AN EFFECTIVE COLD START RECOMMENDATION METHOD USING A WEB OF TRUST

Yu-Hao Wan, Department of Information Management, National Taiwan University, Taipei, Taiwan, R.O.C., r98725027@ntu.edu.tw

Chien Chin Chen, Department of Information Management, National Taiwan University, Taipei, Taiwan, R.O.C., paton@im.ntu.edu.tw

Abstract

Cold start recommendations are important because they help build user loyalty, which is the key to the success of e-services and e-commerce systems. Recommending useful information for new users generally creates a sense of belonging and loyalty, and encourages them to visit e-commerce systems frequently. However, as new users take time to become familiar with recommendation systems, the systems usually have limited information about newcomers and have difficulty providing appropriate recommendations. The cold start phenomenon has a serious impact on the performance of recommendation systems. To address the problem, we propose a cold start recommendation method that integrates a web of trust with a user model to identify trustworthy users. The suggestions of those users are then aggregated to provide useful recommendations for cold start users. Experiments based on the well-known Epinions dataset demonstrate that the proposed method is effective and efficient, and outperforms well-known recommendation methods by a significant margin.

Keywords: Recommendation Systems, Collaborative Filtering, Trust Network.
1 INTRODUCTION

With the rapid development of the World Wide Web, many online publishing tools (e.g., weblogs and online forums) enable users to share and obtain knowledge. However, although the Web is an abundant source of information, the enormous number of web documents often makes searching for information a difficult task. Even with the support of search engines, the number of returned documents is too large for users to obtain desired information (Das et al. 2007). To help users acquire desired information efficiently, a great deal of research has been devoted to improving recommendation systems. Given the profile of a user, which usually contains the user’s historical item ratings, a recommendation system suggests items related to his/her interests. An item can be a picture, a video clip, a book, or a product review, depending on the domain of the recommendation system. One of the most popular recommendation system approaches is collaborative filtering, which is based on the concept that like-minded people prefer similar items. Collaborative filtering thus uses the information in a user’s profile to identify reference users whose tastes are similar to those of the target user. It then makes recommendations to the target user based on the items that interest the reference users. More specifically, collaborative filtering recommends items by aggregating the historical item preferences (i.e., item ratings) of reference users stored in a database. Because collaborative filtering is effective, it is has been adopted by many e-commerce websites, such as Amazon.com and Epinions.com.

Luarn and Lin (2003) posited that user loyalty is the key to the success of e-services and e-commerce systems. In the case of new users, providing them with useful information usually creates a sense of belonging and loyalty, and encourages them to visit e-commerce systems frequently. Making appropriate item recommendations to new users are thus essential. Resnick and Varian (1997) observed that making recommendations to new users is a difficult task because of the cold start phenomenon. As new users take time to become familiar with recommendation systems, the systems usually have limited profiles of such users. Consequently, it is difficult for collaborative filtering techniques to identify effective reference users and make useful recommendations to new users. Since then, a number of approaches have been proposed to provide useful recommendations for new users (e.g., Park et al. 2006; Ahn 2008). Most e-commerce systems allow users to establish social relationships. For instance, at Epinions.com, users can create trust lists and block lists. The trust lists form a web of trust (WOT) that indicates the credibility of the listed users. Intuitively, people believe others on their trust lists (or users trusted by their trusted users) and accept that items recommended by them may be useful. Sinha and Swearingen (2001) suggested using the web of trust concept for collaborative filtering, and proposed a trust-based collaborative filtering method that treats trusted users as reference users to derive appropriate recommendations. More recently, Victor et al. (2008) utilized the web of trust concept to solve the cold start recommendation problem, but they found that the web of trust is also affected by the cold start problem. We investigated the utility of the web of trust on the Epinions dataset (Avesani and Massa 2007), and found that the average number of trusted users for new users is 0.68, which is far lower than the number for experienced user1. Since new users do not know which users are trustworthy, limited (i.e., short) trust lists would degrade the quality of cold start recommendations.

