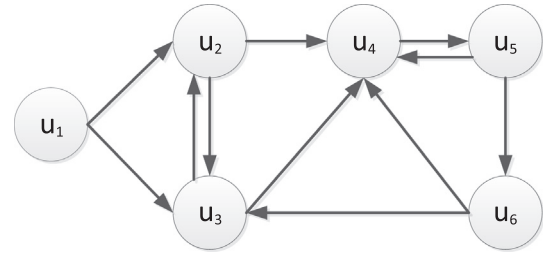


Fig. 2. The trust network for a cold user u_1 .

Table 2

The 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	0	1	1	2	3	4			
t_{u_1, u_k}	1.0	1.0	1.0	0.5	0.33	0.25			

Table 3

The merged rating profile for user u_1 .

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9
\tilde{r}_{u_1, i_j}	4.33	4	5	3	2.73			1.72	
c_{u_1, i_j}	0.19	0.38	1.0	0.25	0.47			0.47	

an instance. The ratings of users u_2 and u_4 will be averaged and weighted by their trust values, i.e.,

$$\tilde{r}_{u_1, i_1} = \frac{5 \times 1.0 + 3 \times 0.5}{1.0 + 0.5} = 4.33$$

For rating scales from 1 to 5, the median rating is 3. According to Eq. (8), user u_2 's rating 5 is regraded as positive, while user u_4 's rating 3 is negative. The confidence is derived by:

$$c_{u_1, i_1} = c(1, 1) = 0.19$$

This procedure continues until all the items rated by at least a trusted neighbor has been covered. A new rating profile is formed and shown in Table 3. Since there are only a few trusted neighbors, the computed confidence is relatively small. The merged rating profile is much more complete than the original.

Third, user similarity is computed by Eq. (10) based on the formed rating profile (see Table 3), taking into account the rating confidence. The results are shown in Table 4. For consistency, the similarity between user u_1 and herself is 1.0. For comparison purpose, we also show the similarity values computed by conventional PCC (see Eq. (6)). It is noted that PCC values are less distinguishable than CPCC values, and the differences between CPCC and PCC values could be large. In other words, the confidence plays an important role in our similarity computation. A set of users $NN_{u_1} = \{u_2, u_4, u_5, u_8\}$ are selected as nearest neighbors, whose similarity is greater than the threshold $\theta_s = 0$ and who have rated the target item i_5 (noted that user u_3 did not rate item i_5).

Finally, a prediction for item i_5 is generated by Eq. (4):

$$\hat{r}_{u_1, i_5} = \frac{3 \times 0.66 + 2 \times 0.995 + 3 \times 0.84 + 2 \times 0.99}{0.66 + 0.995 + 0.84 + 0.99} = 2.43$$

Table 4

The 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
CPCC	1.0	0.66	0.995	0.98	0.84	-0.78		0.99	-0.98
PCC	1.0	0.87	0.992	0.91	0.91	-0.91		1.0	-0.95

and the rating confidence is:

$$c_{u_1, i_5} = c(0, 4) = 0.53$$

Compared with the values (2.73, 0.47) shown in Table 3, the final prediction is different from the merged rating which is only based on trusted neighbors, and the final rating confidence is higher than the merged one since more ratings of similar users are used. In other words, generating a prediction only based on trusted neighbors may not be reliable, and the resultant rating confidence could be low if only few trusted neighbors can be identified. This is the situation for the cold users. In contrast, by merging the ratings of trusted neighbors, the ratings of similar users can be adopted to smooth the predictions and enhance the confidence. Furthermore, the only item that user u_1 has rated is i_3 which receives many ratings from system users. That is, item i_3 is a popular item. The conventional CF will treat all the users as similar users and hence the extreme ratings given by users u_6 and u_9 will bias the final prediction for item i_5 . By forming a more complete rating profile for the active user u_1 , the Merge method is able to identify that users u_6 and u_9 in fact have different preferences and hence they will be excluded to generate the final prediction. As a result, the prediction generated by our method is likely to be more accurate and reliable.

