Send Orders for Reprints to reprints@benthamscience.ae

The Open Cybernetics & Systemics Journal, 2014, *8*, 843-852

Open Access

A Random Walk Method Using Trust Factor in Collaborative Filtering

Ming He* and Zheng Wei

College of Computer Science, Beijing University of Technology, Chaoyang District, Beijing, 100124, China

Abstract: Collaborative filtering is one of the most widely used techniques for recommendation system which has been successfully applied in many applications. However, it suffers from the cold start users who rate only a small fraction of the available items. In addition, these methods can not indicate confidence they are for recommendation. Trust-based recommendation methods assume the additional knowledge of a trust network among users and can alleviate the cold start users, since users only need to be simply connected to the trust network. On the other hand, the sparse user item ratings lead the trust-based method to consider ratings of indirect neighbors that are only weakly trusted, which may decrease its precision. In this paper, we improved the random walk model combining the trust factor-based and the collaborative filtering method for recommendation. The trust factor is considered as important a measure of guiding recommendations. The empirical analysis on the Epinions dataset demonstrates that our method outperform other trust-based and collaborative filtering methods.

Keywords: Collaborative filtering, random walk, recommdndation system, trust.

1. INTRODUCTION

With the information available on the Internet to us growing explosively far more rapidly than our ability to process it, technologies to help people select the relevant part from the huge amount of information efficiently is becoming necessary to overcome the resulted information overload problem. Recommender system is considered one such promising technology that aims to generate item recommendations from a huge collection of items based on users' preferences.

The collaborative filtering [1] is one of the most widely used approaches in recommendation system which uses the known preferences of users to make recommendations or predictions to a target user. With the advent of online social networks as important channels of online information, the trust-based method for recommendation has emerged. This method assumes that a trust network among users and makes recommendations based on the ratings of the users that are directly or indirectly trusted by users.

Collaborative filtering achieves effectively predicts user preferences when users have expressed enough ratings to have common ratings with other user, but it suffers from the well-known cold start problem – the prediction performance on new items and on new users. The problem is common to all kinds of recommender system, both contend-based and collaborative recommenders. Using similarity methods, it is unlikely to find similar users since the cold start users only have a few ratings. In fact, cold start can be considered as a sub problem of coverage because it measures the system coverage over a specific set of items and users. Trust-based recommender system allows us to base recommendations only on ratings given by users trusted directly by the current user or indirectly, for example trusted by another trusted user. The intuition that people tend to rely more on recommendations from people they trust than on online recommender systems which generate recommendations based on anonymous people similar to them. In this way it is possible to cut out malicious users who are trying to influence recommendation accuracy.

Using a trust network therefore improves the coverage of recommendations and alleviates the cold start problem in a trust network of a recommender system. However, the further away we go from source user u in the trust network, the weaker the trust will be between these users and the source user. Meanwhile user ratings will become noisy and unreliable. Therefore, we have to use the ratings expressed by users in the neighbourhood close to the user u. But, in this case the probability of finding a rating expressed on the item will be very low and we will not be able to compute a prediction.

The focus of this paper is on a trust-based recommender system that can better deal with cold start users and the sparsity of the user item ratings. On the other hand, the trust-based method is also much more robust to fraudulent attacks. The method described here combining trust-based and item-based recommendation. The proposed method considers not only ratings of the target item, but also those of similar items. The probability of using the rating of a similar item instead of a rating for the target item increases with increasing length of the walk. Specifically, our method has considered a trust factor when we have to select one of directly trusted neighbors of user u to continue the random walk to that user u in the trust network. More specifically, the higher trust factor a user u has, the more likely it is to be selected in a random walk. It is more tally with the actual situation that the probability of visiting a user u in a random walk of trust network where each random step is jumping to a next directly trusted neighbours of user u with uniform probability.

The remainder of this paper is organized as follows. In Section 2, we provide some related work. Section 3 describes the problem definition we study in this paper. Section 4 details our proposed method. Some desirable properties of our method is presented in Section 5. The results of an empirical analysis are presented in Section 6, followed by the conclusion and future work in Section 7.

2. RELATED WORK

In this section, we review several ways to study trust-based recommendations.

Trust-based recommender system [2] is an emerging field to provide users personalized item recommendations based on the historical ratings given by users and the trust relationships among users. The intuitions are that users tend to adopt items recommended by trusted friends rather than some strangers, and that trust is positively and strongly correlated with user preference. It has been reported that trust-based recommender systems can alleviate many issues from which traditional systems suffer, such as data sparsity and cold start [3].

In a broad view, trust in recommender systems can be divided into two common types: explicit trust and implicit trust. Explicit trust refers to the trust information explicitly specified by users in the systems. For example, users in FilmTrust [4] can directly add others as trusted neighbors. Many explicit trust-based recommender systems have been proposed [2,3,4] and their effectiveness has been empirically demonstrated. On the other hand, implicit trust is generally inferred from user behaviors such as user ratings rather than specified by users. By analyzing the value that is conveyed *via* ratings given by users, it is possible to identify the valuable users who are trustworthy and whose ratings are useful for item recommendation. A number of studies have been conducted to interpret such a perspective, such as [5-7].

TidalTrust [8] is a recommendation system that allows users to specify a degree of trust for each person in the network. It first searches shortest paths from a source user to a target user. After that, it backtracks from the target user, level by level, to the source user through previously searched strongest (the path that has the largest trust weight, *i.e.*, most reliable) shortest paths. The trust is then accumulated over neighbors of varying distance to create a ranked list approach. The top ranked items are then presented to the user, and are the only ones used in the evaluation. Since TidalTrust only uses information from raters at the nearest distance, it may lose a lot of valuable ratings from users a little further apart in the network.