In this paper, we propose a two-stage method to solve the cold start recommendation problem. In the first stage, we construct a user model by employing the K-means clustering algorithm (Manning et al. 2008) to group users into clusters. The users in each cluster have similar item preferences. Rather than using the limited trust lists discussed above, the proposed method suggests trustworthy users to cold start users. We construct a web of trust for each cluster and utilize the PageRank algorithm (Page et al. 1998) to identify experts in a cluster. In the second stage, the identified experts are exploited as reference users to make useful recommendations for cold start users. We also propose a technique to identify the implicit links among users in a web of trust. Although recommendation systems provide user-friendly interfaces for compiling trust lists, many users may be unwilling to use the function due

1 The average number of trusted users for experienced users is 12.2.
to privacy concerns. The proposed technique resolves the problem by analyzing the rating behavior patterns of users and identifying instances of implicit trust to enrich the web of trust. The evaluations based on the Epinions dataset demonstrate that the proposed method outperforms well-known collaborative filtering methods in terms of the coverage rate and execution time, without a significant reduction in the precision of the recommendations.

The remainder of this paper is organized as follows. The next section contains a review of related works on collaborative filtering and trust-based collaborative filtering approaches. We introduce the proposed two-stage recommendation method in Section 3, and evaluate it in Section 4. Then, in Section 5, we provide some concluding remarks and consider future research avenues.

2 RELATED WORK

In this section, we consider a number of collaborative filtering and trust-based recommendation methods.

2.1 Collaborative Filtering

There are three types of collaborative filtering approaches, namely, memory-based, model-based, and hybrid approaches. The memory-based approach is the most widely used collaborative filtering technique (Resnick et al. 1994; Breese and Kadie 1998; Nicholas and Soboroff 2000). Given the historical item ratings of a user $u_n$ (i.e., the user profile of $u_n$) the memory-based approach predicts the preference (i.e., item rating) of an un-rated item $i_m$ for $u_n$ by the following equation:

$$
\hat{u}_{n,m} = \bar{u}_n + \sum_{j \in U_n} w_{n,j} \left( u_{j,m} - \bar{u}_j \right),
$$

(1)

where $\hat{u}_{n,m}$ denotes the predicted item rating; $\bar{u}_n$ is $u_n$’s average item rating; $U_n$ is a set of reference users whose preferences are similar to that of $u_n$; $w_{n,j}$ indicates the similarity between users $u_n$ and $u_j$; and $u_{j,m}$ is the rating of $i_m$ given by user $u_j$. Specifically, the memory-based approach predicts $\hat{u}_{n,m}$ by averaging the item ratings of the reference users. However, the approach needs to record the ratings of all users and a database scan is required to extract similar users as the reference users. Consequently, the computation cost of the approach is expensive. The correlation coefficient (Resnick et al. 1994) and the vector space similarity (Breese and Kadie 1998) are two popular methods used to determine the similarity of users’ ratings for items. Breese and Kadie observed that the correlation coefficient method generally outperforms the vector space method.

The model-based approach constructs a model of users’ preferences, instead of recording all user ratings. A number of works (Ungar and Foster 1998; Kohrs and Merialdo 1999; Breese and Kadie 1998) employ clustering algorithms to group users with similar item ratings. After creating the clusters, recommendations can be made by averaging the item ratings in the cluster that the target user belongs to. The approach is efficient because a database scan is not required. However, as clustering abstracts detailed user preferences, the recommendation precision is usually inferior to that of the memory-based approach. Puzia and Hofmann (1999) proposed using a probabilistic aspect model, which represents user preferences as a convex combination of latent class variables. A class variable represents a user cluster and is associated with pairs of users and items. The aspect model assumes that users and items are independent given the latent class variables.

Pennock et al. (2000) proposed a hybrid method that integrates the memory-based approach with the model-based approach. Given the ratings of users, the method computes the probability that a user belongs to a certain cluster. The rating distribution of the cluster then assigns the missing ratings of the target user to derive appropriate memory-based recommendations. Subsequently, Xue et al. (2005) proposed a hybrid method that partitions users into clusters and uses the rating distributions of the clusters to smooth the users’ ratings. The smoothed ratings enrich user profiles and improve the
performance of collaborative filtering. Empirical studies have shown that the method outperforms many well-known collaborative filtering methods.

One problem with collaborative filtering is data sparsity. Generally, e-commerce systems contain a lot of items but users have limited interests. Consequently, most people only rate a small number of items and many items are never rated or purchased. In the model-based approach, the lack of user ratings biases user clustering; while in the memory-based approach, the sparsity of item ratings hampers the identification of reference users. Thus, the data sparsity problem has a serious impact on the recommendation performance. The problem is even worse for new users because they need time to become familiar with e-commerce systems. The cold start phenomenon leads to short user profiles and may produce useless recommendations. To ensure that new (cold start) users revisit e-commerce systems, it is crucial to resolve the cold start phenomenon so that useful recommendations can be made for such users (Avesani and Massa 2007).

