Using a Trust Network to Improve Top-N Recommendation

Mohsen Jamali
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
mohsen_jamali@cs.sfu.ca

Martin Ester
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
ester@cs.sfu.ca

ABSTRACT

Top-N item recommendation is one of the important tasks of recommenders. Collaborative filtering is the most popular approach to building recommender systems which can predict ratings for a given user and item. Collaborative filtering can be extended for top-N recommendation, but this approach does not work accurately for cold start users that have rated only a very small number of items. In this paper we propose novel methods exploiting a trust network to improve the quality of top-N recommendation. The first method performs a random walk on the trust network, considering the similarity of users in its termination condition. The second method combines the collaborative filtering and trust-based approach. Our experimental evaluation on the Epinions dataset demonstrates that approaches using a trust network clearly outperform the collaborative filtering approach in terms of recall, in particular for cold start users.

Categories and Subject Descriptors

H.2.8.d [**Information Technology and Systems**]: Database Applications - Data Mining

General Terms

Algorithms, Design, Measurement, Experimentation

Keywords

Trust, Recommendation, Top-N

1. INTRODUCTION

With the rapidly growing amount of information available on the WWW, it becomes necessary to have tools to help users to select the relevant part of online information. To satisfy this need, recommender systems have emerged, e.g. there are popular recommenders for movies[1], books[2], music[3], and so on.

[1]http://www.netflix.com
[2]http://www.amazon.com
[3]http://www.last.fm

Typically in a recommender system, we have a set of *users* and a set of *items*. Each user u rates a (small) subset of the set of all items with some numeric score, e.g. on a scale from 1 to 5. The recommender system has to predict the unknown rating for source user u on a non-rated target item i based on the known ratings. Collaborative filtering [3] methods make recommendations based on the ratings of item i by a set of users whose rating profiles are most similar to that of user u. This approach is most effective when users have expressed enough ratings to have common ratings with a good number of other users, but it performs poorly for so-called cold start users, i.e. new users who have expressed only very few ratings. This is due to the fact that cold start users, having only a few ratings, are unlikely to have users with similar rating profiles.

With the advent of online social networks, the trust-based approach to recommendation has emerged. This approach assumes a trust network among users and makes recommendations based on the ratings of the users that are directly or indirectly trusted by u. Trust-based recommenders can make recommendations as long as a new user is connected to a large enough component of the trust network. Therefore, trust-based methods tend to outperform collaborative filtering methods for cold start users.

Predicting the rating of user u on target item i is not the only task of a recommendation system. Often a user does not request the prediction of the score for a particular item, but wishes to obtain a ranked list of items that he has not yet rated but is likely to rate highly. This problem definition may be more natural in particular in the common scenario that there is a very large number of items and that the user is not aware of all of the existing items. This task is referred to as *top-N item recommendation* in the literature[1][6][7].

Collaborative filtering methods, first proposed for predicting ratings of single items, have been extended for top-N recommendation[1][6][7], but they inherit the weaknesses of these methods for cold start users.

In this paper, we explore the trust-based approach to top-N item recommendation in order to improve the recommendation quality, in particular for cold start users. The first method proposed extends the random walk method that we recently introduced [4]. The random walk method is accurate for predicting ratings of single items, since the random walks continue until ratings for the target item or similar items are found. However, it is not as effective for top-N item recommendation, since the direct and indirect neighbors in the trust network often have many highly rated items, other than the items actually rated highly by the source user, and the random walk likely will not reach ratings for the actual top-N items further away in the trust network. Therefore, we propose a second method for top-N item recommendation which combines the trust-based and the collaborative filtering approach, performing a weighted merge of the results from both approaches. This method combines the strength of collaborative filtering, higher density of the neighborhood for normal users, with the strength of trust-based recommendation, good performance for cold start users.

The rest of the paper is organized as follows: Section 2 reviews some related work. Preliminaries and problem definition are introduced in section 3, as well as baseline methods which extend user-based and item-based collaborative filtering approaches for top-N recommendation. In section 4, we propose novel methods to exploit the trust network. Our experimental results are discussed in section 5, and section 6 concludes the paper with a summary and some directions for future research.

2. RELATED WORK

Most state of the art methods for recommendation address the problem of predicting a single rating for a user. However, there has also been some work on top-N recommendation, which is reviewed in this section. [1] and [5] deal with Boolean ratings (indicating whether an item was purchased), while [6], [7] and [9] deal with numeric ratings (integer-valued scores). All of these methods adopt the collaborative filtering approach, none of them explores the trust-based approach. Generally, collaborative filtering approaches find a neighborhood of similar users and rank the items rated by these users to perform top-N recommendation. In the second subsection, we discuss trust-based approaches for the prediction of a single rating.

2.1 Top-N Recommendation

[1] is one of the first works addressing the problem of top-N recommendation. They extend the item-based collaborative filtering method[11] and present two alternative item-to-item similarity measures. The first one models the items as vectors in the user space and uses the cosine function to measure the similarity, whereas the second one uses a technique based on the conditional probability between two items. The second measure can differentiate between users with varying amounts of historical information as well as between frequently and infrequently purchased items. To perform the actual recommendation, the top similar items are computed for each item purchased by the target user, items are ranked according to the frequency of appearing in the set of similar items for different items purchased, and the top-ranked N items are returned. Intuitively more similar items should have more impact on the recommended items, but this is not supported. In addition, this method cannot handle cold start users well, since these users have purchased only few items which do not provide enough information to compute the top-N recommended items.