MoleTrust [9] is a depth-first graph walking algorithm with a tunable trust propagation horizon that allows us to control the distance to which trust is propagate. It is similar to TidalTrust, since they both work in a breadth first search fashion. One difference is that in MoleTrust the trust propagation horizon is an input parameter and hence it is tunable so that it is possible to test how different levels of locality in the trust propagation affect accuracy and coverage of the trust metric. [2] chooses MoleTrust as a local trust metric to propagate trust allows users trusted by trusted users or even further away users, to be considered as possible neighbours.

One popular trust metric is the one used by open-source developer community website Advogato[10]. The Advogato metric is based on network flow and decides which members appear to be trustworthy, competent open source developers. Since the number of users to trust is independent of users and items and there is no distinction between the trusted users, this approach is not appropriate for trust-based recommendation.

The PhD thesis of Ziegler concentrates on recommendation system from different points of research. About the integration of trust, he proposes a solution very similar to ours, *i.e.* neighbours formation by means of trust network analysis. He has designed a local trust metric, Appleseed [11], that computes the top-M nearest trust neighbours for every user. AppleSeed considers the trust to be additive. If there are many weakly trusted paths between two users, this pair of users will obtain a high trust value, which is not intuitive.

[12] develops a set of five natural axioms and show that no recommendation system can simultaneously expected to satisfy all the axioms. However, for any subset of four of the five axioms it exhibits a recommendation system that satisfies those axioms. A unique recommendation system based on random walks is obtained by replacing an axiom capturing a notion of transitivity with ones capturing trust propagation and duplication. [12] does not show any experimental evaluation or comparison to other methods.

[13] introduces the notion of trust in reference to the degree to which one might trust a specific profile when it comes to making a specific rating prediction. Two different trust models are developed, one that operates at the level of the profile and one at the level of the items within a profile. In both of these models trust is estimated by monitoring the accuracy of a profile at making predictions over an extended period of time. Trust then is the percentage of correct predictions that a profile has made in general (profile-level trust) or with respect to a particular item (item-level trust). The social network is extracted from the similarity of users' profiles is not providing additional information as is provided by trust network. Moreover, it does not use the transitivity of trust, but only directly trusted users. This method is not a trust-based recommendation method in the sense in which we use this term in this paper.

Our work combines the trust-based and collaborative filtering approaches for recommendation. Target users take a finite-step random walk on a trust network, so as to use the ratings by trusted users to assist prediction. We introduce a trust factor for neighbor selection when we have to select one of directly trusted neighbors of user u to continue the random walk to that user u in the trust network. The idea behind our work is the higher trust factor user u has, the more likely it is to be selected in a random walk.

3. MODEL DESCCRITION

There exit an intrinsic challenge in trust-based recommendation is to decide how far to go while exploring the network. In addition, we also need to consider the trade-off between precision and coverage: the further we go, the more likely to find raters, but the less trust-worthy their ratings we get. Our approach initiated from the following observation: Ratings issued by strongly trusted friends on similar items are more reliable than those issued by weakly trusted far neighbors on the exact target item. This motivates us to combine trust-based and item-based approach.

In this work we present a random walk model, called RandomTrustWalker, which considers both ratings of the target item and similar items. The probability of adopting the rating of a similar item instead of a rating for the target item increases with increasing walking distance. Our model consists of two basic parts: the random walk on the trust network and the probabilistic item selection. The random walk carries out the search in the trust network, and the item selection part considers ratings on similar items to prevent from going too deep in the network. Based on the above work, the precision and coverage are improved respectively by preferring raters at a nearer distance and considering similar items as well as the exact target item.

To predict a rating on a target item i for a given user u_0, we perform random walks on the trust network, each starting at u_0 to find a user having expressed rating for i or that items similar to i. Each random walk returns a rating. The aggregation of all ratings returned by different random walks is considered as predicted rating through several random walks.

Suppose we are given a set of users $U=\{u_1, u_2,\ldots, u_n\}$ and a set of items $I=\{i_1, i_2,\ldots, i_m\}$. The set of items that have been rated by user u is denoted by $I_u = \{u_{i_1}, u_{i_2}, \ldots, u_{i_k}\}$. The user's rating on item i can be represented by $r_{u,i}$. $r_{u,i}$ can be any real number, but often ratings are integers ranging between 1 and 5. We also have a trust network among users in a trust-based recommender system. If u trusts v, then $t_{u,v}$ indicates the value of this trust as a real number ranging from 0 to 1. 0 means no trust and 1 means full trust. Binary trust network are the most common trust networks. We define $TU_u = \{v \in U \mid t_{u,v} = 1\}$ where TU_u represents the set of users directly trusted by u. We have the definition of trust network as a graph $G=<U, TU>$ where $TU = \{(u,v) \mid u \in U, v \in TU_u\}$. There is a node corresponding to each user, and an edge corresponding to each trust statement.

The task of recommender system in this work is predicting the rating $\overline{r}_{u,i}$ for a source user $u \in U$ on a target item $i \in I$. Normally, users rate only a very small percentage of the items, and $r_{u,i}$ is unknown for most pairs of (u, i). In particular, recommender systems based on collaborative filtering estimate $\overline{r}_{u,i}$ rely on the ratings expressed by similar users. Basically, they try to find a neighborhood of raters who have a rating profile similar to the source user, and aggregate their ratings. In trust-based recommender systems, we use the trust network to define the neighborhood instead of rating similarities. To predict a rating we visit our directly trusted neighbors whether they know the rating for the item. If it is, then they return it. Otherwise, they visit recursively their direct neighbors. The neighborhood in trust-based recommender system is described as the set of raters trusted by the source user. The ratings from these rates are aggregated to generate a recommendation.