2.2 Trust-based Recommendation System

Because of the prevalence of social computing, many recommendation research works have started using social networks. Since most recommendation systems allow users to compile trust lists, a web of trust can be constructed by aggregating the lists. Normally, users believe their trusted users (i.e., the users on their trust lists), so the recommendations made by those users can be exploited (Sinha and Swearingen 2001). Golbeck (2006) proposed the TidalTrust system, which derives item ratings from trust lists by the following equation:

$$\hat{u}_{nm} = \frac{\sum_{u_j \in U_n^+} t_{n,j} U_{jm}}{\sum_{u_j \in U_n^+} t_{n,j}},$$

where $U_n^+$ represents a set of users trusted by user $u_n$; and $t_{n,j}$ is the weight of the trust between users $u_n$ and $u_j$, which is derived by an iterative network propagation algorithm. Specifically, the method predicts an item’s rating by averaging the ratings made by the trusted users. Golbeck showed that the method outperforms a baseline approach that averages the ratings of all users to make recommendations. O’Donovan and Smyth (2005) proposed the following trust-based collaborative filtering method:

$$\hat{u}_{nm} = \bar{u}_n + \frac{\sum_{u_j \in U_n^+} w_{n,j} (u_{jm} - \bar{u}_j)}{\sum_{u_j \in U_n^+} w_{n,j}},$$

where $U_n^+ = U_n^+ \cap U_n^+$. This method modifies the original collaborative filtering approach (i.e., Equation 1) by integrating trust lists. The reference users in $U_n^+$ are the similar users trusted by user $u_n$. The MoleTrust system (Avesani and Massa 2004) also modified Equation 1 for trust-based collaborative filtering as follows:

$$\hat{u}_{nm} = \bar{u}_n + \frac{\sum_{u_j \in U_n^+} t_{n,j} (u_{jm} - \bar{u}_j)}{\sum_{u_j \in U_n^+} t_{n,j}},$$

Instead of exploiting the similarities of users’ interests, MoleTrust weights the ratings of trusted users with $t_{n,j}$. As the performance of trust-based collaborative filtering depends on the quality of the web of trust, users are encouraged to connect with each other in a trust network. Trust-based collaborative filtering is generally superior to collaborative filtering; however, it also suffers from the cold start problem because new users obviously cannot compile informative trust lists (Victor et al. 2008). In this paper, we propose a novel cold start recommendation method that integrates a web of trust with the model-based approach to suggest trustworthy users to cold start users. By aggregating the preferences of the suggested experts, effective recommendations can be provided for cold start users.
3 METHODOLOGY

3.1 Data Definition and System Structure

First, we introduce the notations used in the remainder of the paper. Let \( I = \{i_1, i_2, \ldots, i_M\} \) be a set of items in a recommendation system, and let \( U = \{u_1, u_2, \ldots, u_N\} \) be a set of system users. We represent a user as an \( M \)-dimensional vector \( u_n \), where an entry \( u_{n,m} \) indicates the rating of \( i_m \) given by \( u_n \); and \( u_{n,m} \) is a non-negative number that equals zero if \( u_n \) has not rated \( i_m \). Otherwise, the value indicates the degree of preference (ranking) for the item. Let \( \bar{u}_n \) denote the average item rating of \( u_n \), and let \( |u_n| \) indicate the number of ratings made by \( u_n \). A user is considered a cold start user if \( |u_n| \) is smaller than a pre-defined threshold \( \alpha \). Our goal is to predict the possible rating \( \bar{u}_{n,m} \) of an un-rated item \( i_m \) for a cold start user \( u_n \).

- First Stage: Model Construction

- Second Stage: Recommendation

![Figure 1. The System Structure](image)

Figure 1 shows the structure of our cold start recommendation method, which integrates a web of trust with the model-based approach in two stages: the model construction stage and the recommendation stage. In the first stage, we use the set of experienced users (i.e., non-cold-start users) to construct a user model. We then partition the users into different clusters so that users in the same cluster have similar item preferences. To alleviate the cold start problem, we construct a web of trust for each cluster and employ the PageRank algorithm to identify the trustworthy experts in each cluster. The experts then become reference users who produce useful recommendations for cold start users. We
also propose an implicit link identification technique to discover the implicit links in a web of trust and improve the selection of experts. In the recommendation stage, we compute the possible rating of an un-rated item for a cold start user. After identifying the cluster that the item belongs to, we aggregate the ratings of the experts in the cluster to make recommendations. We describe each system component in detail in the following sub-sections.