[5] also employs item-based collaborative filtering for top-N recommendation and introduces various evaluation metrics for top-N recommendation. Since they are using item-based collaborative filtering, they have similar issues with cold start users and similar items as the previous method.

[6] is another work exploring the item-based collaborative filtering approach, taking user feedback into account. They compute an error matrix recording the difference between the actual ratings and the predicted ratings. For a source user u and a target item i for which u has not expressed a rating, the predicted rating is computed by summation of the average error on predictions for u and the average error on predictions for i. The error matrix is updated upon receiving a new actual rating. This model is also unable to handle users and items with few ratings.

[7] builds a top-N recommendation method on top of existing recommendation systems for single ratings. They observe that the prediction accuracy decreases with increasing rating variance. Therefore, various methods are presented that filter out items with high rating variance. Since this approach reduces the diversity of items among the recommendations over all users, another method is proposed which does not filter out, but only penalizes high variance items. Variance-based filtering and weighting can be applied to any top-N recommendation algorithm including memory-based or model-based collaborative filtering (which are considered in [7]) and the new methods presented in this paper.

[9] has exploited user-based collaborative filtering to perform top-N recommendation. They introduce the Belief Distribution Algorithm that computes the belief (distribution) of rating differences instead of point estimates of the rating as done by existing methods. They estimate the belief difference between each user's average rating and the estimated rating on the items. The predicted belief difference for the source user and a given item is computed and added to the source user's average rating to obtain the predicted rating. Finally, the items having one of the top-N predicted ratings are returned as the recommended items. This method is an extension of the collaborative filtering algorithm for top-N recommendation which is orthogonal to our work.

2.2 Trust-based Recommendation

In this subsection, we discuss trust-based methods for the recommendation of the rating of a single item.

TidalTrust[2] performs a modified breadth first search in the trust network to compute a prediction. Basically, it finds all raters with the shortest path distance from the source user and aggregates their ratings weighted by the trust between the source user and these raters. To compute the trust value between users u and v who are not directly connected, TidalTrust aggregates the trust value between u's direct neighbors and v weighted by the direct trust values of u for its direct neighbors.

[8] introduces MoleTrust. The ideas used in MoleTrust and TidalTrust are similar, but MoleTrust considers all raters up to a *maximum-depth* which is given as an input. *maximum-depth* is independent of any specific user and item. Also, to compute the trust value between u and v, a backward exploration is performed. It means that the trust value from u to v is the aggregation of trust values between u and users directly trusting v weighted by the direct trust values. It should be noted that TidalTrust[2] considers only paths with the shortest distance instead of all paths of length up to *maximum-depth*. Moreover, TidalTrust does not perform the backward exploration.

TrustWalker[4] has been introduced as a random walk method to combine the trust-based approach and item-based CF approach for predicting the rating of single items. TrustWalker performs random walks on the trust network to find ratings for the target items or similar items. The stopping criteria for a single random walk at a certain user(node) depends on the similarity of items rated by that user and the target item, and on the current step of the random walk. In this paper, we extend this work to perform top-N recommendation.

3. PRELIMINARIES

A recommender system assumes a set of users $U = \{u_1, \dots u_N\}$ and a set of items $I = \{i_1, \dots i_M\}$. Each user u rates a set of items $I_u = \{i_{u_1}, \dots i_{u_k}\}$. The rating of user u on item i is denoted by $r_{u,i}$. $r_{u,i}$ can be any real number, but more often ratings are integers, e.g. in the range $[1, 5]$. In a trust-based system, there is also a trust network among users. If u trusts v, then there is a value $t_{u,v}$ for this trust which is a real number in $[0, 1]$. Zero means no trust and one means full trust. Binary trust networks are the most common trust networks (Amazon[4], eBay[5], ...), and we use a binary trust matrix in this paper. We define $T_u = \{v \in U| \ t_{u,v} = 1\}$ where T_u denotes the set of users directly trusted by u. The trust network can be representd as a directed graph $G =< U, T >$ where $T = \{(u, v)| \ u \in U, \ v \in T_u\}$. Nodes correspond to users, and edges correspond to trust statements between pairs of users.

The common task of recommendation is as follows: Given a user $u \in U$ and an item $i \in I$ for which $r_{u,i}$ is unknown, predict a rating for u on item i. We call u the source user and i the target item. The predicted rating is denoted by $\hat{r}_{u,i}$. Typically, users rate only a very small percentage of the items, and $r_{u,i}$ is unknown for most pairs (u,i).

The problem of top-N item recommendation can be formalized as follows: Given a user u, recommend a set of items $\hat{I}_u$ where $|\hat{I}_u| \leq N$ and $\hat{I}_u \cap I_u = \emptyset$. Note that the result is a set, not a list, of items, i.e. the rank within the collection of resulting items does not matter.

Existing top-N item recommender systems find a neighborhood of similar users and rank the items rated by these users to perform top-N recommendation. In the following subsections, we present two extensions of collaborative filtering for top-N recommendation, which we will use as baseline methods in our experiments. We describe the approach for systems with integer value ratings similar to [9][7][6].