Trust-based recommendation works based on the effects of selection and social influence that have been postulated by sociologists for a long time. Selection means that people are inclined to relate to people with similar attributes, and due to social influence related people in a social network influence each other to become more similar. The increasing availability of online social network data has finally allowed a verification of these sociological models. [15] experimentally verified that people are similar to their neighbors in a social network for these reasons. They had a network of people having social interactions and a similarity network in which users are connected to their most similar users. It was shown that the social interaction and similarity graphs have little overlap, sharing fewer than 15% of their edges in common. The results of [14] and of similar work confirm that a social network provides an independent source of information which can be exploited to improve the quality of recommendations.

Exploiting the trust network in recommenders does not necessarily improve the precision of system, but it allows to compute the recommendation for more pairs of (u, i) which results in better coverage. Coverage refers to the number of target user-target item pairs for which a prediction can be generate. This information can in particular help to generate recommendations for cold start users. Moreover, using a trust network will protect the recommender system against attacks like fake profiles. As faked profiles are not being trusted, they cannot affect the recommender.

3.1. Random Trust Walk

Following the motivation provided in the previous section, we now formally define the basic setting of RandomTrustWalker. In the remainder of the paper, we refer to such model simply as RTW, for brevity.

Starting from source user u_0, we perform our random walk. At each step k of a random walk, we are at a certain node u. If u already has the rating on target item i, then we stop our random walk and return $r_{u,i}$ as the result of random walk. If u does not have a rating on i, then we have the following two options:

Firstly, with probability $\alpha_{u,i,k}$, if we decide not to continue the random walk. We stay at node u and randomly select one of the items j similar to the target item i rated by u and return $r_{u,j}$. The idea is that we define a similarity measure between items, and for each item $j \in I_u$, we assign a probability of selecting proportional to the similarity of i and j. We'll discuss the details of the similarity metric later.

$$\Pr\left(\sigma_{u,i} = j\right) = \frac{sim_{i,j}}{\sum_{h \in I_u} sim_{i,h}} \quad (1)$$

where $\sigma_{u,i}$ is the random variable for selecting item j amongst items rated by u while looking for an item similar to target item i. we return $r_{u,j}$ as the result of this random walk.

Secondly, with probability $1-\alpha_{u,i,k}$, if we decide to continue the random walk at node *u*, we have to select one of *u*'s directly trusted neighbors *v* ($v \in TU_u$) to continue the random walk. We define $\chi_{u,v}$ as a trust factor for selecting a user *v* from TU_u:

$$\chi_{u,v} = \left(0.5 + 0.5 * sim(u,v)\right) * \frac{1}{1+e^{-IN_v}} \tag{2}$$

$$IN_v = \sum_{w \in U \wedge w \neq v} t_{w,v} \tag{3}$$

This intuition is inspired by [15]. To each user *v* we associate a trusted value IN_v, which is in-degree of the *v* trusted directly by other users in the trust network. The trusted value is indicated as the importance of other users to the user *v*. If an edge from user *u* to user *v* represents a trust statement expressed by *u* to *v*. The *sim*(*u*,*v*) is a measure of similarity between user *u* and *v*, which is traditionally calculated as Pearson's correlation coefficient. Values of $\chi_{u,v}$ are in the range of [0, 1].

Now, we have:

$$\begin{aligned} \Pr\left(\beta_{u_0,i,k+1} = v \mid \beta_{u_0,i,k} = u, \overset{\nwarrow\searrow}{R}_{u,i}\right) \\ = \left(1-\alpha_{u,i,k}\right) \times \Pr\left(\chi_{u,v}\right) \end{aligned} \tag{4}$$

$$\Pr\left(\chi_{u,v}\right) = \frac{\chi_{u,v}}{\sum_{z \in TU_u} \chi_{u,z}} \tag{5}$$

Here, $\beta_{u_0,i,k}$ denotes the random variable for being at node *v* in step *k* while looking for a prediction on target item *i* for source user u_0. Details of computing $\Pr\left(\beta_{u_0,i,k} = v\right)$ are discussed later. Also we have a condition that the user u in step *k*-1 does not have the rating for item *i* (denoted by $\overset{\nwarrow\searrow}{R}_{u,i,k}$). The probability of walking from user *u* to *v* is independent of previous steps. But, since $\beta_{u_0,i,k}$ depends on the step *k*, it is not independent of the step of the random walk.

3.1.1. Item Similarity

In memory-based recommendation, the similarity of items can be computed using their features. However in collaborative filtering, the only information available about items is their ratings. Hence, to compute the similarity of two items, we use the Pearson Correlation of ratings expressed for both items, as used in [16]. Values of the Pearson correlation are in the range [-1, 1]. Negative correlations mean that the ratings expressed for two items are in opposite directions, so these items are not useful for our purpose. Therefore, we only consider items with positive correlation.

$$corr_{i,j} = \frac{\sum_{u \in UC_{i,j}} \left(r_{u,i} - \bar{r}_u\right)\left(r_{u,j} - \bar{r}_u\right)}{\sqrt{\sum_{u \in UC_{i,j}} \left(r_{u,i} - \bar{r}_u\right)^2} \sqrt{\sum_{u \in U} \left(r_{u,i} - \bar{r}_u\right)^2}} \tag{6}$$