3.2 User Model Construction

As the proposed method is model-based, we construct a user model by clustering experienced users. Let \( C = \{c_1, c_2, ..., c_K\} \) denotes a set of user clusters. A cluster \( c_k \) comprises users with similar item ratings. We employ the K-means algorithm for user clustering because it is effective and efficient (Manning et al. 2008). In the first step, the algorithm randomly selects \( K \) users as the initial cluster centroids. A cluster centroid, denoted as \( \bar{c}_k \), is an \( M \)-dimensional vector in which an entry \( c_{k,m} \) indicates the average rating of \( i_m \) over all users in the cluster. Then, the algorithm executes the following assignment and computation operations iteratively until the clustering result is stable. The assignment operation assigns each user to the cluster with the maximum similarity. We adopt the following correlation coefficient method to calculate the similarity between users and clusters:

\[
sim(u_n, c_k) = \frac{\sum_{u_{n,m} > 0, c_{k,m} > 0} (u_{n,m} - \overline{u}_n) \times (c_{k,m} - \bar{c}_k)}{\sqrt{\sum_{u_{n,m} > 0, c_{k,m} > 0} (u_{n,m} - \overline{u}_n)^2 \times \sum_{u_{n,m} > 0, c_{k,m} > 0} (c_{k,m} - \bar{c}_k)^2}},
\]

where \( \bar{c}_k \) is the average item rating of \( c_k \). Specifically, users with similar rating preferences are grouped together. The computation operation updates a centroid by the following equation:

\[
c_k = \frac{1}{|c_k|} \sum_{u_n \in c_k} u_n,
\]

where \( |c_k| \) denotes the number of users in \( c_k \). When the clustering result is stable, \( C \) partitions users into \( K \) clusters. We then identify experts in the clusters to derive cold start recommendations.

3.3 Identifying Cluster Experts

After users have been grouped into clusters based on their preferences, the recommendations provided by cluster experts should be useful for cold start users. To identify experts (i.e., trustworthy users) in a cluster, we measure each user’s trust score. Most e-commerce systems now allow users to establish social relationships. For instance, at Epinions.com, users can create trust lists and block lists. A user can add other users to her trust list if she thinks their preferences or recommended items are useful. By connecting the trust relationships among users, we can construct a web of trust for each cluster. Specifically, the web of trust of cluster \( c_k \) is a directed graph \( G_k = (c_k, e_k) \), where the users in \( c_k \) form a set of web nodes and \( e = \{(u_i, u_j)\} \) is a set of directed edges. A directed edge \((u_i, u_j)\) specifies that user \( u_j \) is trusted by user \( u_i \). Let \( r_k \) be a \(|c_k|\)-dimensional trust vector in which an entry \( r_{k,j} \) indicates the trust score of user \( u_j \) in \( c_k \). To derive the trust score, we employ the well-known PageRank algorithm, which is used to identify and rank informative web pages. The rationale behind PageRank is that a page is deemed informative if it is pointed to (i.e., linked) by a large number of informative pages. Similarly, in a web of trust, users trusted by a large number of trusted users would be regarded as cluster experts. The concept can be expressed by the following equation:

\[
r_{i,j} = \beta \frac{1}{|c_k|} + (1 - \beta) \sum_{(u_m, u_j) \in e_k} \frac{r_{k,j}}{\deg(u_i)},
\]
where $\text{deg}(u_i)$ denotes the out degree of $u_i$ in $G_k$; and parameter $\beta$ is a damping factor set at 0.15, as suggested by Page et al. Equation (7) is a recursive function as the trust score of a user depends on the trust scores of trusted users. We use the iterative algorithm shown in Figure 2 to calculate the trust score. In the first step, we randomly initialize a trust vector $r_k$. Then, we update the trust scores of users iteratively by using Equation 7. Erkan and Radev (2004) showed that $r_k$ will converge to a unique stationary distribution regardless the choice of initialized vector.

\begin{verbatim}
randomly initialize $r_k$
repeat
    $r_{old} = r_k$
    update $r_{k,j}$ by using Equation 7
    $\delta = ||r_k - r_{old}||$
until $\delta == 0$
return $r_k$
\end{verbatim}

Figure 2. The Trust Score Algorithm

After $r_k$ has converged, we rank cluster users according to their trust scores, and select the top-ranked users as experts to make recommendations for cold start users.

