An Adaptive Recommendation Method Based on Small-World Implicit Trust Network

Fuzhi Zhang, Huan Wang, Huawei Yi
School of Information Science and Engineering, Yanshan University, Qinhuangdao, China
The Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province, Qinhuangdao, China
Email: xjzfz@ysu.edu.cn, WangHuanaaa@126.com, yihuawei@126.com

***Abstract*—Collaborative filtering (CF) is widely used in e-commerce recommender systems, which helps the online users to identify the right products to purchase. However, CF-based recommender systems suffer poor quality of recommendation due to the sparsity issue. To address this problem, in this paper we propose an adaptive recommendation method based on small-world implicit trust network. We first present a method to construct the small-world implicit trust network based on user clustering and implicit trust among users. Then we develop an adaptive recommendation algorithm by taking into account the topology of the constructed trust network, which generates recommendations using different strategies. To demonstrate the effectiveness of the proposed method, we conduct experiments on the MovieLens dataset and compare our method with others. Experimental results show that the proposed method can significantly improve the quality of recommendation.**

***Index Terms*—data sparsity, user clustering, implicit trust, small-world network, adaptive recommendation algorithm**

I. Introduction

Recommender systems can help to solve the problem of information overload on the Internet by providing personalized recommendations for users [1] [2]. Among the recommendation approaches, collaborative filtering has been proved to be one of the most successful techniques used in recommender systems [3]. CF-based methods generate recommendations based on the similarity between users or items. Due to the problem of data sparsity [4], however, the similarity between users or items can not be calculated accurately, which leads to poor quality of recommendation for CF-based methods.

To address the sparsity problem, various approaches have been proposed. One of them is matrix completion, which reduces the sparsity by filling null ratings. Zhang et al. [5] use BP neural network algorithm to fill null ratings. This method can resist the noise data, but it takes more time to search for the nearest neighbor. Chedrawy et al. [6] calculate the similarity between items according to the attribute association among them, and use the most relevant information to fill null ratings. However, items in different categories may have different descriptions, so this method can't make prediction among different categories.

Dimension reduction is commonly used to solve the sparsity problem. Cheng et al. [7] and Zhang et al. [8] use the matrix factorization based on least trimmed squares and singular value decomposition to reduce the dimension of the matrix, respectively. The results showed that recommendation algorithms based on the low-dimension matrix can alleviate the sparsity problem. However, some potential useful information may be lost during the process of reduction.

Hybrid recommendation method is also used to overcome the sparsity issue. The most common hybrid model is constructed by combining CF with other algorithms. Burke [9] and Choi et al [10] propose a hybrid model, which combines CF and content-based method. Ye et al. [11] present a recommendation method combined association rules mining and CF. Although the hybrid method has been successful in solving the sparsity issue, it has some limitations such as the availability of user profiles and their explicit suggestions on target items. Therefore, this method is difficult to be widely used in practice.

Trust has been introduced by researchers as a solution to the sparsity problem. O'Donovan et al. [12] propose the profile-level and item-level trust models and show that these trust models can improve the accuracy of recommendation. However, trust generated by the above models depends on the similarity between users, so it is difficult to calculate the trust value when the dataset is sparse. Papagelis et al. [13] present a method to alleviate the sparsity by using trust propagation, but this method encounters poor performance when the data is extremely sparse. Similarly, Seo et al. [14] construct a trust network and use link prediction and clustering algorithm to solve the sparsity problem. However, this method requires complex combining of algorithms, which is difficult to realize.

Yuan et al. [15] confirm the small-world property of trust network by experiments, and use this property to optimize the conventional trust-aware recommender systems. However, this work focuses on using the explicit trust among users, which is time consuming or expensive to get. Shu et al. [16] apply the small-world property to the implicit trust network and optimize the process of

This work was supported by the National Natural Science Foundation of China (No.61379116), the Natural Science Foundation of Hebei Province, China (No. F2011203219, No. F2013203124), and the Key Program of Research on Science and Technology of Higher Education Institutions of Hebei Province, China (No. ZH2012028).

doi:10.4304/jcp.9.3.618-625

prediction by setting the maximum distance of trust propagation, so the quality of recommendation can be improved effectively.

To solve the problem of data sparsity, in this paper we propose an adaptive recommendation method based on small-world implicit trust network. Our contributions are summarized as follows:

(1) We propose a method to construct the small-world implicit trust network based on user clustering and implicit trust among users.

(2) We present an adaptive recommendation algorithm based on the constructed small-world implicit trust network, which generates recommendations for the target user using different strategies.

(3) We conduct experiments on the MovieLens dataset to demonstrate the effectiveness of the proposed method.

II. NOTATIONS AND DEFINITIONS

A. Description of Notations

Table I illustrates some notations used in this paper.