3.1 User-based CF for Top-N Recommendation

Using user-based collaborative filtering for top-N recommendation has been proposed in [9]. We use this approach as one of our baseline methods in this paper, replacing the standard Pearson correlation by a novel similarity measure introduced in [4]. In this subsection, we describe the details of the user-based collaborative filtering approach for top-N recommendation. In this approach, we first find the top K similar users to the source user u, denoted by N_u. The similarity measure we use is defined in equation 2. Each user $v \in N_u$ has rated a set of items I_v. We aggregate the list of items rated by similar users. Then we find the top-N highly ranked items in this aggregated list and return them as the top-N recommended items. In the aggregated list, the aggregated rating of each item i would be:

$$\hat{r_{cu,i}} = \frac{\sum_{v \in N_u, i \in I_v} sim_{u,v} \times r_{v,i}}{\sum_{v \in N_u, i \in I_v} sim_{u,v}} \qquad (1)$$

In the above equation, $\hat{r_{cu,i}}$ is the predicted rating of item i for the source user u using CF. The top-N recommended items are the items with the top-N highest values of $\hat{r_{cu,i}}$. $sim_{u,v}$ is a metric which is related to correlation of ratings of users u and v. Similar to [4], we consider two factors affecting $sim_{u,v}$: The correlation of ratings expressed by u and v, and the size of set of common items rated by both u and v.

$$sim_{u,v} = corr_{u,v} \times \frac{1}{1 + e^{-|UC_{u,v}|/2}} \qquad (2)$$

Here, $corr_{u,v}$ is the Pearson correlation of ratings of users u and v. $UC_{u,v}$ is the set of common items rated by both u and v. Basically, we punish the similarity of users who have a few items rated in common by considering a sigmoid function of $UC_{u,v}$ as a factor in $sim_{u,v}$. It should be noted that we ignore users with negative correlations, since they are anti-correlated to the source user.

[4]www.amazon.com
[5]www.ebay.com

3.2 Item-based CF for Top-N Recommendation

The idea of item-base collaborative filtering for top-N recommendation has been used in [1][5]. But the rating matrix they consider is only a binary matrix. Basically users have a basket of items, instead of ratings on items. In this section, we describe the item-based approach for top-N recommendation in ratings system as another baseline method in this paper. In this approach, for each item rated by the user, we find the set of top K similar items to that item, N_i, and aggregate these items to compute the set of top-N recommended items. We use equation 4 to compute the similarities of items. Notice that since the source user u has already expressed ratings on items in I_u, we can not choose all items rated by the source user and find similar items to them. Instead, we just choose highly rated items. Basically we define $I'_u = \{i \in I_u | r_{u,i} = max_u\}$, where $max_u = max_{i \in I_u} r_{u,i}$. We compute a score in N_i for all items i rated by u as follows:

$$s_i = \sum_{j \in I'_u, i \in N_j} sim_{i,j} \qquad (3)$$

In the above equation, s_i is the score computed for each item i similar to one of the items in I'_u. The top N items with highest values of s_i will be returned as the top-N recommended items. $sim_{i,j}$ measures the similarity of items i and j. Similar to the user based CF, we compute $sim_{i,j}$ as follows:

$$sim_{i,j} = corr_{i,j} \times \frac{1}{1 + e^{-|UC_{i,j}|/2}} \qquad (4)$$

Here, $corri, j$ is the Pearson Correlation of ratings for items i and j. Also $UC_{i,j}$ is the set of users who have rated both i and j. Notice that in [1], they do not distinguish similar users and all similar items are being considered as the same. In other words, values of $sim_{i,j}$ are being considered as 1. But in this definition, we are able to put more weights on items more similar to items rated by the source user.

4. TRUST-BASED APPROACHES TO TOP-N RECOMMENDATION

In this section, we propose novel methods which exploit trust networks to improve the quality of top-N recommendation.

4.1 Random Walk Approach

Recently, in [4] we presented a random walk method to combine the trust-based and item-base approaches for prediction of ratings on single items. In this subsection, we extend this approach to recommend top-N items for a source user u.

Starting from user u, we perform a random walk on the trust network. Each random walk stops at a certain user. Then the items rated highly by that user will be considered as the recommended items, ordered according to the ratings expressed by that user. We perform several random walks to gather more information and compute a more confident result. The estimated rating of each item is the average of ratings for that item over all raters considered. At the end, we output items with the highest estimated rating as top-N recommended items. Similar to [4], we compute the variance of the estimated rating for items and continue performing further random walks until the variance converges.

Now, we discuss the details of a single random walk. We start each random walk from the source user u. At each node v, with probability $\phi_{u,v}$, we stop the random walk and return items rated by v. With probability $1 - \phi_{u,v}$, we continue the random walk to one of the neighbors of v. We select the neighbor of v uniformly from directly trusted neighbors of v. $\phi_{u,v}$ depends on the similarity of user v with the source user u. The more similar they are the more likely the random walk stops at v. Also, the further away from the source user we go in the network, the probability of stopping the random walk should get higher to avoid noisy data located far from the source user u. So we denote the probability of stopping at node v by $\phi_{u,v,k}$ where k is the current step of the random walk. Similar to [4], we define this probability as follows:

$$\phi_{u,v,k} = (0.5 + \frac{sim_{u,v}}{2}) \times \frac{1}{1 + e^{-k/2}} \tag{5}$$

In the above equation, $sim_{u,v}$ is computed using equation 2. Since the values of $sim_{u,v}$ are in the range [-1,1], we shift the values of similarity to get a value in the range [0,1] that can be interpreted as a probability. Basically, if v is not similar to u at all ($sim_{u,v}$ is a very small positive number), this offset allows the random walk to still have the possibility to stop at v. But if $sim_{u,v}$ is close to -1, the probability of stopping at v would be very close to zero, and the random walk will most likely continue to find another user in the network.