3.4 Implicit Link Identification

When analyzing the web of trust of the Epinions dataset, we observed that many trust lists are very short. This is because users often have privacy concerns and they are unwilling to compile comprehensive trust lists. The web of trust is therefore too sparse to derive informative experts. However, even if users avoid compiling trust lists, their activities reveal their trust orientations. We propose the algorithm shown in Figure 3 to identify the missing links in a web of trust.

\begin{verbatim}
for each user $u_i$ in $c_k$
    for each user $u_j$ in $c_k$ and $(u_i,u_j) \in e_k$
        if $f_{i,j} \geq \bar{f}_{t_i} - \delta_{f_{t_i}}$ and $\bar{\bar{t}}_{i,j} \geq \bar{\bar{t}_{i}} - \delta_{t_{i}}$
            add $(u_i,u_j)$ to $e_k$
        end if
    end for
end for
\end{verbatim}

Figure 3. The Implicit Link Identification Algorithm

In the algorithm, $f_{i,j}$ denotes the frequency of ratings given by $u_i$ to the items recommended by $u_j$; $\bar{f}_{t_i}$ denotes the average rating frequency of $u_i$ over all his/her trusted users, and $\delta_{f_{t_i}}$ is the corresponding standard deviation; $\bar{\bar{t}}_{i,j}$ indicates the average rating given by $u_i$ to the items recommended by $u_j$; $\bar{\bar{t}_{i}}$ denotes the average rating given by $u_i$ to the items recommended by his/her trusted users, and $\delta_{t_{i}}$ is the corresponding stand deviation. Specifically, we construct an implicit link from $u_i$ to $u_j$ if $u_i$ frequently gives a high rating to the items recommended by $u_j$. In the experiment section, we compare the recommendation performance with and without implicit link identification.

3.5 Item Recommendation

In the recommendation stage, we predict the possible rating $\hat{u}_{n,m}$ of an un-rated item $i_m$ for a cold start user $u_n$. In the model construction stage, users were grouped into clusters. As different clusters represent diverse item preferences, the first step of the recommendation stage involves selecting an appropriate cluster for $i_m$ recommendation. For each cluster, we record the number of times that an
item has been rated by the users in the cluster. Then, the cluster with the highest rating frequency of $i_m$ is selected to make the prediction. Let $c_k$ denote the selected cluster and let $\text{exp}_k$ represent the set of experts in the cluster. We compute $\hat{u}_{n,m}$ by the following equation:

$$\hat{u}_{n,m} = \bar{u}_n + \frac{\sum_{u_j \in \text{exp}_k} r_{k,j}(u_{j,m} - \bar{u}_j)}{\sum_{u_j \in \text{exp}_k} r_{k,j}},$$

(8)

In contrast to the MoleTrust system (i.e., Equation 4), we compute $\hat{u}_{n,m}$ by aggregating the ratings of the cluster experts, instead of exploiting users’ trust lists.

### 4 EXPERIMENTS

#### 4.1 Evaluation Dataset and Performance Metrics

We use the Epinions dataset to evaluate the performance of the proposed approach. The dataset is collected from the Epinions.com e-commerce website, where users exchange product information by posting reviews. In the dataset, the task of cold start recommendation involves recommending useful reviews (i.e., items) for cold start users. After reading a review, a user can rate it according to his/her preferences. The dataset has 132,000 users, and contains 1,560,144 reviews and 13,668,319 review ratings. Among the users, 11,514 did not provide any review ratings, so we regarded them as invalid users and excluded them from our evaluations. Among the remaining 120,486 users, 71,073 were deemed cold start users because each one had rated less than 5 items (Victor et al. 2008). Table 1 details the statistics of the dataset. Because it contains a vast amount of the data, Epinions is a popular evaluation dataset. In addition, the dataset records the social relationships between users and has 717,667 trust relations. As a result, it has been used to evaluate the performance of several trust-based recommendation methods.