TABLE I.
DESCRIPTION OF NOTAIONS

notation	meaning
D	$D=\{U,I,R\}$,a user-item ratings dataset
U	$U=\{u_1,u_2,\ldots,u_m\}$, a set of m users
I	$I=\{i_1,i_2,\ldots,i_n\}$, a set of n items
R	user-item rating matrix
$r_{i,j}$	the rating of user u_i on item I_j
$\bar{r}_i$	the average rating of user u_i
$I(u_i)$	the item set rated by user u_i
$\|A\|$	the number of elements that are contained in set A
$C(G)$	the clustering coefficient of network G
$L(G)$	the average path length of network G
$C^R(G)$	the clustering coefficient of network G's corresponding random network
$L^R(G)$	the average path length of network G's corresponding random network
cluster	a set of several clusters
center	a set of clustering centers

B. Definitions

Definition 1. (Weighted Similarity). For user u_a, $u_b \in U$, let $I(u_a, u_b)=\{i_j \mid r_{a,j}\neq Ø, r_{b,j}\neq Ø, i_j \in I\}$ be the set of items co-rated by users u_a and u_b. The weighted similarity sim_{u_a,u_b} between u_a and u_b is computed as

$$sim_{u_a,u_b} = \frac{|I(u_a,u_b)|}{|I(u_a)\cup I(u_b)|}\frac{\sum_{i_j\in I(u_a,u_b)}(r_{a,j}-\bar{r}_a)(r_{b,j}-\bar{r}_b)}{\sqrt{\sum_{i_j\in I(u_a,u_b)}(r_{a,j}-\bar{r}_a)^2}\sqrt{\sum_{i_j\in I(u_a,u_b)}(r_{b,j}-\bar{r}_b)^2}} \quad (1)$$

Definition 2. (Rating Prediction Error). For user u_a, $u_b \in U$, the rating prediction error of user u_b to u_a, which is denoted by $E_{u_b\to u_a}$, is computed as

$$E_{u_b\to u_a} = \frac{1}{|I(u_a,u_b)|}\sum_{j\in I(u_a,u_b)}\frac{|p^b_{a,j}-r_{a,j}|}{r_{\max}} \quad (2)$$

where $p^b_{a,j} = \bar{r}_a + (r_{b,j}-\bar{r}_b)$, $r_{\max}$ is the maximum rating.

Definition 3. (Implicit Trust). For user u_a, $u_b \in U$, the implicit trust $t_{u_a\to u_b}$ of u_a with respect to u_b is computed as

$$t_{u_a\to u_b} = \frac{1}{2}+\left(\frac{\arcsin(sim_{u_a,u_b})}{\pi}\right)(1-E_{u_b\to u_a}) \quad (3)$$

Definition 4. (Inferred Trust). For user u_0, $u_k \in U$, let $NI=\{u_i|i=1,2,\ldots k\text{-}1\}$ be a set of intermediate nodes that connect u_0 and u_k, the inferred trust $it_{u_0\to u_k}$ of u_0 with respect to u_k is computed as

$$it_{u_0\to u_k} = \prod_{i=0}^{k-1} t_{u_i\to u_{i+1}} \quad (4)$$

where $t_{u_i\to u_{i+1}}$ is the implicit trust of user u_i to u_{i+1}, which is computed by Formula (3).

For an arbitrary network G, let N be the number of nodes in G, E_i be the number of edges that connect the neighbors of node i, k_i be the degree of node i, k be the average degree of all the nodes, $d_{i,j}$ be the distance between nodes of G. Then the structural properties $C(G)$, $L(G)$ and $L^R(G)$ can be described as

$$C(G) = \frac{1}{N}\sum_{i=1}^{N}\frac{2E_i}{k_i(k_i-1)} \quad (5)$$

$$L(G) = \frac{1}{N(N-1)}\sum_{i,j=1}^{N} d_{i,j} \quad (6)$$

$$L^R(G) = \frac{\ln(N)}{\ln(k)} \quad (7)$$

Definition 5. (Small-World Implicit Trust Network). Let the weighted directed graph $G(V,E,W)$ be a trust network, where $V(G)$ is the set of users, $E(G)$ is the set of implicit trust relationships between users, and $W(G)$ is a mapping from $E(G)$ to (0,1]. If the structural properties of the trust network G meet the following conditions:

(1)$C(G)$ is much larger than $C^R(G)$,

(2)$L(G)$ is almost as small as $L^R(G)$,

then the trust network G is called a small-world implicit trust network.

Definition 6. (Sparsity Level). Given a user-item rating dataset D, let N_E be the number of non-empty ratings in D, $|U|$ be the number of users, $|I|$ be the number of items. The sparsity level S_D of the dataset D is computed as

$$S_D = 1-\frac{N_E}{|U|\times|I|} \quad (8)$$

III. THE CONSTRUCTION OF SMALL-WORLD IMPLICIT TRUST NETWORK

A. Process of Trust Network Construction

The process of constructing small-world implicit trust network is illustrated in Fig. 1. At the first stage, users in the rating dataset are divided into k groups using clustering algorithm, which ensures the constructed network to have a large clustering coefficient. At the second stage, the implicit trust between users in each group is calculated and k subnets are constructed based on the implicit trust relationships. At the last stage, a small-world implicit trust network is constructed by connecting the k subnets.