Notice that the factor $\frac{1}{1+e^{-k/2}}$ is a sigmoid function which injects the effect of the current step of random walk into the stopping probability. Using the sigmoid function to consider the step of random walk (k) was introduced in [4]. $\phi_{u,v,k}$ depends on the similarity of users u and v, while the stopping criteria in [4] depends on the similarity of items rated by v and the target item. This is due to the different tasks being performed, in particular there is no target item in top-N recommendation. The intuition behind the new stopping probability in this paper is that users with similar rating patterns are more likely to agree on their top-N items.

Alternatively, we can ignore the effect of similarity and just use the trust network in our random walk. In other words, we set $\phi_{u,v,k} = \frac{1}{1+e^{-k/2}}$. We evaluated both alternatives in our experiments. It should be noted that we also associate a parameter $maxDepth$ with the random walk approach which determines the maximum depth to which a random walk can be continued. Random walks are forced to terminate after $maxDepth$ steps.

4.2 Combined Approach

When a user u trusts another user v it does not necessarily mean that they rate the same items. Basically, when u trusts v, it means that if they both rate an item, it's more likely for them to rate this item in a similar way. So, if we use leave-one-out method and ask the trust-base approach to recommend top-N items, it may not be able to recommend the exact withheld item, although the recommended items are actually interesting for the source user. On the other hand, in the collaborative filtering approach, we consider users who have similar rating patterns. Two users who have already rated some common items, tend to rate more items in common. Hence, it's more likely for users with similar rating patterns to the source users to also rate the withheld item.

Our experiments confirm this effect, which causes collaborative filtering to slightly outperform the trust-based random walk. Therefore we propose a combined approach to benefit from properties of both trust-based and collaborative filtering approaches.

In this approach we compute the top K trusted users in the network and rank the items rated by these trusted users to compute top-N recommended items. We use the collaborative filtering approach to compute another set of top-N recommended items. Finally we merge these two lists to produce a combined lit of top-N recommended items.

This approach uses similar users which are more likely to have the withheld items, and it also uses a trust network to deal with cold start users. Notice that half of the users are cold start users, and collaborative filtering is not successful in finding similar users for these users.

To compute top K trusted users, we can use two different alternatives:

- Breadth First Search.
- Random Walk in the social network.

Breadth First Search (BFS). In this approach, we perform a BFS to find k_2 closest users to the source user u in the trust network. Then, we merge the items rated by these trusted users to find the top-N recommended items returned by trust-base approach (We denote these items as TR_u). We estimate the rating of items rated by trusted neighbors as follows:

$$\hat{r}_{tu,i} = \frac{\sum_{v \in Nt_u, i \in I_v} w_t(u,v) \times r_{v,i}}{\sum_{v \in Nt_u, i \in I_v} w_t(u,v)} \tag{6}$$

In the above equation, $\hat{r}_{tu,i}$ denotes the estimated rating using the trusted neighborhood. Also Nt_u denotes the top k_2 trusted users found by BFS. $w_t(u,v)$ is the influence coefficient of each user in the trusted neighborhood. We define $w_t(u,v) = 1/d_v$, where d_v is the depth at which we found v in the BFS starting from u.

Random Walk. In the random walk approach, we perform random walks similar to the ones introduced in the previous subsection. Each random walk stops at a certain user. This user will be considered as a trusted user. We continue performing random walks until we get k_2 users. The rest is the same as BFS approach, with $w_t(u,v)$ equal to the number of times user v has been returned as the result of a random walk.

After finding top k_2 trusted users (with either approach), we combine the results of the trust-based approach and the CF based approach. Suppose that in the CF based approach we use k_1 top similar users and merge their items. We denote the items returned by CF approach as CF_u. Now we define the merge method as follows:

$$\hat{r}_{u,i} = \begin{cases} \frac{\hat{r}_{cu,i} + \hat{r}_{tu,i}}{2} & i \in TR_u \,;\, i \in CF_u \\ \hat{r}_{tu,i} & i \in TR_u \,;\, i \notin CF_u \\ \hat{r}_{cu,i} & i \in CF_u \,;\, i \notin TR_u \end{cases} \tag{7}$$

The top-N items with highest values of $\hat{r}_{u,i}$ will be returned as the top-N recommended items. It should be noted that we could do a weighted averaging instead of just taking the means for items which appear in both TR_u and CF_u. To consider the weights, we define $\hat{r}_{u,i}$ as follows:

$$\hat{r}_{u,i} = \frac{k_1 \times \hat{r}_{cu,i} + k_2 \times \hat{r}_{tu,i}}{k_1 + k_2}$$

We discuss the effect of weighted merging of results of CF approach and trust-based approach in section 5.

5. EXPERIMENTS

This section reports our experimental results on a real life data set comparing existing CF-based methods for top-N recommendation against our methods for using trust network in top-N recommendation.

5.1 Data Set Specifications

Most data sets for recommendation have no social network among users. To the best of our knowledge, the Epinions[6] data set is the only data set publicly available which has both a trust network and ratings expressed by users.