$UC_{i,j}$ is the set of common users who have rated both items *i* and *j*, and $\bar{r}_u$ denotes the average of ratings expressed by *u*. $corr_{i,j}$ denotes the correlation of items *i*,*j*. The size of the set of common users is also important. For example, if $corr_{i,j} = corr_{i,l}$, but $|UC_{i,j}| > |UC_{i,l}|$, then, since *i* and *j* have been rated by more common users, so the correlation between them is stronger and *sim*(*i*, *j*) should be greater than *sim*(*i*, *l*). We consider $|UC_{i,j}|$ in the similarity measure as follows:

$$sim_{i,j} = \frac{1}{1+e^{-\frac{|UC_{i,j}|}{2}}} \times corr_{i,j} \tag{7}$$

We used the sigmoid function to avoid favoring the size of $UC_{i,j}$ too much and to keep the similarity value in the range [0,1]. If the size of the set of common users is big enough, then the first part of equation 7 would converge to 1, but for small sets of common users, the factor would be 0.6. The number 2 in the denominator of the exponent is because we wanted to have a factor of greater than 0.9 if the size is greater than 5.

3.1.2. Termination of a Single Random Trust Walk

At each user *u*, we have a probability $\alpha_{u,i,k}$ of staying at *u* to select one of his items at step *k* of the random walk, while we are looking for a prediction on target item *i*. This probability should be related to the similarities of items rated by *u* and the target item *i*. Similarity values are real numbers in [0, 1], so they can also be considered as probabilities. We consider the maximum similarity of items rated by *u* with target item *i* as the probability of staying at node *u*.

Furthermore, ratings that on target item *i* from users far away from source user u_0 are noisy, but ratings expressed by trusted users nearby in the network are more reliable. So, the deeper we go into the network, the probability of continuing our random walk should decrease and so $\alpha_{u,i,k}$ should increase.

To inject the factor *k* in $\alpha_{u,i,k}$, we should use a function *f* (*k*) which gives value 1 for big values of *k*, and a small value for small values of *k*. Since the sigmoid function satisfies our constraints for *f* (*k*), we consider a sigmoid function of the *k* as another factor affecting $\alpha_{u,i,k}$.

$$\alpha_{u,i,k} = \max_{j \in I_u} sim_{i,j} \times \frac{1}{1+e^{-\frac{k}{2}}} \tag{8}$$

Each random walk has three alternatives to stop:

1. Reaching a node which has expressed a rating on the target item *i*.

2. At some user node *u*, we decide to stay at the node and select one of the items rated by *u* and return the rating for that item as the result of random walk.

3. There is chance for a single random walk to continue for ever. To avoid such a case in our implementation of random walk, we terminate the random walk when we go very

far from the source user (k > max-depth). Based on the idea of "six-degrees of separation"[17], we set max-depth = 6.

3.2. Recommendation

In RTW, we have the probability of selecting items rated by different users and returning that rating as the result of a random walk. These items could be either the exact target item i, or another item. The estimated rating for source user u on target item i would be the expected value of ratings returned by different random walks.

$$\hat{r}_{u,i} = \sum_{\{(v,j)|R_{v,j}\}} \Pr\left(\alpha\sigma_{u,i} = (v,j)\right) r_{v,j} \tag{9}$$

In the above equation, $\alpha\sigma_{u,i}$ is the random variable for stopping the random walk at node v and selecting item j rated by v, while we start the random walk from source user u looking for target item i. Notice that the value for $\alpha\sigma$ are ordered pairs. As used before, $R_{v,j}$ is a boolean variable denoting whether v has a rating on item j. Now we have:

$$\Pr\left(\alpha\sigma_{u,i} = (v,j)\right) = \begin{cases} \Pr\left(\beta_{u,i} = v\right)\alpha_{v,i} \Pr\left(\sigma_{u,i} = j\right) & v \neq u; i \neq j \\ \Pr\left(\beta_{u,i} = v\right) & v \neq u; i = j \\ \alpha_{v,i} \Pr\left(\sigma_{u,i} = j\right) & v = u; i \neq j \end{cases} \tag{10}$$

In this equation, $\beta_{u,i}$ is the random variable for being at node v at some step in a random walk starting from source user u looking for item i. Notice that in above formula, we used $\beta_{u,i}$ instead of $\beta_{u,i,k}$ for the first case. Since we don't know the number of steps needed to reach v, we don't consider the factor k (Actually $\beta_{u,i} = \beta_{u,i,\infty}$). It should be noted that if we actually perform random walks, we can consider the step k in the first case. But to have a closed form formula, we ignore the factor k at the last user v which gives us a pretty good approximation of the probability. Also, we should note that the case $v = u$ and $i = j$ is trivial since the user himself has the rating on the target item.

A random walk starting from u can reach v using different number of steps. As mentioned before, we use random variable $\beta_{u,i,k}$ for being at node v in k steps

$$\begin{aligned} &\Pr\left(\beta_{u_0,i,k+1} = v\right) \\ &= \sum_{w \in U} \Pr\left(\beta_{u_0,i,k-1} = w\right)\left(1 - \alpha_{w,i,k}\right)\Pr\left(\chi_{w,v}\right) \end{aligned} \tag{11}$$

Also we have $\Pr\left(\beta_{u,i,0} = u\right) = 1$ as the base for the above equation. Since the random walks have a probability of stopping at each step $\sum_{v \in U} \Pr\left(\beta_{u,i,k} = v\right) \neq 1$ To make $\Pr\left(\beta_{u,i,k} = v\right)$ a probability distribution, we define a dead state $\perp$ to which all users go after deciding to terminate that random walk. So, we have

$$\Pr\left(\beta_{u,i,k} = \perp\right) = 1 - \sum_{v \in U} \Pr\left(\beta_{u,i,k} = v\right) \tag{12}$$