B. Rating Similarity-Based User Clustering Algorithm

The steps of the proposed user clustering algorithm

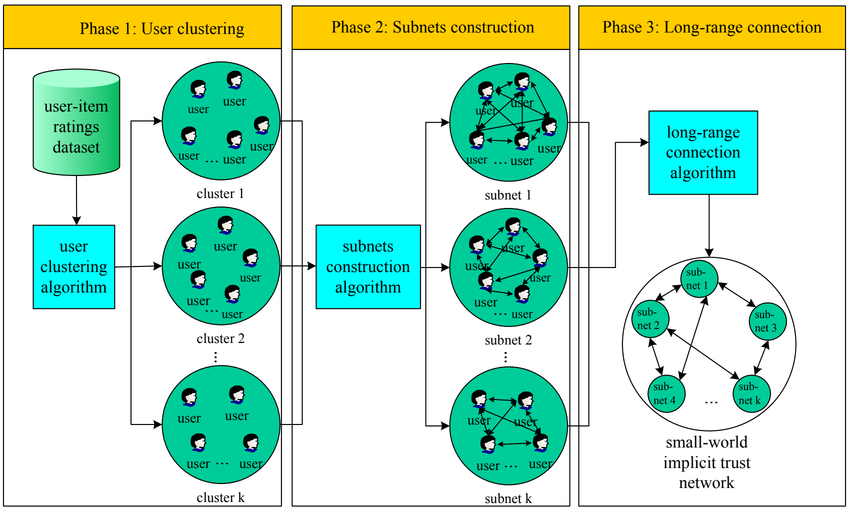


Figure 1. Process of constructing small-world implicit trust network

(UCA) are listed as follows.

Step 1. Treat each user as a single cluster and use the hierarchical clustering algorithm to cluster users, until the number of clusters is equal to k. Let m be the total number of users, we set k to $\lfloor\sqrt{m}\rfloor$ according to the method proposed by Yu et al [17].

Step 2. Calculate the clustering center of each cluster. Suppose each $center_i \in center$ is a virtual user, then the rating attribute of $center_i$ can be denoted by $R_{center_i} = (rc_{i,1}, rc_{i,2}, ..., rc_{i,n})$, where $rc_{i,j}$ is computed as follows:

$$rc_{i,j} = \frac{\sum_{u \in cluster_i} r_{u,j}}{|cluster_i|} \qquad j = 1,2,\ldots,n \tag{9}$$

Step 3. Perform clustering on users using K-means algorithm.

Step 4. Repeat steps 3 and 4 until the result converged.

Based on the above steps, the UCA algorithm is described as follows.

Algorithm 1 UCA

Input: the user-item rating dataset D, the threshold ε

Output: *cluster* and *center*

Begin

1 $num \leftarrow m$, $k \leftarrow \lfloor\sqrt{m}\rfloor$;
2 **for** l=1 to m **do**
3 $cluster_l \leftarrow u_l$;
// treat each user as a single cluster
4 **end for**
5 **for** i=1 to m **do**
6 **for** j=1 to m **do**
7 compute sim_{u_i,u_j} by Formula (1);
8 **end for**
9 **end for**
10 $M_1 \leftarrow \{[s_{ij}]_{m\times m} \mid \forall i \neq j\ s_{ij} = sim_{u_i,u_j}, \forall i = j\ s_{ij} = 0\}$;
11 **repeat**
12 $(u_x, u_y) \leftarrow max(M_1)$;
// get the most similar user u_x and u_y
13 $s_{xy} \leftarrow 0$; // x and $y \in [1,m]$
14 $cluster_x \leftarrow locate(u_x)$;
// get the cluster $cluster_x$ which user u_x belongs to
15 $cluster_y \leftarrow locate(u_y)$;
16 merge $cluster_x$ and $cluster_y$;
17 $num \leftarrow num$-1;
18 **until** num=k;
19 $cluster \leftarrow \{cluster_1, cluster_2, \ldots, cluster_k\}$;
20 **repeat**
21 $old\_cluster \leftarrow cluster$;
22 **for** each $cluster_i \in cluster$ **do**
23 compute $center_i$ by Formula (9);
24 $cluster_i \leftarrow \varnothing$;
25 **end for**
26 $center \leftarrow \{center_1, center_2, \ldots, center_k\}$;
27 **for** each $u_i \in U$ **do**
28 **for** q=1 to k **do**
29 compute $sim_{u_i,center_q}$ by Formula (1);
30 **end for**
31 $M_2 \leftarrow \{[s_{1q}]_{1\times k} \mid s_{1q} = sim_{u_i,center_q}, q = 1,2,\ldots,k\}$;
32 $center_p \leftarrow max(M_2)$;
//get the most similar clustering center $center_p$ of user u_i, $p \in [1,k]$
33 $cluster_p \leftarrow cluster_p \cup u_i$;
34 **end for**
35 $cluster \leftarrow \{cluster_1, cluster_2, \ldots, cluster_k\}$;
36 **until** $\|cluster - old\_cluster\| \leq ε$
37 **return** *cluster* and *center*

End

Algorithm 1 consists of two parts: The first part (lines 1 to 19) is to get the initial *cluster* by using the hierarchical clustering algorithm. The second part (lines 20 to 37) is to perform clustering on users by means of the K-means algorithm.