We used the version of the Epinions data set[7] published by the authors of [10]. This data set is very sparse. The data set contains 49k users with at least one rating, out of which 24k users are cold start users. We consider users with less than 5 ratings as cold start users (similar to [8],[4]). We have 104k items with 575k ratings expressed for them, and 508k trust statements among pairs of users. As mentioned above, 49% of users are cold start users which is a very large portion. So, considering the performance of the recommendation for cold start users is very important. The distribution of the number of ratings per users follows a power law. Notice that there exists another version of the Epinions data set[8] prepared and published by Paolo Massa[8]. We also ran experiments on this data set with very similar results. The data set we used in our experiments has additional features (such as categories of items, timestamps) which may be helpful for future works.

5.2 Experimental Setups

In our experiments we set $N = 100$ for the following reasons. Our data set contains significantly more items than users which is different from many other data sets. So using a small value for N will produce generally poor results for all compare methods. Actually, we have performed experiments for $N = 10$, and the recall for collaborative filtering on cold start users was 1.87% while the recall for the random walk based approach was 2.12%. See section 5.3 for a precise definition of recall.

As discussed in previous sections, collaborative filtering approaches use a neighborhood size of k and aggregate the items rated by these neighbors to recommend top-N items. Therefore, we report the results of user-based collaborative filtering for different values of k.

As discussed earlier, we analyze two instances of the random walk approach: In one of them we consider the similarity of users, but in the other one, we ignore the similarities and use $\phi_{u,v,k} = \frac{1}{1+e^{-k/2}}$.

In the combined approach, we merge the result of collaborative filtering and trust based approach. The number of neighbors considered in CF approaches is k_1, and the number of trusted neighbors being considered in trust-base approach is k_2. We run different instances of the combined approach:

1. In this setting we set $k_1 = k_2$ and run the combined approach to merge user based CF and trust based approach. We run experiments for different values of $k_1 = k_2$.
2. Here, we fix k_1 and set it to the value of k_1 which has the best result in user based CF. Then we run experiments for different values of k_2. Finally we merge the results of these two approached as shown in equation 7.
3. This setting is similar to the second setting, but merging the results of collaborative filtering approach and trust based approach is different. We take the weighted mean of the results of CF and trust-based approach. The weights are proportional to k_1 and k_2 as discussed earlier in the paper.
4. This instance performs random walks for computing the trusted neighborhood. Note that the previous two instance use BFS to compute the trusted neighborhood. In merging the results of collaborative filtering approach and trust based approach, we do not consider weights in this setting.

We implemented all the methods in Java. We used an Intel Core2 Duo 2.2 GHz CPU with 2GB RAM to run our experiments on a Windows XP system.

In our experiments, we compare the results for different methods. Following is the description of labels we use to denote each of these algorithms:

- *CF-User*: User based Collaborative Filtering Approach.
- *CF-Item*: Item based Collaborative Filtering Approach.
- *TrustWalkerList D2*: This is the Random Walk approach with *maxDepth=2*.
- *TrustWalkerList D2-pure*: The random Walk approach ignoring similarities.
- *TrustWalkerList D3*: This is the Random Walk approach with *maxDepth=3*.
- *Trust-CF (k1=k2)* The combined approach with $k_1 = k_2$.
- *Trust-CF (k1=70)*. The instance of combined approach where a fixed value of k_1 is used. The top k_2 trusted users are computed using BFS. The results for different values of k_2 are shown in figures 1 to 6.
- *Trust-CF-RW* is the same as *Trust-CF (k1=70)*, but the top k_2 trusted users are computed using a random walk approach.
- *Trust-CF Weighted*. Combined approach with a fixed k_1 and weighted merge of results of CF and trust-based approach. The trusted neighborhood is computed using BFS. The results for different values of k_2 are shown in the figures.

5.3 Evaluation Metrics

Typically, the leave-one-out method is used to evaluate recommendation systems [2][8][11]. In the leave-one-out method for predicting a single rating, we withhold a rating and try to predict it using the trust network and the remaining ratings. For top-N recommendation, we have to adjust the leave-one-out method. We withhold a user's rating on an item, and ask the recommender to recommend top-N items for this user. If the withheld item is among the N recommended items, then we say that a hit has occurred. We compute the recall as follows:

$$recall = \frac{Number of hits}{L} \tag{8}$$

where L is the number of pairs of <user,item> being considered in the leave-one-out method. It should be noted that in the leave-one-out method we do not withhold all items of a user. Only items which have been highly rated will be withheld. In our experiments, we computed the maximum rating value expressed by each user and then we considered only items rated with the value of that maximum rating as items being withheld. It should be noted that $recall$ has been also called *hit-ratio* in related works [1][5][6].