3.4. The strength of the merge method

Principally, the Merge method has two distinct advantages relative to other methods. First, it can effectively ameliorate the data sparsity and cold start problems. The essential challenge of the two issues is that the small amount of items commonly rated by two users makes it difficult to accurately compute user similarity, and hence to find reliable similar users. Even worse, two users may not have any co-rated items in common, resulting in non-computable user similarity. The Merge method cope with the cold conditions by merging the ratings of the trusted neighbors to form a new rating profile which is used to represent the preferences of the active user. Specifically, the relation $I_u \in \tilde{I}_u$ can be inferred from Eq. (3) because of $u \in TN_u$, that is, the newly formed rating profile covers more items than the original rating profile. The previous example also confirmed this point. Thus, more similar users can be identified in terms of user similarity, especially useful for cold users with only a few or none ratings. The example in Section 3.3 also showed that the computed similarity tended to be more reliable and distinguishable by considering confidence. As a consequence, our method can alleviate the data sparsity and cold start problems.

Second, the Merge method can function well in the case of either sparse rating or sparse trust information. Previously, many trust-based approaches such as MoleTrust [25] and TidalTrust [34] predict item ratings only based on the ratings provided by the trusted neighbors. Hence these approaches may also suffer from the similar cold-start problem where some users may only specify a small number of other users as their trusted neighbors, which has been demonstrated in three real-world data sets and shown in Fig. 1. Hence this issue could be a common case for many online systems, especially when users are lack of incentives to proactively connect with each other. In this case, the performance will be limited since only a few neighbors can be incorporated for recommendation. In contrast, the Merge method addresses this problem by also making use of the ratings of the active users if any. In particular, when the active user has not specified any trusted neighbors but rated a certain number of items, the merged rating profile will then be exactly the same as her own and real rating profile because the only trusted neighbor is herself. The Merge method will have no differences with the conventional CF method. On the other hand, when the cold user has not rated any items but specified some trusted neighbors, then the ratings of these trusted neighbors can be merged as we described. In either case, our method

Table 5

The specifications of three data sets.

Data set	# Users	# Items	# Ratings	# Trust	Sparsity (%)
FilmTrust	1986	2071	35,497	1853	98.86
Flixster	53 K	18 K	410 K	650 K	99.96
Epinions	49 K	139 K	664 K	478 K	99.95

is competent to form a new rating profile and hence mitigate the cold start problem. Although our method will fail to work when there is neither rating nor trust information of the active users, other kind of information may be needed to help model user preference which is beyond the scope of this paper.

In this regard, our method possess some advantages of hybrid approaches. By merging user- and item-based approaches together, hybrid methods can also alleviate the concern issues, such as [21,30]. However, our method differs from those methods in threefold. Firstly, our work is only based on rating information, namely item ratings and trust ratings. Hybrid methods usually depend on more heterogeneous information, such as music genre in [21] or item taxonomy in [30]. Hence, our method is more generic than the hybrid ones. Secondly, more complex information needs more computational steps to deal with, hence the hybrid methods are usually more complex and hard to be implemented than single approaches. Lastly, as a user-based approach, our method holds the potential to be incorporated with other item-based approaches to form more powerful hybrid methods in the future.

4. Evaluation

In order to verify the effectiveness of the Merge method, we conduct experiments on three real-world data sets. Specifically, we aim to find out: (1) how the performance of our method in comparison with other counterparts; and (2) what is the effect of trust propagation to our method and the others.

4.1. Data acquisition

Three real-world data sets are used in our experiments, namely FilmTrust,² Flixster³ and Epinions⁴ as they are possibly the only available data sets that contain both the data of explicit trust statements and user-item ratings. The specifications of the three data sets are summarized in Table 5.

FilmTrust is a trust-based social site in which users can rate and review movies. Since there is no publicly available data sets due to the preservation of user privacy, we crawled the whole site in June 2011, collecting 1986 users, 2071 movies and 35,497 ratings. The ratings take values from 0.5 to 4.0 with step 0.5. In addition, we also gathered 1853 trust ratings that are issued by 609 users. The average number of trusted neighbors per user is less than 1. Originally, users can specify other users as trusted neighbors with a certain level of trust from 1 to 10. However, these trust values are not available due to the sharing policy. We can only get the link information among users and hence the trust value is 1 if a link exists between two users otherwise the value is 0.