C. Implicit Trust-Based Subnet Construction Algorithm

In this subsection, a subnet construction algorithm (SCA) is proposed based on the implicit trust among users in a cluster. The steps of SCA are listed as follows.

Step 1. Treat users in each cluster as a set of nodes.

Step 2. Establish edges between nodes according to user trust relationships and generate a set of edges for each cluster.

Step 3. Create a weight set for each cluster based on the calculated trust values between users.

Based on the above steps, the SCA algorithm is described as follows.

Algorithm 2 SCA
Input: *cluster*
Output: a set of subnets $Sub=\{Sub_1,Sub_2,\ldots,Sub_k\}$
Begin
for i=1 to k **do**
  $V_i \leftarrow$ {the users of $cluster_i$};
  $E_i \leftarrow \varnothing$;
  $W_i \leftarrow \varnothing$;
  for each pair of users $(u_a, u_b) \in V_i$ **do**
    compute the implicit trust $t_{u_a \to u_b}$ and $t_{u_b \to u_a}$ by Formula (3);
    if $t_{u_a \to u_b} > 0$ **then**
      $E_i \leftarrow E_i \cup edge_{u_a \to u_b}$; $W_i \leftarrow W_i \cup t_{u_a \to u_b}$;
    end if
    if $t_{u_b \to u_a} > 0$ **then**
      $E_i \leftarrow E_i \cup edge_{u_b \to u_a}$; $W_i \leftarrow W_i \cup t_{u_b \to u_a}$;
    end if
  end for
  $Sub_i \leftarrow G_i(V_i, E_i, W_i)$;
end for
return *Sub*
End

D. Long-range Connection Algorithm

In this subsection, we propose a long-range connection algorithm (LRCA) to connect k subnets constructed by algorithm 2. The steps of LRCA are listed as follows.

Step 1. Infer the correlation among subnets and store the numbers of correlative subnets respectively to create k relevant lists $r=\{r_i|i=1,2,\ldots,k\}$.

Step 2. Calculate the average trust degree of users in each subnet and store the user's number whose average trust degree is greater than $T_{\min}$ to create k top trusted lists $T=\{T_i|i=1,2,\ldots,k\}$.

Step 3. Establish long-range connections among the correlative subnets.

Based on the above steps, the LRCA algorithm is described as follows.

Algorithm 3 LRCA
Input: *Sub*, *center*, and the threshold $T_{\min}$.
Output: the network $G(V,E,W)$
Begin
1 $V \leftarrow \{V_1, V_2,\ldots, V_k\}$;
2 $E \leftarrow \{E_1, E_2,\ldots, E_k\}$;
3 $W \leftarrow \{W_1, W_2,\ldots, W_k\}$;
4 **for** l=1 to k **do**
5   $r_l \leftarrow \varnothing$, $T_l \leftarrow \varnothing$;
6 **end for**
7 **for** each $Sub_i \in Sub$ **do**
8   **for** j=1 to k **do**
9   compute $sim_{center_i,center_j}$ by Formula (1);
10   **if** $sim_{center_i,center_j}>0$ and $j \neq i$ **then**
11       $r_i \leftarrow r_i \cup j$;
12   **end if**
13   **end for**
14   **for** each $u_j \in Sub_i$ **do**
15     $avg\_trust = \dfrac{\sum_{u_k \in sub_i, u_k \neq u_j} t_{u_k \to u_j}}{|sub_i|-1}$;
    //compute the average trust degree of user u_j,
16   **if** $avg\_trust>T_{min}$ **then**
17       $T_i \leftarrow T_i \cup u_j$;
18   **end if**
19   **end for**
20 **end for**
21 **for** each $Sub_i \in Sub$ **do**
22   **for** each $j \in r_i$ **do**
23     $max_1 \leftarrow 0$, $max_2 \leftarrow 0$;
24   **for** each $u_i \in T_i$ and $u_j \in T_j$ **do**
25   compute $t_{u_i \to u_j}$ and $t_{u_j \to u_i}$ by Formula (3);
26   **if** $t_{u_i \to u_j} > max_1$ **then**
27         $max_1 \leftarrow t_{u_i \to u_j}$, $Key_1 \leftarrow u_i$, $Key_2 \leftarrow u_j$;
28   **end if**
29   **if** $t_{u_j \to u_i} > max_2$ **then**
30         $max_2 \leftarrow t_{u_j \to u_i}$, $Key_3 \leftarrow u_j$, $Key_4 \leftarrow u_i$;
31   **end if**
32   **end for**
33     $E \leftarrow E \cup edge_{Key_1,Key_2} \cup edge_{Key_3,Key_4}$;
34     $W \leftarrow W \cup t_{Key_1 \to Key_2} \cup t_{Key_3 \to Key_4}$;
35   **end for**
36 **end for**
37 **return** G
End