[6]www.epinions.com
[7]http://alchemy.cs.washington.edu/data/epinions/
[8]http://www.trustlet.org/wiki/Epinions_dataset

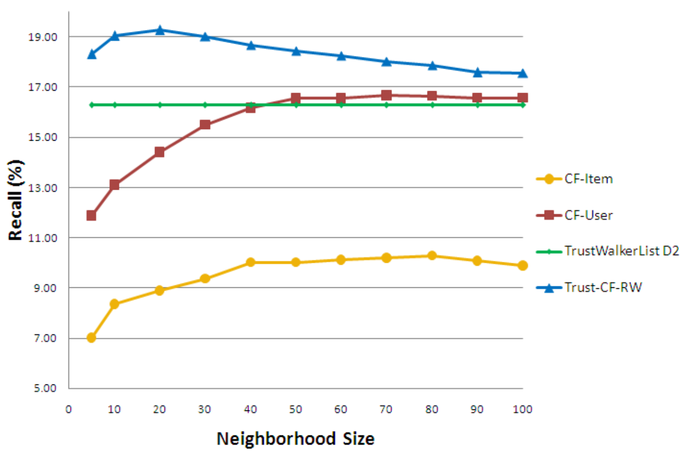

Figure 1: Results for ALL Users. The Y axis shows the recall of different methods in percent. The X Axis shows different neighborhood sizes. Only the best instances of the combined approaches and the random walk approaches are shown in this figure

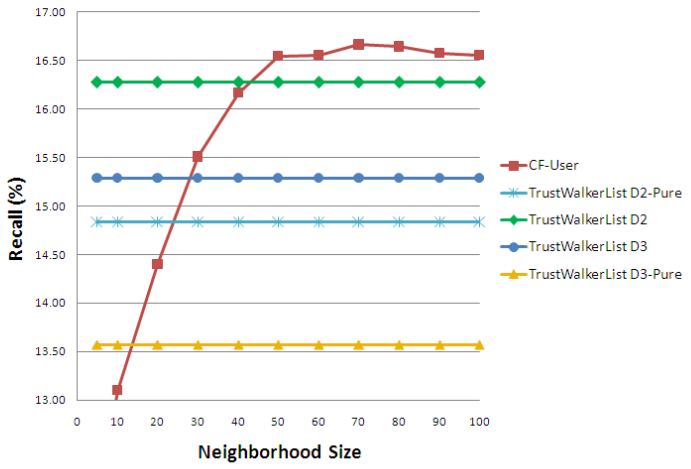

Figure 2: Results of different Random Walk based approaches for ALL Users compared to the results of User-based Collaborative Filtering. The Y axis shows the recall of different methods in percentage. The X axis shows different neighborhood sizes.

5.4 Experimental Results

In this subsection, we present the experimental results of different methods introduced in this paper and their comparison with existing methods.

We perform our experiments in two general ways. The first one is performing the experiment on all users. Since the main purpose of using a trust network is to improve the quality of recommendation for cold start users, we also perform the experiments for only cold start users.

Figures 1,2, and 3 present the recall for different methods on all users for varying neighborhood sizes. Note that neighborhood size is only a parameter for CF-based and combined approaches, so pure random walk based approaches produce constant curves. These figures show that the optimum k for user-based collaborative filtering is 70 which leads to a recall of 16.66%. The Item-based Collaborative filtering achieves poor results compared to the User-based CF, as shown in figure 1. The best recall for Item-based CF is 10.26% which is very low compared to 16.66% of User-based CF.

User-based CF for top-N recommendation relies on similar users which have similar rating patterns. These users are more likely to rate the withheld item. But item-based CF relies on items similar to items rated by the user. The withheld item is not necessarily among items similar to items rated by the user unless this withheld item is actually similar to other items rated by the user (which is not always true). So, the results of item-based CF for top-N recommendation are generally poor compared to user-based CF.

As shown in figure 2, among different instances of the random walk based approach the one with maximum depth of 2 achieves the best results. The figure also shows that taking the similarity of users into account improves the results. The random walk approach which exploits the similarity of users, *TrustWalkerList D2*, has comparable results to *CF-User*, but *CF-User* outperforms it in its best performance.

Intuitively, *TrustWalkerList* should perform better than *CF-User* since it is using extra information embedded in the trust network, but the results do not show the better performance. We believe that this is due to the leave-one-out method used for evaluation. When a user u trusts another user v it does not necessarily mean that they rate the same items. It means that if they both rate an item, it is more likely for them to rate this item in a similar way. But using leave-one-out and recall as evaluation criteria, we are explicitly looking for the withheld items which may have not been rated by trusted users. Manually checking the results revealed that the recommended items are actually related to the interests of a user, but leave-one-out and recall are unable to capture this. On the other hand, in *CF-User* we select users who have rated items in a similar way to the source user and aggregate their ratings. If two users rate similar items (they have similar rating patterns), then it is more likely that a similar user also rates the withheld item. Hence, the recall of *CF-User* is better than that of *TrustWalkerList*.

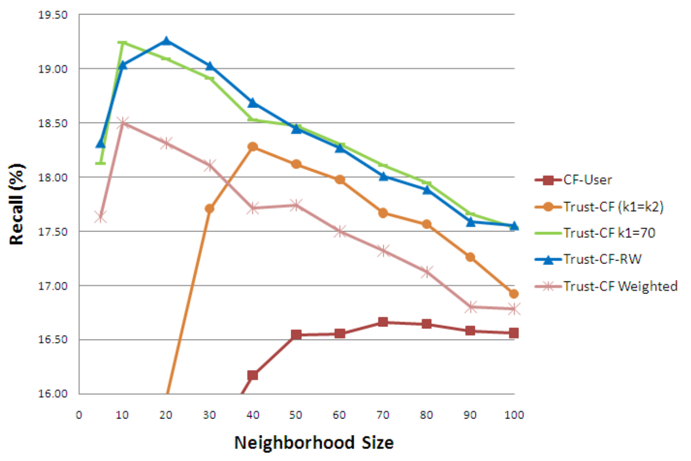

Figure 3: Results of different instances of combined approach for ALL Users compared to the results of User-based Collaborative Filtering. The Y axis shows the recall of different methods in percentage. The X axis shows different neighborhood sizes.