This state $\perp$ will be added to U for convenience in formalization of our method, but we don't consider this state in any actual random walk. Now, we can compute $\Pr\left(\beta_{u,i} = v\right)$ as follows:

$$\Pr\left(\beta_{u,i} = v\right) = \frac{\sum_{k=1}^{\infty} \Pr\left(\beta_{u,i,k} = v\right)}{\sum_{w \in U} \sum_{k=1}^{\infty} \Pr\left(\beta_{u,i,k} = w\right)} \tag{13}$$

3.3. Termination of the Overall Method

The results of performing actual random walks approximate the results given by equation (14). We perform several random walks to be able to get a more reliable prediction. We need to be able to decide when we have done enough random walks to have a precise estimate of $\hat{r}_{u,i}$. We compute the variance in the results of all the walks as follows:

$$\varepsilon^2 = \frac{\sum_{i=1}^{T}\left(r_i - \bar{r}\right)^2}{T} \tag{14}$$

Here, r_i is the result of i^{th} random walk, and $\bar{r}$ denotes the average of the ratings returned by random walks. T is the number of random walks we perform to compute the prediction. We also define ε_i^2 as the variance in the results of the first i random walks. Since the values of ratings are infinite range of [1, 5], it can be proved that ε^2 converges to a constant value. So we can terminate RTW if $\left|\varepsilon_{i+1}^2 - \varepsilon_i^2\right| \leq \zeta$.

It should be noted that we have a constant threshold of 10000 for the maximum number of unsuccessful random walks, and after that we consider the pair <user, item> as non-covered.

4. PROPERTIES OF RTW

Our model includes Item-based Collaborative Filtering and pure Trust-based Recommendation as its extreme special cases. If $\alpha_{u,i,k} = 1$ for all $u \in U$, then our random walk will never start, and it will return the rating expressed by the source user u_0 on one of its rated items. Since the probability of selecting an item is proportional to its similarity to the target item i, the expected value of the recommended rating would be the weighted average of the ratings on items in I_{u_0} with weights proportional to the similarities of these items to the target item i. This is the same as the result of Item-based collaborative filtering proposed in [17].

On the other hand, if we set $\alpha_{u,i,k} = 0$ for all $u \in U$, then all random walks will continue until they have found a rating for the exact target item i. The recommended rating would be the aggregation of ratings expressed by users having the rating on i weighted by the probability of reaching these users from u_0. Existing methods [2] [8] try to approximate these probabilities by simplifying the problem. So our RTW, in one of its extreme cases, can be considered as an ideal trust-based recommender.

Table 1. Summary statistics of the original data set.

Users	items	ratings	trust
40136	139738	664824	442979

In our model, to predict $\hat{r}_{u,i}$, we compute $\Pr(\beta_{u,i} = v)$ for all users *v*. The results of different random walks are from different user. The most frequent users are user with high probability of $\Pr(\beta_{u,i} = v)$. We can output these users as users whose ratings are most influential on the prediction.

Also considering ratings on some items are more frequently used in the results of different random walks. These items are items with high values of $\Pr(\alpha\sigma_{u,i} = (v, j))$.

Now we can use these most frequent users and items to explain why we predicted the rating with $\hat{r}_{u,i}$. We can explain to users that this prediction is based on ratings from these trusted users and these similar items.

5. EXPERIMENTS

In order to verify the effectiveness of proposed method, we employ the traditional collaborative filtering as the bench method and conducted experiments on a data set comparing various versions of RTW with the bench method. We implemented different kind of RTW which have different maximum walk step to confirm the influence of step. We also performed the bench method as two fundamental similarity based recommendation methods.

We implemented all methods proposed in Java. All experiments were performed on an Intel core 2 duo, 2.66Ghz, 4G Bytes of memory and an Windows 7 operating system.

5.1. Data Sets

We evaluate the methods described above using data from the Epinions data set. The reason why we choose this particular dataset is that it provides not only user ratings for items but also directional trust information while most data sets for recommendation have no trust info. According to discussion above, Epinions is the best suitable data set to our experiments.

The version of the Epinions data set, which we used, is very sparse. There are 664824 ratings in our data set which are expressed to 139738 items form 40136 users. So we can compute the spares rate is close to 0.01%. In addition, we have 442979 trust statements among pairs of users. We consider those users as cold start users with less than 5 ratings. As mentioned above, 47% of users, which is a huge portion of users, are cold start users. We split the original dataset into two data sets. The set DS1 is separated by random sampling and the other set DS2 includes completely with cold start users. The percentages of test set and training set in both datasets are nearly 5:1. The detailed statistics of original data set is illustrated in Table **1**.

5.2. Experiment Design

To verify the effectiveness of the proposed method, we designed a series of measure to compare the different methods and the same method with different value of parameters. The details of experiment were designed as the following:

1. Finding the most appropriate step of random walk in the trust network. We set the step of random walk to be 1 and 6 respectively, and a random value depending on the property of $\alpha_{u,i,k}$.

2. Determining the value of ζ. We set ζ to be 0.01, 0.005, 0.001 and 0.0005 respectively. This way, we can find the more appropriate ζ that makes RTW has lower error than all the other methods.

3. Comparing RTW with traditional rating-oriented approaches such as user-based and item-based algorithms. We used the reinforcing Pearson Correlation as similarity metric, and DS1 and DS2 we used can observe the efficiency of RTW on different conditions.

4. Evaluating the effect of similarity metric. We compared the MAE of methods using reinforcing Pearson Correlation and traditional Pearson Correlation respectively on DS1 and DS2.

5. Evaluating the effect of trust factor $\chi_{u,v}$, we used $\Pr(\chi_{u,v})$ and $1/|TU_u|$ separately to select a directly trusted neighbor *v* from TU_u during the random walk in two different methods.