Algorithm 3 consists of three parts: The first part (lines 1 to 13) is to calculate the correlation between subnets. The second part (lines 14 to 20) is to calculate the *avg_trust* of all users in each subnet, and add users whose *avg_trust* meets the condition to T. The third part (lines 21 to 37) is to select key nodes for each subnet and establish the long-range connections among the key nodes of the subnets.

We select the MovieLens 100k dataset as experimental data to demonstrate the effectiveness of the above algorithms. We adopt the method proposed by Yuan et al. [18] to prove that the constructed implicit trust network G is a small-world network. We use Formula (5) to calculate the clustering coefficient of the network G and get $C(G)$=0.93888. Since the clustering coefficient of random network is much smaller than 1, the first condition of definition 5 is met. We calculate the average path length of the network G and its corresponding random network by Formulae (6) and (7) respectively, and get $L(G)$=2.886 and $L^R(G)$=2.0137. Obviously, the network G has similar (the same order of magnitude) average path length as its corresponding random network, so the second condition is also met.

Therefore, the constructed implicit trust network G is a small-world implicit trust network.

IV. Adaptive Recommendation Algorithm Based on Small-World Implicit Trust Network

In this section, an adaptive recommendation algorithm (ARA) based on the constructed small-world implicit trust network is proposed.

Let s be the number of similar users in a subnet which the target user belongs to, N_1 and N_2 denote two thresholds ($N_1>N_2$), the following three recommendation strategies can be used when ARA algorithm works.

Strategy 1. When $s\geq N_1$, that is, there are sufficient similar neighbors in a subnet which the target user belongs to, we will use the standard CF method to make recommendations. The predicted rating is calculated by Formula (10).

$$p_{a,i} = \bar{r}_a + \frac{\sum_{b\in R^+} w_{a,b}(r_{b,i} - \bar{r}_b)}{\sum_{b\in R^+} w_{a,b}} \tag{10}$$

where R^+ is a set of similar neighbors, $w_{a,b}$ is the similarity between user u_a and u_b, which is calculated by Formula (1).

Strategy 2. When $s\geq N_2$ and $s<N_1$, we will combine similarity-based and trust-based CF method to make recommendations. The predicted rating is calculated by Formula (11).

$$p_{a,i} = \bar{r}_a + \frac{\sum_{b\in R^+} w_{a,b}(r_{b,i} - \bar{r}_b) + \sum_{b\in R^T\setminus R^+} t_{a,b}(r_{b,i} - \bar{r}_b)}{\sum_{b\in R^+} w_{a,b} + \sum_{b\in R^T\setminus R^+} t_{a,b}} \tag{11}$$

where R^T is a set of trust neighbors, $t_{a,b}$ is the degree of trust between user u_a and u_b, which is calculated by Formula (4).

Strategy 3. When $s<N_2$, that is, there are few similar neighbors in a subnet the target user belongs to, we will use trust-based CF method to make recommendations. The predicted rating is calculated by Formula (12).

$$p_{a,i} = \bar{r}_a + \frac{\sum_{b\in R^T} t_{a,b}(r_{b,i} - \bar{r}_b)}{\sum_{b\in R^T} t_{a,b}} \tag{12}$$

where $t_{a,b}$ is the degree of trust between user u_a and u_b. If the target user and its neighbors are in the same subset, then $t_{a,b}$ is calculated by Formula (3), otherwise it is calculated by Formula (4).

According to the above recommendation strategies, ARA algorithm is described as follows.

Algorithm 4 ARA

Input: user-item rating matrix R, small-world implicit trust network G, a set of relevant lists r, a set of trusted lists T, target user u_a, target item I_i, the number of neighbors l and the threshold N_1, N_2.

Output: predicted rating $P_{a,i}$.