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 set⁵ collected by [16] which includes a large amount of data. The ratings are real values ranged from 0.5 to 4.0 with an interval 0.5, and the trust statements

² <http://trust.mindswap.org/FilmTrust/>.

³ <http://www.flixster.com/>.

⁴ <http://www.epinions.com/>.

⁵ <http://www.cs.sfu.ca/sja25/personal/datasets/>.

are scaled from 1 to 10 but not available. Hence, they are converted into binary values the same as FilmTrust, 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 is symmetric. We sample a subset by randomly choosing 53 K users who issued 410 K item ratings and 655 K trust ratings.

Epinions is a website in which users can express their opinions about items (such as movies, books, and software) by assigning numerical ratings and writing text reviews. Users can specify other users as trustworthy (to the trust list) or untrustworthy (to the distrust list) according to whether the text reviews and comments of other users are consistently valuable to them or not. The data set⁶ is generated by [25], consisting of 49 K users who issued 664 K ratings over 139 K different items and 478 K trust statements. The ratings are integers ranged from 1 to 5, and the trust values are also binary (either 1 or 0). The rating sparsity is computed by:

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

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

4.2. Experimental settings

In the experiments, we compare the performance of our method Merge 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, selects the users whose similarity is above the predefined similarity threshold θ_s for Eq. (11), and uses their ratings to generate item predictions by Eq. (12). In this work, the threshold θ_s is set 0 for all methods.
- **MT** $x(x = 1, 2, 3)$ is the implementation of the MoleTrust algorithm [25] 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 [27] 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.
- **TCF** $x(x = 1, 2)$ denotes the approach proposed by [4] 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 incorporate 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.
- **Mergex** $(x = 1, 2, 3)$ is our method with the trust propagation length x , aiming to investigate the impact of trust propagation on the Merge method. Besides, we denote **Merge- α** as a variant where parameter α in Eq. (5) is set 1, meaning the importance weight is completely determined by user similarity. Further, we also denote **Merge- β** as a variant with the best performance when parameter β in Eq. (5) is set 0, meaning explicitly specified or inferred trust value is not used.

In addition, we split each data set into two different views as defined in [25]: the view of **All Users** represents that all users and their ratings will be tested whereas the view of **Cold Users** denotes that only the cold 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 *Cold 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:

$$MAE = \frac{\sum_u \sum_i |\hat{r}_{u,i} - r_{u,i}|}{N}, \quad (13)$$

where N is the number of testing ratings. Hence, the smaller the MAE value is, the closer a prediction is to the ground truth. Inspired by [15] 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:

$$iMAE = 1 - \frac{MAE}{r_{\max} - r_{\min}}, \quad (14)$$

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:

$$RC = \frac{M}{N}, \quad (15)$$

where M and N are 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 [15], the F-measure is computed by:

$$F1 = \frac{2 \cdot iMAE \cdot RC}{iMAE + RC}. \quad (16)$$

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

4.4. Results and analysis

In this section, we conduct a series of experiments on three real-world data sets to demonstrate the effectiveness of our approach relative to others, and thus to answer the research questions proposed in Section 4. Both data set views, namely *All Users* and *Cold Users* are tested. The results are presented in Tables 6–8 corresponding to the predictive performance on the FilmTrust, Flixster, and Epinions data sets, respectively.

4.4.1. Effect of similarity threshold θ_s

The Merge method requires to select a set of nearest neighbors to make a prediction, referring to Eq. (11) where a similarity threshold θ_s is used. In this experiment, we intend to determine the best similarity threshold for cold users. For simplicity, we set $\alpha = 0.5, \beta = 0.3$ (see Eq. (5), explained in the next subsection), and vary the threshold θ from 0.0 to 0.9 with step 0.1. The

⁶ http://www.trustlet.org/datasets/downloaded_epinions.

Table 6

The predictive performance on the FilmTrust data set.