5.3. Evalutation Metric

The major criterion for evaluating traditional rating-oriented collaborative filtering algorithms is the rating prediction accuracy. Commonly used measures for accuracy include the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE), both of which depend on difference between true rating and predicted rating. MAE penalizes each miss by the distance to actual rating and RMSE emphasizes large errors compared to MAE measure. We employ the MAE to measure the error in recommendation, the metrics MAE is defined as:

$$MAE = \frac{\sum_{i=1}^{N} \left| r_{u,i} - \hat{r}_{u,i} \right|}{N} \tag{14}$$

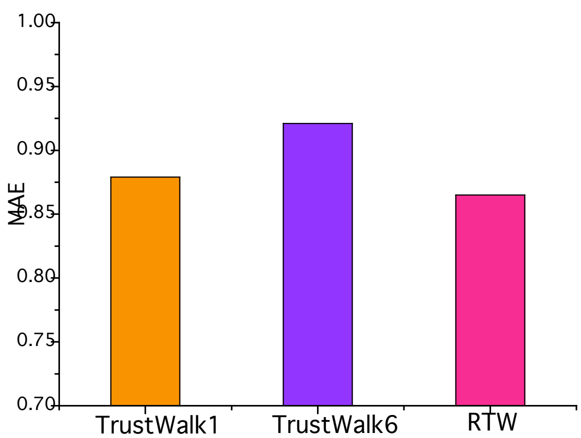
where *N* is the number of predicted ratings, $r_{u,i}$ and $\hat{r}_{u,i}$ denotes the actual and recommended rating for user *u* on item *j* respectively. Thus, the smaller the value of MAE means better predictive accuracy.

5.4. Data Sets Comparision of Algorithms

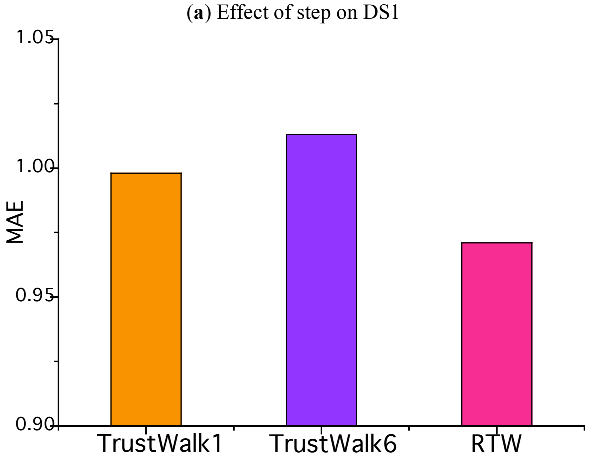
We compared the results for different methods. Following is the description of labels we use to denote each of these methods:

- User based. We performed the user based collaborative filtering, with the reinforcing Pearson Correlation and traditional Pearson Correlation as similarity metric.
- Item based. We also performed the item based collaborative filtering using two similarity measure the same as user based.
- TrustWalker1. This is a kind of TrustWalker methods who must have one step walk in the trust network unless the user has no trust users.
- TrustWalker6. This method is a version of TrustWalker methods and similar to TrustWalker1. What the different between two methods is walk step. In this method, it will walk six steps with $\alpha_{u,i,k} = 0$.
- RTW. This is the full method which we discuss above.

We also chose different values of ζ and different similarity measures to verify he effectiveness of the Random-Walk method under different conditions. We present the results of our experiments as following figures and tables.



(**a**) Effect of step on DS1



(**b**) Effect of step on DS2

Fig. (1). Comparison of MAE on different data sets with the parameter $\zeta = 0.001$.

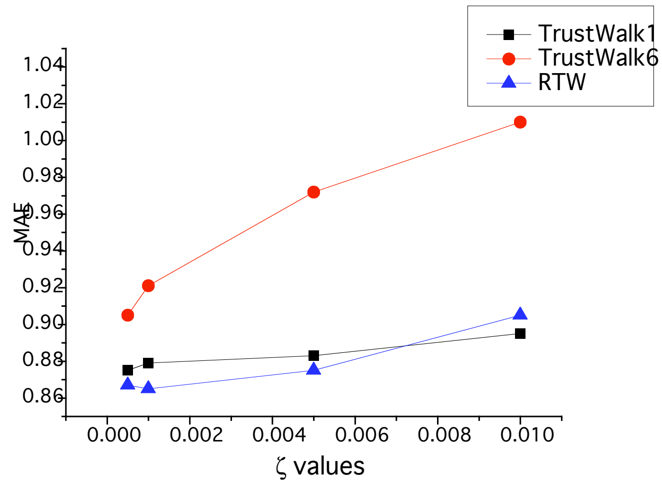


Fig. (2). Comparison effect of ζ for different methods on DS1.