Figure 3 compares the results of different instances of the combined approach. It shows that all of them outperform *CF-User*. *CF-User* achieves its best results for k=70, so we consider $k1 = 70$ for all combined approaches. Among the combined approaches, *Trust-CF (k1=70)* and *Trust-CF-RW* outperform the other methods, but *Trust-CF-RW* slightly outperforms *Trust-CF (k1=70)*. The best result of *Trust-CF-RW* is achieved for $k_2 = 20$. The best recall for *Trust-CF-RW* is 19.26% as opposed to 16.66% for *CF-User*, which leads to an improvement of 15.5% over *CF-User*. Also, figure 3 shows that weighted merging of results of collaborative filtering and trust-based approach (*Trust-CF-Weighted*) does not improve the results.

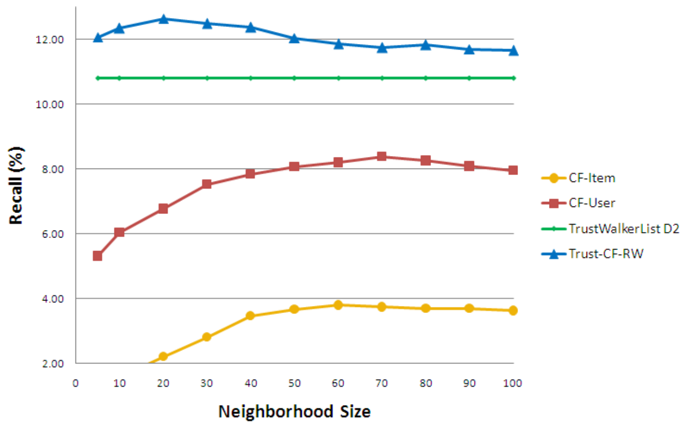

Figure 4: Results for COLD START Users. The Y axis shows the recall of different methods in percentage. The X axis shows different neighborhood sizes. Only the best instances of combined approaches and random walk based approaches are shown in this figure.

Figures 4, 5 and 6 show the results for cold start users. The recall for cold start users is generally lower than those for all users since there are only few ratings available for cold start users. Notice that among different instances of Random Walk based and combined approach, we only show the instances with the best results to make the figure clearer. The result for different instances of random walk based and combined approaches are shown in figures 5 and 6.

As shown in figures 4, 5, and 6, all approaches exploiting a trust network outperform collaborative filtering approaches. This confirms our claim that using a trust network improves the quality of recommendation for cold start users. The improvement of the quality of recommendation in terms of recall gained by exploiting trust is much higher for cold start users compared to all users. *Trust-CF-RW* achieves a recall of 12.65% ($k_1 = 70$ and $k_2 = 20$) as compared to 8.38% for *CF-User* ($k = 70$). *Trust-CF-RW* improves *CF-User* by 50.1%, while this improvement for all users is 15.5%. This great improvement for cold start users is mainly because only very few ratings are expressed by cold start users which makes it hard for collaborative filtering approaches to find similar users.

In this paper, we weight different similar or trusted users by their correlation with the source user ($w_{u,v}$). Most works in the literature [1][5][6] do not distinguish them and consider $w_{u,v} = 1$ for all similar users. Figure 7 displays the effect of considering the similarity and trustworthiness of users in the neighborhood on the results of recommendation. It clearly shows that distinguishing different users in the neighborhood improves the recall for both collaborative filtering and trust-based approaches.

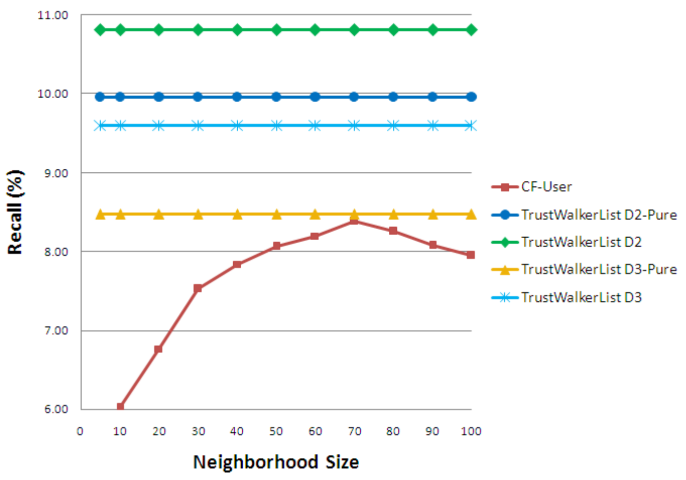

Figure 5: Results of different Random Walk based approaches for COLD START Users compared to the results of User-based Collaborative Filtering. The Y axis shows the recall of different methods in percentage. The X axis shows different neighborhood sizes.

6. CONCLUSION

Recommender systems are emerging as tools of choice to select the online information relevant to a given user. Collaborative filtering is the most popular approach to building recommender systems and has been successfully employed in many applications. However, it cannot make accurate recommendations for so-called cold start users that have rated only a very small number of items. Trust-based recommender systems can better deal with cold start users, since users only need to be simply connected to the trust network.