<table>
<thead>
<tr>
<th># of experienced users</th>
<th>49,413</th>
</tr>
</thead>
<tbody>
<tr>
<td># of cold start users</td>
<td>71,073</td>
</tr>
<tr>
<td># of reviews</td>
<td>1,560,144</td>
</tr>
<tr>
<td># of review ratings</td>
<td>13,668,319</td>
</tr>
<tr>
<td># of trust relations</td>
<td>717,667</td>
</tr>
<tr>
<td>Avg. trust list length – experienced users</td>
<td>12.214</td>
</tr>
<tr>
<td>Avg. trust list length – cold start users</td>
<td>0.679</td>
</tr>
</tbody>
</table>

Table 1. The statistics of the Epinions dataset

We compare our method with the well-known collaborative filtering method (i.e., Equation 1) and the MoleTrust system (i.e., Equation 4), which is also an effective trust-based recommendation system. In the collaborative filtering method, the correlation coefficient is used to select positively correlated users as reference users. The conventional leave-one-out procedure (Avesani and Massa 2007) is adopted to evaluate the performance of the compared methods. For each cold start user, we evaluate the recommendation performance of the three methods over multiple runs. Each evaluation run treats one rated review as an un-rated item and predicts a rating for it by using the information about the remaining rated reviews. The results of all the evaluation runs of each method are averaged to obtain the overall recommendation performance. The evaluation metrics are the mean absolute error (MAE), the coverage rate, and the execution time. The MAE measures the average distance between an item’s real rating and the predicted rating made by a recommendation method. The lower the value, the better will be the method’s performance. The coverage rate measures the percentage of successful predictions made by a method as follows:
\[
\text{coverage} = \frac{\sum_{u_{\text{cold}}} \sum_{n,m > 0} \text{predictable}(\hat{u}_{n,m})}{\sum_{u_{\text{cold}}} |U_n|},
\]

where \( U_{\text{cold}} \) represents the set of cold start users. The \( \text{predictable}() \) function returns 1 if a method can derive \( \hat{u}_{n,m} \) from the profile of \( u_n \); otherwise, it returns 0. Herlocker et al. (2004) posited that the coverage rate is an important metric because the MAE cannot reflect the real utility of a method. Due to the sparsity and cold start problems, recommendation systems usually have a limited number of user profiles. As a result, the profiles may identify biased reference users who have never rated \( i_m \) and cannot aggregate \( \hat{u}_{n,m} \). The coverage rate is especially important in solutions for the cold start recommendation problem because cold start users relatively need recommendations. We also measure the time needed by the compared methods in order to assess their efficiency.

### 4.2 Performance Evaluations

**Figure 4. The System Performance**
Figure 4 shows the performance of our method. The number of clusters (i.e., $K$) is set at 40 because of its superior performance. It is noteworthy that the method’s coverage rate increases as the number of selected experts (i.e., $|\text{exp}_k|$) increases. This is because a large set of experts help aggregate the preferences of an un-rated item, which increases the chances of producing successful rating predictions. However, the improvement in coverage derived by including a new expert is not significant when $|\text{exp}_k| = 6, 7, 8, 9,$ and $10$. The result indicates that the first five experts are sufficiently representative since their preferences cover nearly all the preferences of users in the cluster. The MAE of our method also increases as the number of experts increases. Because the experts are ranked according to their trust scores, the quality of low-ranked experts is generally inferior to that of top-ranked experts. Consequently, recommendations derived from a large set of experts are error-prone. Finally, our method’s execution time is stable because users’ trust scores are computed during the construction of the model. Therefore, including new experts is a constant time operation that does not affect the execution time of making predictions. The figure also shows the advantage of identifying implicit links. By using the implicit link information, we can improve the quality of the selected experts and thus reduce the MAE without a significant reduction in the coverage rate.

Table 2 shows the performance of the compared methods. MoleTrust-0 denotes the MoleTrust method without propagating a web of trust; that is, the method exploits the users on the trust list of a cold start user to make recommendations. We also show the performance of MoleTrust with propagating a web of trust in Table 3. For instance, MoleTrust-1 represents MoleTrust with one level of propagation. In other words, it combines the recommendations made by trusted users and the users trusted by the trusted users to predict $\hat{u}_{n,m}$. For our method, the number of clusters is set at 40; $|\text{exp}_k|$ denotes the number of cluster experts used to make recommendations, and ILI is the acronym for implicit link identification.