Views	Approaches measured by MAE, RC and F1											
	CF	MT1	MT2	MT3	RN	TCF1	TCF2	Merge- α	Merge- β	Merge1	Merge2	Merge3
All	0.703	0.852	0.795	0.771	0.571	0.714	0.719	0.703	0.704	0.705	0.707	0.708
Users	93.84%	21.20%	27.96%	30.38%	0.74%	94.92%	95.19%	94.06%	94.21%	94.77%	94.94%	95.06%
	0.8631	0.3312	0.4106	0.4373	0.0147	0.8658	0.8661	0.8640	0.8647	0.8667	0.8672	0.8674
Cold	0.744	0.853	0.880	0.819	NaN	0.751	0.751	0.737	0.764	0.768	0.772	0.768
Users	39.64%	17.11%	23.19%	23.85%	0.00%	39.97%	40.79%	39.80%	43.26%	53.45%	54.11%	54.28%
	0.5273	0.2791	0.3541	0.3637	NaN	0.5298	0.5369	0.5292	0.5569	0.6345	0.6387	0.6404

Table 7

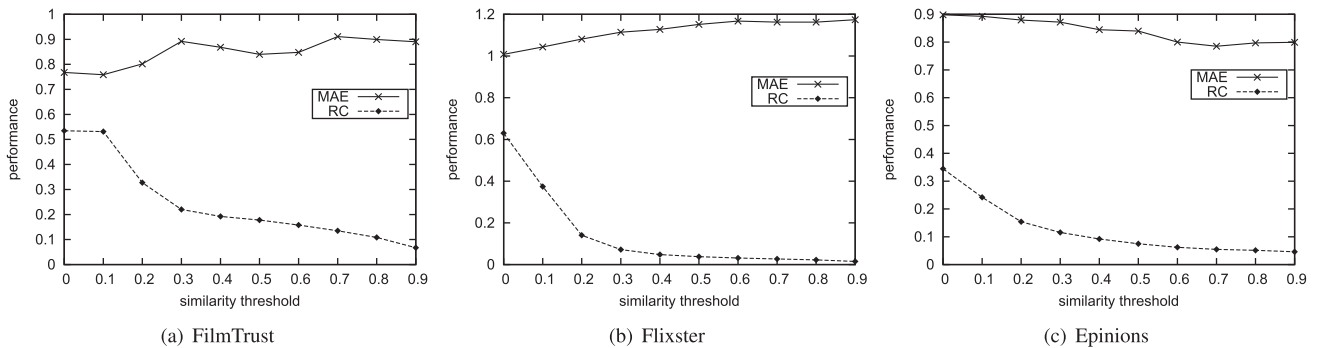
The predictive performance on the Flixster data set.

Views	Approaches measured by MAE, RC and F1											
	CF	MT1	MT2	MT3	RN	TCF1	TCF2	Merge- α	Merge- β	Merge1	Merge2	Merge3
All	0.928	1.060	0.932	0.862	0.858	0.870	0.850	0.917	0.903	0.890	0.877	0.875
Users	68.56%	12.36%	71.37%	90.71%	0.38%	80.92%	85.23%	69.63%	82.93%	89.64%	94.39%	95.03%
	0.7357	0.2128	0.7512	0.8549	0.0076	0.8079	0.8312	0.7429	0.8141	0.8467	0.8690	0.8720
Cold	1.153	1.127	1.005	0.934	NaN	1.047	0.923	1.147	1.018	1.008	0.960	0.949
Users	3.27%	8.11%	52.69%	79.55%	0.00%	12.97%	21.41%	3.30%	41.57%	63.08%	83.13%	85.15%
	0.0626	0.1464	0.6279	0.7939	NaN	0.2219	0.3373	0.0632	0.5409	0.6959	0.8083	0.8190

Table 8

The predictive performance on the Epinions data set.