Begin

1 $Sub_t \leftarrow locate(u_a)$;
// get subnet Sub_t which the target user u_a belongs to
2 $su \leftarrow \emptyset$;
3 **for** each $u_j \in Sub_t$ **do**
4 compute sim_{u_a,u_j} by Formula (1);
5 **if** $u_j \neq u_a$ and $r_{j,i} \neq 0$ and $sim_{u_a,u_j} > 0$ **then**
6 $su \leftarrow su \cup u_j$;
7 **end if**
8 **end for**
9 $s \leftarrow |su|$;
10 **if** $s >= N_1$ **then**
11 $R^+ \leftarrow$ top(su, l);
//select the l most similar users from su
12 compute $P_{a,i}$ by Formula (10);
13 **else if** $s< N_1$ and $s >= N_2$ **then**
14 $R^+ \leftarrow su$;
15 $tu \leftarrow \emptyset$;
16 **for** each $c \in r_t$ **do**
//r_t is the relevance list of Sub_t
17 **for** each $u_k \in T_c$ **do**
// T_c is the trusted list of Sub_c
18 compute $it_{u_a \to u_k}$ by Formula (4);
19 **if** $r_{k,i} \neq 0$ and $it_{u_a \to u_k} > 0$ **then**
20 $tu \leftarrow tu \cup u_k$
21 **end if**
22 **end for**
23 **end for**
24 $R^T \leftarrow$ top(tu, l);
25 compute $P_{a,i}$ by Formula (11);
26 **else**
27 $t_1 \leftarrow \emptyset$, $t_2 \leftarrow \emptyset$;
28 **for** each $u_m \in Sub_t$ **do**
29 compute $t_{u_a \to u_m}$ by Formula (3);
30 **if** $u_m \neq u_a$ and $r_{m,i} \neq 0$ and $t_{u_a \to u_m} > 0$ **then**
31 $t_1 \leftarrow t_1 \cup u_m$;
32 **end if**
33 **end for**
34 **for** each $c \in r_t$ **do**
35 **for** each $u_n \in T_c$ **do**
36 compute $it_{u_a \to u_n}$ by Formula (4);
37 **if** $r_{k,i} \neq 0$ and $it_{u_a \to u_n} > 0$ **then**
38 $t_2 \leftarrow t_2 \cup u_n$
39 **end if**
40 **end for**
41 **end for**
42 $tu \leftarrow t_1 \cup t_2$;
43 $R^T \leftarrow$ top(tu, l);
44 compute $P_{a,i}$ by Formula (12);
45 **end if**
46 **return** $P_{a,i}$

End

Algorithm 4 consists of four parts: The first part (lines 1 to 9) is to locate the subnet which the target user belongs to and calculate the number of similar users in the subnet. The second part (lines 10 to 12) is to make recommendations using the similarity-based strategy. The third part (lines 13 to 25) is to make recommendations using the reconciliation strategy. The last part (lines 26 to 45) is to search for the trust neighbors in the small-world implicit trust network and make recommendations using the trust-based strategy.

V. Experimental Evaluation

A. Experimental Data and Settings

We select the Movielens 100K dataset as the experimental data. This dataset contains 100000 ratings from 943 users on 1682 movies. Movies are rated on a scale of one to five, and each user has rated at least 20 movies. The sparsity level of this dataset is 1-100000 ÷ (943 × 1682)=0.937.

For the purpose of experiments, we use random sampling technique to extract 80000, 60000 and 40000 ratings from the original dataset and generate three sampled datasets. These sampled datasets have the same number of users and movies with the original dataset, but they are sparser than the original one. The sparsity level of three sampled datasets is 0.949, 0.962 and 0.975 respectively. Each of the sampled datasets is divided randomly in a ratio 80:20 into training and test sets.

B. Evaluation Metrics

The MAE and F-measure metrics are used to measure the performance of the proposed ARA algorithm.

MAE is commonly used in recommender systems as the measurement of predictive accuracy. The smaller the MAE is, the higher the predictive accuracy of algorithm is. The MAE is defined as follows:

$$\text{MAE} = \frac{\sum_{j=1}^{n} |p_j - r_j|}{n} \tag{13}$$

Where n is the number of items, p_j is the predicted rating on item I_j , r_j is the actual rating.

F-measure is commonly used to evaluate how well the recommendation lists match the user's preferences. The bigger the F-measure is, the higher the quality of recommendation is. The F-measure is defined as follows:

$$\text{F-measure} = \frac{2 \times recall \times precision}{recall + precision} \tag{14}$$

C. Experimental Results and Analysis

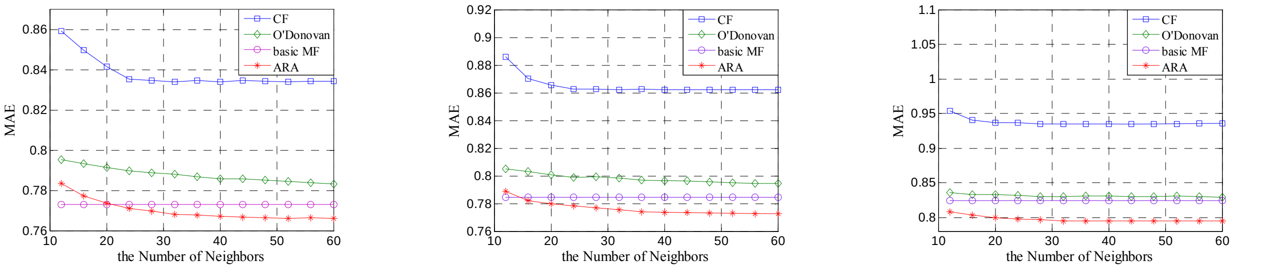
To evaluate the performance of ARA algorithm, we conduct experiments on the three sampled datasets and compare ARA algorithm with the following algorithms.