<table>
<thead>
<tr>
<th>Method</th>
<th>MAE</th>
<th>Coverage</th>
<th>Time(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaborative filtering</td>
<td>0.68</td>
<td>3.2%</td>
<td>83.3</td>
</tr>
<tr>
<td>MoleTrust-0</td>
<td>0.69</td>
<td>5.7%</td>
<td>13.5</td>
</tr>
<tr>
<td>Our method ($K=40$, $</td>
<td>\text{exp}_k</td>
<td>=1$)</td>
<td>0.69</td>
</tr>
<tr>
<td>Our method ($K=40$, $</td>
<td>\text{exp}_k</td>
<td>=1$, ILI)</td>
<td>0.69</td>
</tr>
</tbody>
</table>

Table 2. The results of the compared methods

<table>
<thead>
<tr>
<th>Method</th>
<th>MAE</th>
<th>Coverage</th>
<th>Time(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoleTrust-1</td>
<td>0.70</td>
<td>12.5%</td>
<td>37.7</td>
</tr>
<tr>
<td>MoleTrust-2</td>
<td>0.74</td>
<td>26.6%</td>
<td>1150.0</td>
</tr>
<tr>
<td>Our method ($K=40$, $</td>
<td>\text{exp}_k</td>
<td>=5$)</td>
<td>0.73</td>
</tr>
<tr>
<td>Our method ($K=40$, $</td>
<td>\text{exp}_k</td>
<td>=5$, ILI)</td>
<td>0.72</td>
</tr>
</tbody>
</table>

Table 3. The results of the compared methods with trust network propagation

As shown in Table 2, the collaborative filtering method produces the worst coverage rate. As the profiles of cold start users are very short, the method has difficulty finding reference users; hence, the coverage rate is low. However, the identified reference users generally provide correct recommendations, so the method’s MAE score is low. By using a web of trust, MoleTrust and the proposed method improve the coverage rate of cold start recommendations without a significant reduction in the recommendation precision. Our method achieves the best coverage rate. The significant improvement over the MoleTrust system indicates that 1) cold start users cannot normally compile effective trust lists; and 2) the experts suggested by our method are trustworthy and can therefore make effective recommendations. The collaborative filtering method requires the most time to make a recommendation because it needs to scan all the ratings of all users to identify a set of reference users. In contrast, the model-based MoleTrust method and our method reduce the execution time by using the constructed user models.
Table 3 shows that by propagating a web of truth by one level, MoleTrust produces a coverage range comparable to that of our method. However, MoleTrust requires a great deal of time to collect the neighbors of trusted users, and the total execution time (37.7 ms) is twice as long as that of our method. Although our method only considers 5 experts, its coverage rates are superior. To achieve a comparable coverage rate, MoleTrust must propagate a web of trust on two levels. However, the complex propagation process degrades the response time such that the method becomes inefficient. In contrast, users’ trust scores are computed during the construction of our model, so including new experts does not have a significant impact on the execution time.

To summarize, our method’s superior coverage rates demonstrate that the experts it identifies are trustworthy. Even one expert is sufficient to provide reliable recommendations. In addition, as the experts and their trust scores are pre-computed in the model construction stage, no extra computation is required to make precise and efficient recommendations.

5 CONCLUSION AND FUTURE WORK

Research on cold start recommendations is important because retaining new users is the key to the success of e-services and e-commerce systems. Generally, recommendations provided by experts are useful. However, new users normally do not know who they can trust. To resolve the problem, we have proposed a cold start recommendation method that analyzes the web of trust of e-commerce users in order to identify trustworthy experts and derive useful recommendations for new users. Experiments based on the well-known Epinions dataset demonstrate that the proposed method is effective and efficient, and outperforms the well-known recommendation methods.

In this work, we have focused on users’ trust networks. However, many recommendation systems also track instances of distrust among users. In our future work, we will investigate using a web of distrust to identify untrustworthy users in e-commerce systems. Cross-referencing trustworthy and untrustworthy users may help us refine the quality of experts and improve our method’s recommendation performance. In addition, blocking the items recommended by untrustworthy users will enable new users to access valuable information and thereby improve the reputation of the e-commerce system. We will also investigate the effectiveness of the system parameters, such as the size of user clusters and the number of cluster experts, and propose effective methods to derive appropriate parameter values.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was supported in part by NSC 99-2221-E-002-182.

References