Views	Approaches measured by MAE, RC and F1											
	CF	MT1	MT2	MT3	RN	TCF1	TCF2	Merge- α	Merge- β	Merge1	Merge2	Merge3
All	0.876	0.845	0.852	0.832	0.673	0.867	0.864	0.851	0.841	0.839	0.824	0.820
Users	51.24%	26.34%	57.64%	71.68%	9.87%	70.28%	77.48%	59.72%	68.61%	73.35%	78.50%	80.02%
	0.6188	0.3949	0.6654	0.7525	0.1765	0.7409	0.7794	0.6792	0.7343	0.7608	0.7895	0.7976
Cold	1.033	0.756	0.916	0.890	NaN	0.982	0.941	1.038	0.913	0.898	0.876	0.867
Users	3.22%	6.57%	22.06%	41.73%	0.00%	7.16%	10.45%	3.37%	18.62%	34.49%	49.59%	52.66%
	0.0617	0.1216	0.3431	0.5431	NaN	0.1308	0.1839	0.0644	0.3000	0.4774	0.6066	0.6298

**Fig. 3.** The performance in the view of Cold Users on three data sets.

performance in the view of Cold Users on three data sets is illustrated in Fig. 3.

The results show that as similarity threshold increases, the rating coverage (RC) decreases dramatically. It is because less nearest neighbors are used to make predictions. Although users with greater similarities are adopted, it does not mean that the predictions generated by few highly similar users will be more reliable than those generated by many users with smaller similarities. Hence, the trends of accuracy (MAE) may vary in different data sets. We select the similarity threshold such that both the accuracy and rating coverage are high. Therefore, in the following experiments the similarity threshold is set $\theta_s = 0$.

4.4.2. Importance weights with parameters α and β

An important step for the Merge method is to compute the importance weights of trusted neighbors which is a linear combination of rating similarity, trust value and social similarity with parameters α and β (see Eq. (5)). When $\alpha = 1$, the weights of trusted neighbors depend completely on the rating similarity, and the performance of this variant is denoted by Merge- α . When $\beta = 0$, the trust values are not considered and the importance weights are totally dependent on rating and social similarities. The best performance in this case is obtained when $\alpha = 0.5$ on FilmTrust and Flixster, and $\alpha = 0.7$ on Epinions. We denote this specific variant as Merge- β . When $\alpha \in (0, 1)$, both rating-based

value and trust-based value will be used. In fact, the experiments show that the settings of (α, β) are (0.5, 0.3) on FilmTrust, (0.5, 0.4) on Flixster and (0.2, 0.4) on Epinions achieve the best performance. It is denoted as **Merge1** when trust propagation is not used. Merge2 and Merge3 adopt the same settings except the length of trust propagation.

From Tables 6–8, it is observed that Merge- α achieves similar performance with the baseline CF method in both views of *All Users* and *Cold Users*. More specifically, both accuracy (in terms of MAE) and coverage (in RC) are slightly improved as well as the overall performance (in F1). This is because for cold users (view *Cold Users*), although a number of trusted neighbors may be identified, the number of trusted neighbors with computable and positive similarity could be small due to only few ratings available. For other type of users (view *All Users*), many similar users can be identified which hence also include the trusted neighbors with computable similarity. In other words, these similar users may already cover a high ratio of items, resulting in smaller effect of the Merging process.

For the Merge- β where social similarity is considered in addition to rating similarity, the performance is significantly improved relative to Merge- α , especially in Flixster and Epinions. However in FilmTrust, the differences between Merge- α and Merge- β are marginal. This may be explained by the fact that the CF method already achieves sufficiently good performance. When social information is added, the quality of rating similarity may be decreased due to possible noise embedded, and hence the accuracy may be slightly decreased whereas the coverage is increased to a limited extent. Nevertheless, the overall performance (in F1) is improved. In contrast, the rating information is less useful in Flixster and Epinions, and using social similarity can effectively increase both the accuracy (up to 12% increments) and coverage (up to 38% increments) as well as the overall performance, especially in the view of *Cold Users* which is the main concern of this work.

Furthermore, by taking into account both rating and trust information, Merge1 achieves even better performance than Merge- β method in terms of accuracy and coverage. Again, the greatest improvement is observed in the view of *Cold Users*, especially in coverage (up to 20% increments). As the best parameters are set by different value combinations across three data sets, we may conclude that similarity (0.5) is more important than trust value (0.3 or 0.4) which is superior to social similarity in determining user preferences. Furthermore, it shows that both rating and trust information are useful and should be integrated to improve the recommendation performance.