(1) CF: The conventional user-based collaborative filtering algorithm.

(2) O'Donovan: The collaborative filtering algorithm based on the item-level trust model proposed by O'Donovan.

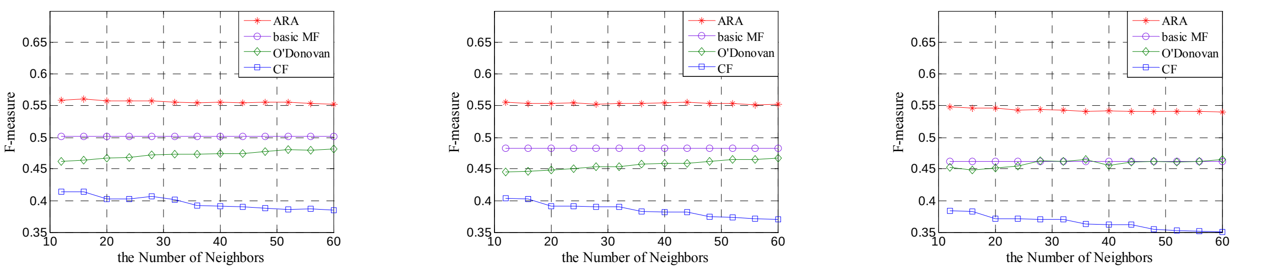
(3) basic MF: The collaborative filtering algorithm based on basic matrix factorization.

Figures 2 and 3 show the comparison of MAE and F-measure for four algorithms on the three sampled datasets.



(a) sampled dataset with sparsity level 0.949 (b) sampled dataset with sparsity level 0.962 (c) sampled dataset with sparsity level 0.975

Figure 2. Comparison of MAE on three sampled datasets in different sparsity levels



(a) sampled dataset with sparsity level 0.949 (b) sampled dataset with sparsity level 0.962 (c) sampled dataset with sparsity level 0.975

Figure 3. Comparison of F-measure on three sampled datasets in different sparsity levels

As shown in Fig. 2, the MAE of CF algorithm on three sampled datasets is the largest, O'Donovan algorithm comes the second, basic MF is the third, and the MAE of ARA algorithm is the smallest. Take the MAE in Fig. 2(b) and Fig. 2(c) for examples, the predictive accuracy of ARA algorithm in Fig. 2(b) is improved by 10.9%, 3.14% and 1.54% respectively compared with CF, O'Donovan and basic MF. Similarly, the predictive accuracy of ARA algorithm in Fig. 2(c) is improved by 15.24%, 4.37% and 3.62% respectively compared with CF, O'Donovan and basic MF. Therefore, the predictive accuracy of ARA algorithm is higher than that of CF, O'Donovan and basic MF.

As shown in Fig. 3, the F-measure of ARA algorithm on three sampled datasets is obviously higher than that of the other three algorithms. Take the F-measure in Fig. 3(b) for example, the F-measure of ARA algorithm is between 0.552 and 0.556, the F-measure of basic MF is 0.483, the F-measure of O'Donovan algorithm is between 0.448 and 0.467, and the F-measure of CF algorithm is between 0.361 and 0.394. With the increase of sparsity level, F-measure of the four algorithms decreases gradually, but the decrement of F-measure for ARA is the smallest. Take the F-measure in Fig. 3(a) and Fig. 3(c) for examples, the F-measure of ARA in Fig. 3(a) is between 0.552 and 0.558, while it is between 0.540 and 0.548 in Fig. 3(c), the average decrement of F-measure from Fig. 3(a) to Fig. 3(c) is 0.011. Similarly, we can calculate the average decrement of F-measure for basic MF, O'Donovan, and CF is 0.041, 0.014 and 0.029 respectively.

Experimental results on the three sampled datasets in different sparsity levels show that the performance of ARA algorithm is better than that of CF, O'Donovan and basic MF. The reason is that the ARA algorithm makes recommendations using different strategies according to the number of similar users in the subnet which the target user belongs to.

VI. Concluding Remarks

Sparsity has a great impact on the quality of recommendation for collaborative filtering approaches. In this paper we propose a method to solve the sparsity problem by constructing the small-world implicit trust network. The construction of small-world implicit trust network is based on user clustering and implicit trust relationship among users. We devise an adaptive recommendation algorithm (ARA) based on the constructed small-world implicit trust network. Experimental results on the three sampled datasets in different sparsity levels show that the performance of ARA algorithm is better than that of the existing recommendation algorithms.