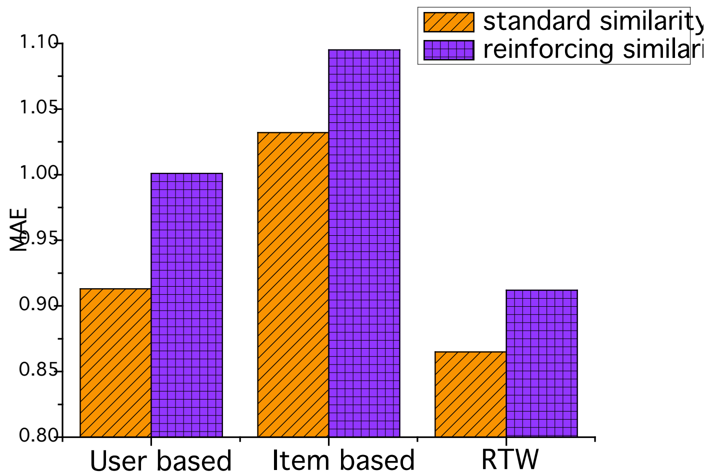
Let us now compare performances of TrustWalk1, TrustWalk6 and RTW on DS1 and DS2. We can see that RTW has a random value of step depending on the property of $\alpha_{u,i,k}$ outperforms other methods in Fig. (**1**), especially on DS2 included totally with cold start users. The MAE achieved by TrustWalk1, TrustWalk6 and RTW is respectively 0.879, 0.921 and 0.865 on DS1. And the MAE is 0.998 for TrustWalk1, 1.013 for TrustWalk6 and 0.971 for RTW on DS2. We observe RTW shows the best result on both DS1 and DS2. So, the improvement on RTW's step is helpful to make recommendation better.

The effect of ζ to trust walk methods is shown in Fig. (**2**). MAE decreased obviously as the value of ζ decreases when ζ is greater than 0.001. However, with the decreasing of ζ further, we observe MAE no longer changes significantly in RTW and TrustWalk1. Moreover, more time need to be spent in making ε^2 convergent. The MAE of TrustWalk6 is 0.905 when ζ is 0.005 that is higher than the other two methods when ζ is less than 0.001. So, the termination condition ζ we set is 0.001 in our experiments.

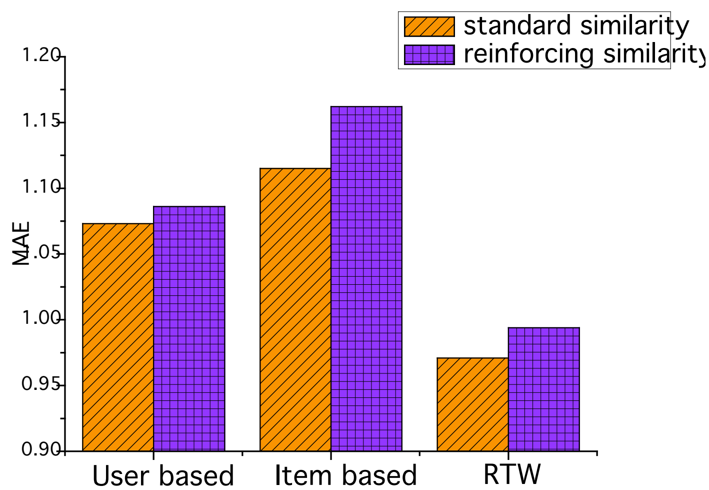
The comparisons for all methods on DS1 and DS2 are exhibited in Table **2**. The MAE of TrustWalk1 and User based on DS1 are close to RTW while RTW has the best performance on DS2. It shows that RTW alleviates the cold start problem availably.

Table 2. Experimental results for all methods on DS1 and DS2 with the parameter $\zeta = 0.001$ and step depending on the property of $\alpha_{u,i,k}$

Methods	DS1	DS2
TrustWalk1	0.879	0.998
TrustWalk6	0.921	1.013
RTW	0.865	0.971
User based	0.913	1.073
Item based	1.032	1.115



(**a**) Effect of step on DS1



(**b**) Effect of step on DS2

Fig. (3). Comparison effect of ζ for different methods on DS1.

Fig. (**3**) shows the comparison of methods using different similarity measures. It shows that methods with low MAE using reinforcing similarity. This can also be see that there are no obvious improvement on DS2 comparing with on DS1. Notice that the most improvement of MAE is only 0.047 in item based method, and it has changed 0.023 and 0.013 in the method of user based and RTW respectively. This is because DS2 is composed of cold users, similarity metric just have a less effect to overall method. We can confirm that the reinforcing Pearson Correlation is useful to decrease the error of recommendation substantially.

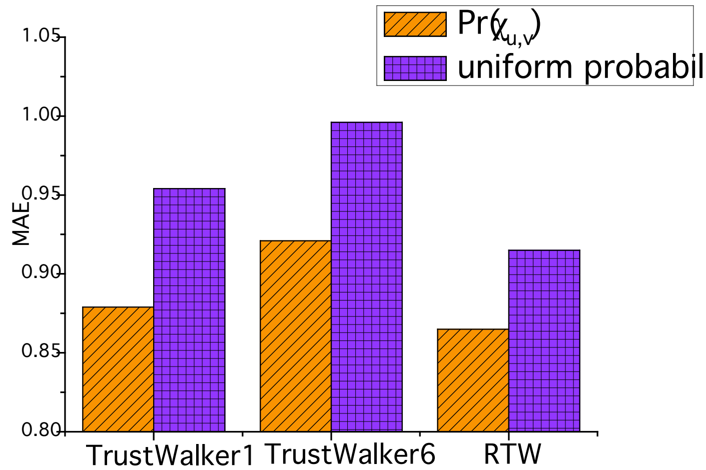
Fig. (**4**) shows the effect of the trust factor $\chi_{u,v}$ for all methods. This result confirms our expectation is reasonable, the user *v* who is one of *u*'s direct trusted neighbors has a high similarity between u and v and is more trusted by other users in the trust network should get more chance to be selected rather than selected with uniform probability in random walk.

In the overall, random walk combined with trust factor outperforms the other methods.