4.4.3. 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 Merge method. Compared with Merge1, Merge2 and Merge3 have a better accuracy and coverage. This may be explained by that the merged ratings will be more accurate and the rating confidence will be greater due to more evidences (i.e., ratings) available.

Note that the differences between Merge2 and Merge3 are less than the differences between Merge1 and Merge2. 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 merged 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 merged ratings (due

Table 9

The improvements of all methods comparing with CF in F1.

Dataset	View	MTx (%)	RN	TCFx (%)	Mergex (%)
FilmTrust	All	−49.33	−98.30%	0.35	0.50
	Cold	−31.03	NaN	1.82	21.45
Flixster	All	16.20	−98.97%	12.98	18.53
	Cold	1168.21	NaN	438.82	1208.31
Epinions	All	21.61	−71.48%	25.95	28.89
	Cold	780.23	NaN	198.06	920.75

to few ratings), and hence harm the predictive performance such as that in the FilmTrust data set.

4.4.4. Comparison with other methods

For other methods, we obtain close results on Epinions as shown in Table 8 relative to those reported in [4,25]. The similar trends of results are also obtained on the other two data sets, as shown in Tables 6 and 7. More specifically, CF cannot achieve large portion of predictable items, especially on the large-scale data sets (i.e., Flixster and Epinions) and the accuracy is usually bad. It confirms that CF suffers from cold start severely. The RN method accomplishes good accuracy but 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 achieve better performance for cold users in all the data sets except in the FilmTrust where only our Merge method outperforms it in both accuracy and coverage. When only direct trusted neighbors are used (MT1, Merge1), our method achieves better accuracy and coverage in FilmTrust and Flixster. In Epinions, MT1 works better than our method in accuracy but much worse in coverage. It shows that MT1 may have a good accuracy in some data sets, but not consistently in all the data sets. When trust is propagated in longer length, both accuracy and coverage are increased in Flixster and Epinions whereas only coverage increases in FilmTrust. Nevertheless, our method outperforms MTx in all the data sets. TCF methods generally obtain better coverage in the view of *All Users*. However, for cold 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 cold conditions. This fact leads to bad performance of TCF methods. In contrast, our method is not subject to the ratings of cold users themselves. Instead, trust information is merged to form a more concrete rating profile for the cold users based on which CF is applied to find similar users and hence generate recommendations. Consistently, we come to a conclusion that the Merge method outperforms the other approaches both in accuracy and coverage as well as a better balance between them.

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 by⁷:

$$\text{Improvement} = \frac{\text{Method.F1} - \text{CF.F1}}{\text{CF.F1}} \times 100\% \quad (17)$$

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 changes between Method and CF, the more improvements we obtain. The results are shown in Table 9, where *All* and *Cold* refer to the cases of *All Users* and *Cold Users* for simplicity, respec-

⁷ The formula can be referred to as the *relative change* defined in http://en.wikipedia.org/wiki/Relative_change_and_difference.

tively. To explain, we take two values in Table 9 as an example, namely 21.45% and 1208.31% for our method *Mergex*. In the *Cold* case of FilmTrust, the best Merge method shown in Table 6 is Merge3 with F1 value 0.6404, while the F1 of CF is 0.5273. Hence, the improvement is $(0.6404 - 0.5273) / 0.5273 * 100\% = 21.45\%$. Similarly, in the *Cold* view of Flixster, Merge3 achieves F1 value 0.8190 while CF has a poor performance with 0.0626 (see Table 7), leading to the improvement $(0.8190 - 0.0626) / 0.0626 * 100\% = 1208.31\%$. Other values can be explained and verified as well. Note that the value NaN indicates the improvement is not computable for the RN method in the view of *Cold*. This can be explained by the fact that RN cannot cope with cold users and predict item ratings (see Tables 6–8). 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.