Due to the dynamic nature of recommender systems, the constructed small-world implicit trust network has to be updated when the new users come. In the future, it is better to develop an algorithm to update the small-world implicit trust network incrementally by menns of the incremental clustering method.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No.61379116), the Natural Science Foundation of Hebei Province, China (No.F2011203219, No.F2013203124), and the Key Program of Research on Science and Technology of Higher Education Institutions of Hebei Province, China (No.ZH2012028).

References

[1]. Li H, Zhang S, Wang X. A Personalization Recommendation Algorithm for E-Commerce[J]. *Journal of Software*, 2013, 8(1):176-183.

[2]. Hu L, Wang W, Wang F, et al. The Design and Implementation of Composite Collaborative Filtering Algorithm for Personalized Recommendation[J]. *Journal of Software*, 2012, 7(9): 2040-2045.

[3]. H. J. Ahn, H. Kang, J. Lee. "Selecting a small number of products for effective user profiling in collaborative filtering," *Expert Systems with Applications*, 2010, 37(4), pp. 3055-3062.

[4]. B. M. Kim, Q. Li, C. S. Park, et al. "A new approach for combining content-based and collaborative filters," *Journal of Intelligent Information Systems*, 2006, 27(1), pp. 79-91.

[5]. F. Zhang, H. Y. Chang. "A collaborative filtering algorithm embedded BP network to ameliorate sparsity issue," *Machine Learning and Cybernetics*, 2005. Proceedings of 2005 International Conference on. IEEE, 2005, 3, pp. 1839-1844.

[6]. Z. Chedrawy, S. S. R. Abidi. "An adaptive personalized recommendation strategy featuring context sensitive content adaptation," *Adaptive Hypermedia and Adaptive Web-Based Systems*. Springer Berlin Heidelberg, 2006, pp. 61-70.

[7]. Z. Cheng, N. Hurley. "Robust collaborative recommendation by least trimmed squares matrix factorization," *Tools with Artificial Intelligence* (*ICTAI*), 2010 22nd IEEE International Conference on. IEEE, 2010, 2, pp. 105-112.

[8]. Z. Q. Zhang, S. J. Qian. "The Research of E-commerce Recommendation System Based on Collaborative Filtering Technology," *Advances in Computer Science and Information Engineering*. Springer Berlin Heidelberg, 2012, pp. 507-512.

[9]. R. Burke, "Hybrid recommender systems: Survey and experiments," *User Model. User-Adapted Interact*, 2002, 12(4), pp. 331–370.

[10]. S. Choi. H, Y. S. Jeong, M. K. Jeong. "A hybrid recommendation method with reduced data for large-scale application," *Systems, Man, and Cybernetics*, Part C: Applications and Reviews, IEEE Transactions on, 2010, 40(5), pp. 557-566.

[11]. H. W. Ye, "A Personalized collaborative filtering recommendation using association rules mining and self-organizing map," *Journal of Software*, 2011, 6(4), pp.732-739.

[12]. J. O'Donovan, B. Smyth. "Trust in recommender systems," *Proceedings of the 10th international conference on Intelligent User Interfaces*. ACM, 2005, pp. 167-174.

[13]. M. Papagelis, D. Plexousakis, T. Kutsuras. "Alleviating the sparsity problem of collaborative filtering using trust inferences," *Trust management. Springer Berlin Heidelberg*, 2005, pp. 224-239.

[14]. J. Seo, S. Choi, S. Han. "The Method of Trust and Reputation Systems Based on Link Prediction and Clustering," *Trust Management VII*. Springer Berlin Heidelberg, 2013, pp. 223-230.

[15]. W. Yuan, D. Guan, Y. K. Lee, et al. "The small-world trust network," *Applied Intelligence*, 2011, 35(3), pp.399-410.

[16]. L. Shu, W. Yuan, H. C. Chao, et al. "ITARS: trust-aware recommender system using implicit trust networks," *Communications, IET*, 2010, 4(14), pp. 1709-1721.

[17]. Jian. Yu, Qiansheng. Cheng. "The search range of the optimal number of clusters in fuzzy clustering methods," *Science China:Series E*, 2002, 32(2), pp.274-280 (in Chinese).

[18]. W. Yuan, D. Guan, Y. K. Lee, et al. "Improved trust-aware recommender system using small-worldness of trust networks," *Knowledge-Based Systems*, 2010, 23(3), pp. 232-238.

Fuzhi Zhang was born in 1964. Currently, he is a professor and PhD supervisor in school of information science and engineering, Yanshan University, Qinhuangdao, China. His research interests include intelligent information processing, network and information security, and service-oriented computing.

.

Huan Wang was born in 1987. Currently, she is a master degree candidate in school of information science and engineering, Yanshan University, Qinhuangdao, China. Her research interests include complex networks and recommender systems.

Huawei Yi was born in 1978. Currently, she is a PHD candidate in school of information science and engineeringYanshan University, Qinhuangdao, China. Her main research interest is recommender systems