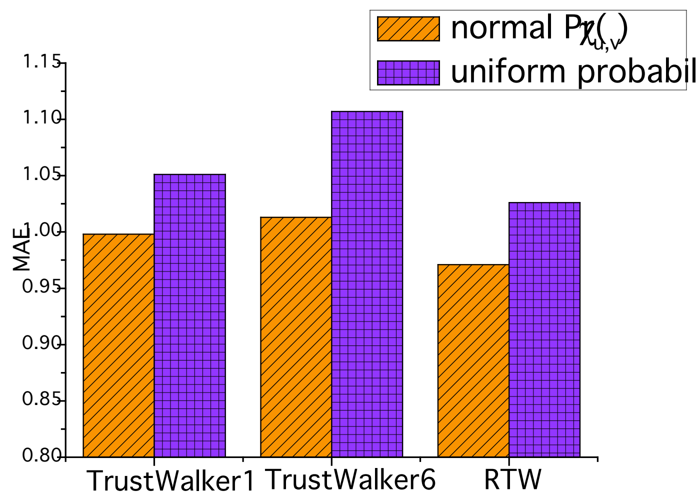
CONCLUSION

With the information available to us growing far more rapidly than our ability to process it, technologies to help people sift through huge amount of information efficiently is becoming increasingly important in order to overcome the resulted information overload problem. Recommender systems have been applied in many fields to provide useful, personal, and high-quality recommendations. The most popular applications are the systems which were built with collaborative filtering. However, it cannot present a well-pleasing result when there are a lot of cold start users who have only rated a tiny fraction of the available items. In addition, the current methods have not taken advantage of the confidence between users.

In this paper, we proposed and experimentally verified a random walk method using trust factor have an important role to play in collaborative filtering. From the experiment results, we can observe that the proposed approach outperforms traditional collaborative filtering algorithms. We considered not only user ratings for items but also directional trust info between users. This method is especially useful



(**a**) Effect of step on DS1



(**b**) Effect of step on DS2

Fig. (4). Comparison effect of trust factor for different methods.

when training data is extremely sparse. Moreover, the RTW can better deal with cold start users, since user only need to simply connected to the trust network.

Many other directions should be considered for generating a better recommend model in the future work. Firstly, we found the efficiency of RTW is not fast enough to support the real-time recommend under the condition upon traditional single computer base. So, it is necessary that extend the method to distributed environments. Subsequently, we plan to use integral variable to describe the trust relationship instead of Boolean variable. Eventually, we also want to combine the time factor to do more in-depth study because interest of people is changing with time.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

The research of this work is supported by the National Science Foundation of China under grants No. 60803086 and the Natural Science Foundation of Beijing under grants No. 4153058.

REFERENCES

[1] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry. "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol.35, pp.61-70, 1992.

[2] P. Massa and P. Avesani. "Trust-aware recommender systems," In Proceedings of the 2007 ACM Conference on Recommender Systems, pp.17-24, Minneapolis, MN, USA, 2007. *ACM.*

[3] G. Guo, J. Zhang, and 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 (UMAP), pp. 114-125, Montreal, Canada, 2012.

[4] J. Golbeck and J. Hendler. "Filmtrust: Movie recommendations using trust in web-based social networks," In Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC) pp. 282-286, Las Vegas, NV, USA, 2006.

[5] N. Lathia, S. Hailes, and L. Capra. "Trust-based collaborative filtering," In IFIPTM 2008: Joint iTrust and PST Conferences on Privacy, Trust Management and Security, Trondheim, Norway, 2008.

[6] M. Papagelis, D. Plexousakis, and T. Kutsuras. "Alleviating the sparsity problem of collaborative filtering using trust inferences," In iTrust'05: Proceedings of the Third international conference on Trust Management, pp.224-239, Paris, France, 2005.

[7] Q. Shambour and J. Lu. “A trust-semantic fusion-based recommendation approach for e-business applications,” *Decision Support Systems*, vol.54, pp.768-780, 2012.

[8] J.Goldberg. “Computing and applying trust in web-based social network,” PhD thesis, University of Maryland College Park, MD, USA, 2005.

[9] P. Massa and P. Avesani. “A Trust-enhanced Recommender System application: Moleskiing,” In Proceedings of the 2005 ACM Symposium on Applied Computing, pp.1589-1593, Santa Fe, New Mexico, USA, 2005, ACM.

[10] Levien and Aiken. Advogato's trust metric. Online at http://advogato.org/trust-metric.html, 2002.

[11] C.N.Ziegler. “Towards decentralized recommender systems,” PhD thesis, University of Freiburg, Germany, 2005.

[12] R. Andersen, C. Borgs, J. Chayes, U. Feige, A. Flaxman, A. Kalai, V. Mirrokni, and M. Tennenholtz. “Trust-based recommendation systems: an axiomatic approach,” In Proceedings of the 17th International Conference on World Wide Web (WWW), pp.199-208, Beijing China, 2008.

[13] J. O'Donovan and B. Smyth. “Trust in recommender systems,” In Proceedings of 10th international conference on intelligent user interfaces, pp.167-174, San Diego, California, USA 2005.

[14] D.Crandall, D.Cosley, D.Huttenlocher, J.Kleinberg, and S.Suri. “Feedback effects between similarity and social influence in online communities,” In KDD’08: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp.160-168, Las Vegas, Nevada, USA, 2008, ACM.

[15] M. Jamali, M. Ester. “Trust Walker: a random walk model for combining trust-based and item-based recommendation,” In KDD’09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 397-405, Paris, France, 2009, ACM.

[16] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. “Item-based collaborative filtering recommendation algorithms,” In WWW ’01: Proceedings of the 10th international conference on World Wide Web, pp.285-295, New York, NY, USA, 2001, ACM.

[17] S. Milgram. “The small world problem. Psychology Today,” vol.1, pp. 61-67, 1967.

Received: September 16, 2014 Revised: December 23, 2014 Accepted: December 31, 2014