5. Conclusion and future work

This paper proposed a novel method to incorporate 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 merged to complement and represent the preferences of the active users, based on which similar users can be identified and recommendations are generated. The quality of merged ratings was measured by the confidence considering the number of ratings involved and the conflicts between positive and negative opinions (i.e., ratings). The rating confidence was incorporated to compute user similarity, and hence a confidence-aware similarity measure was introduced. The prediction of a given item is generated by averaging the ratings of similar users weighted by their importance. Experiments on three real-world data sets were conducted and the results showed that significant improvements against other methods were obtained both in accuracy and 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 both trust and similarity to improve the performance of collaborative filtering.

The present work depends on the explicit trust during the merging 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 Merge method.

References

- [1] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, *IEEE Trans. Knowl. Data Eng.* 17 (2005) 734–749.
- [2] Jøsang Audun, D.K. Walter Quattrochicchi, Taste and trust, in: *Trust Management V*, 2011, pp. 312–322.
- [3] R.M. Bell, Y. Koren, Lessons from the netflix prize challenge, *ACM SIGKDD Explor. Newsl.* 9 (2007) 75–79.
- [4] M. Chowdhury, A. Thomo, B. Wadge, Trust-based infinitesimals for enhanced collaborative filtering, in: *Proceedings of the 15th International Conference on Management of Data*, 2009.
- [5] C. Danis, D. Singer, A wiki instance in the enterprise: opportunities, concerns and reality, in: *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work*, 2008, pp. 495–504.
- [6] C. De Rosa, J. Cantrell, A. Havens, J. Hawk, L. Jenkins, B. Gauder, R. Limes, D. Cellentani, OCLC, Sharing, Privacy and Trust in Our Networked World: A Report to the OCLC Membership, OCLC, 2007.
- [7] E. Diaz-Aviles, L. Drumond, L. Schmidt-Thieme, W. Nejdl, Real-time top-n recommendation in social streams, in: *Proceedings of the 6th ACM Conference on Recommender Systems*, 2012, pp. 59–66.
- [8] J. Golbeck, 2005. Computing and Applying Trust in Web-Based Social Networks. Ph.D. thesis.
- [9] M. Gueye, T. Abdessalem, H. Naacke, Dynamic Recommender System: Using Cluster-Based Biases to Improve the Accuracy of the Predictions, 2012, arXiv preprint arXiv:1212.0763.
- [10] G. Guo, J. Zhang, D. Thalmann, A simple but effective method to incorporate trusted neighbors in recommender systems, in: *Proceedings of the 20th International Conference on User Modeling, Adaptation and Personalization*, 2012, pp. 114–125.
- [11] G. Guo, J. Zhang, N. Yorke-Smith, A novel bayesian similarity measure for recommender systems, in: *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, 2013.
- [12] I. Guy, I. Ronen, E. Wilcox, Do you know?: recommending people to invite into your social network, in: *Proceedings of the 14th International Conference on Intelligent User Interfaces*, 2009, pp. 77–86.
- [13] C. Haydar, A. Boyer, A. Roussanaly, et al., Hybridising collaborative filtering and trust-aware recommender systems, in: *Proceedings of the 8th International Conference on Web Information Systems and Technologies*, 2012.
- [14] C.S. Hwang, Y.P. Chen, Using trust in collaborative filtering recommendation, in: *New Trends in Applied Artificial Intelligence*, 2007, pp. 1052–1060.
- [15] M. Jamali, M. Ester, Trustwalker: a random walk model for combining trust-based and item-based recommendation, in: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 397–406.
- [16] M. Jamali, M. Ester, A matrix factorization technique with trust propagation for recommendation in social networks, in: *Proceedings of the fourth ACM conference on Recommender systems*, ACM, 2010, pp. 135–142.
- [17] D. Jannach, M. Zanker, M. Ge, M. Gröning, Recommender systems in computer science and information systems—a landscape of research, *E-Commerce Web Technol.* (2012) 76–87.