In this paper, we addressed the top-N item recommendation problem which is in many scenarios a more natural problem compared to the well know problem of predicting a user's rating on a given target item. We proposed different methods which exploit the trust network to improve the quality of top-N recommendation. The first approach is a random walk on the trust network which also considers the similarity of users in the network. The second approach combines the collaborative filtering and trust-based approaches. We performed an evaluation on the Epinions data set using recall as evaluation metrics. Our experiments demonstrate that approaches using a trust network outperform collaborative filtering approaches which only use the rating data. In particular for cold start users, we achieved a 50% improvement over the collaborative filtering approach.

This work suggests several interesting directions for future work. The evaluation metric used in this paper has been used in the literature, but seems not to be the best fit for trust based top-N recommendation. Proper evaluation procedures should be further investigated. The diversity of items recommended is also important. Users like to get recommendations from various categories, which should be considered in the evaluation of recommendation methods. In this paper and in the related work, the ratings are assumed to be stored in a centralized repository. However, applications such as mobile social networks require a distributed recommender, and a random walk model is a promising approach for such scenarios.

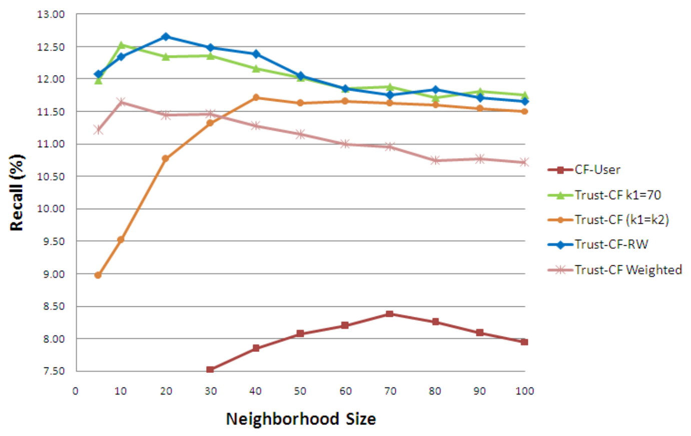

Figure 6: Results of different instances of combined approach for COLD START Users compared to the results of User-based Collaborative Filtering. The Y axis shows the recall of different methods in percentage. The X Axis shows different neighborhood sizes.

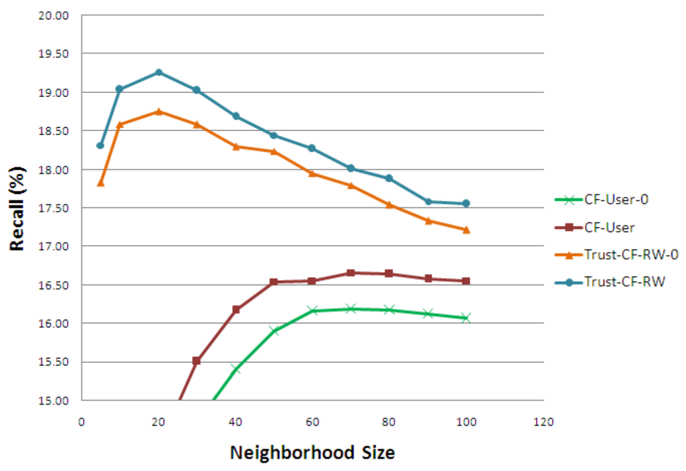

Figure 7: Effect of distinguishing users in the neighborhood by their similarity and trust values. *CF-User-0*, and *Trust-CF-RW-0* respectively denote the versions of *CF-User* and *Trust-CF-RW* where we do not distinguish users in the neighborhood.

7. REFERENCES

[1] M. Deshpande and G. Karypis. Item based top-n recommendation algorithms. *ACM Transactions on Information Systems*, 22:143–177, 2004.

[2] J. Golbeck. *Computing and Applying Trust in Web-based Social Networks*. PhD thesis, University of Maryland College Park, 2005.

[3] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12), 1992.

[4] M. Jamali and M. Ester. Trustwalker: A random walk model for combining trust-based and item-based recommendation. In *KDD'09: The 15th ACM SIGKDD conference on Knowledge Discovery andData Mining*, 2009.

[5] G. Karypis. Evaluation of item-based top-n recommendation algorithms. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 247–254, New York, NY, USA, 2001.

[6] H.-N. Kim, A.-T. Ji, H.-J. Kim, and G.-S. Jo. Error-based collaborative filtering algorithm for top-n recommendation. In *The Joint International Conferences on Asia-Pacific Web Conference and Web-Age Information Management (APWeb/WAIM)*, pages 594–605, Huang Shan, China, June 2007.

[7] Y. Kwon. Improving top-n recommendation techniques using rating variance. In *RecSys'08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 307–310, New York, NY, USA, 2008.

[8] P. Massa and P. Avesani. Trust-aware recommender systems. In *RecSys'07: ACM Recommender Systems Conference*, USA, 2007.

[9] M. R. McLaughlin and J. L. Herlocker. A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *SIGIR '04: Proceedings of the 27th international ACM SIGIR conference on Information Retrieval*, pages 329–336, New York, NY, USA, 2004.

[10] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD'02: The 8th ACM SIGKDD conference on Knowledge Discovery andData Mining*, 2002.

[11] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW'01: 10th International World Wide Web Conference*, 2001.