- [18] B. Knijnenburg, J. O'Donovan, S. Bostandjiev, A. Kobsa, Inspectability and control in social recommenders, in: *Proceedings of the 6th ACM Conference on Recommender Systems*, 2012.
- [19] I. Konstas, V. Stathopoulos, J. Jose, On social networks and collaborative recommendation, in: *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2009, pp. 195–202.
- [20] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, *Computer* 42 (2009) 30–37.
- [21] A.S. Lampropoulos, P.S. Lampropoulou, G.A. Tsihrintzis, A cascade-hybrid music recommender system for mobile services based on musical genre classification and personality diagnosis, *Multimedia Tools Appl.* 59 (2012) 241–258.
- [22] N. Lathia, S. Hailes, L. Capra, Trust-based collaborative filtering (2008) 119–134.
- [23] F. Liu, H. Lee, Use of social network information to enhance collaborative filtering performance, *Expert Syst. Appl.* 37 (2010) 4772–4778.
- [24] H. Ma, I. King, M. Lyu, Learning to recommend with social trust ensemble, in: *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2009, pp. 203–210.
- [25] P. Massa, P. Avesani, Trust-aware recommender systems, in: *Proceedings of the 2007 ACM Conference on Recommender Systems*, 2007, pp. 17–24.
- [26] J. O'Donovan, B. Smyth, Trust in recommender systems, in: *Proceedings of the 10th International Conference on Intelligent User Interfaces*, 2005, pp. 167–174.
- [27] S. Ray, A. Mahanti, Improving prediction accuracy in trust-aware recommender systems, in: *Proceedings of the 43rd Hawaii International Conference on System Sciences*, 2010, pp. 1–9.
- [28] A. Seth, J. Zhang, R. Cohen, Bayesian credibility modeling for personalized recommendation in participatory media, in: *Proceedings of the International Conference on User Modeling, Adaptation, and Personalization*, 2010, pp. 279–290.
- [29] Q. Shambour, J. Lu, A hybrid trust-enhanced collaborative filtering recommendation approach for personalized government-to-business e-services, *Int. J. Intell. Syst.* 26 (2011) 814–843.
- [30] Q. Shambour, J. Lu, A trust-semantic fusion-based recommendation approach for e-business applications, *Decis. Support Syst.* 54 (2012) 768–780.
- [31] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, N. Oliver, Tfmap: Optimizing map for top-n context-aware recommendation, in: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2012, pp. 155–164.
- [32] Y. Shi, M. Larson, A. Hanjalic, How far are we in trust-aware recommendation?, in: *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, Springer, 2011, pp. 704–707.
- [33] P. Singla, M. Richardson, Yes, there is a correlation: from social networks to personal behavior on the web, in: *Proceedings of the 17th International Conference on World Wide Web*, 2008, pp. 655–664.
- [34] P. Victor, C. Cornelis, M. De Cock, A. Teredesai, Trust- and distrust-based recommendations for controversial reviews, in: *Proceedings of the 2009 International Conference on Web Science*, 2009, p. 5.
- [35] Y. Wang, M. Singh, Formal trust model for multiagent systems, in: *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 1551–1556.
- [36] D.J. Watts, *Six Degrees: The Science of A Connected Age*, WW Norton, 2004.
- [37] G. Xue, C. Lin, Q. Yang, W. Xi, H. Zeng, Y. Yu, Z. Chen, Scalable collaborative filtering using cluster-based smoothing, in: *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2005, pp. 114–121.
- [38] Q. Yuan, S. Zhao, L. Chen, Y. Liu, S. Ding, X. Zhang, W. Zheng, Augmenting collaborative recommender by fusing explicit social relationships, in:

- Proceedings of Workshop on Recommender Systems and the Social Web, 2009, pp. 49–56.
- [39] S. Zhang, W. Wang, J. Ford, F. Makedon, [Learning from Incomplete Ratings Using Non-Negative Matrix Factorization](#), SIAM, 2006.
- [40] X. Zhou, Y. Xu, Y. Li, A. Josang, C. Cox, [The state-of-the-art in personalized recommender systems for social networking](#), *Artif. Intell. Rev.* (2012) 1–14.
- [41] C. Ziegler, G. Lausen, [Analyzing correlation between trust and user similarity in online communities](#), in: *Trust Management*, 2004, pp. 251–265.