Journal of Engineering and Applied Sciences 12 (22): 6095-6103, 2017
ISSN: 1816-949X

A Trusted-Community Based Framework for Collaborative Filtering Recommender Systems

[1]Satya Keerthi Gorripati and [2]Valli Kumari Vatsavayi
[1]Department of IT, GMR Institute of Technology, Rajam, Andhra Pradesh, India
[2]Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, India

Abstract: Recommender systems support users in the overwhelming task of examining through large quantities of data in order to select appropriate information or items. Unfortunately such systems may be matter to attack by spam users who want to operate the system's recommendations to outfit their needs: to encourage their own items/services or to originate trouble in the recommender system. Attacks can cause the recommender system to become untrustworthy and unreliable, resulting in user dissatisfaction. Traditional recommender systems rely on like-minded neighbors irrespective of their preferences/tastes when computing predictions and assume users are independent and identically distributed and completely ignore the social activities between users which are not reliable. In reality people heavily rely on their friend's recommdations since, social networks demonstrate a strong community effect. Furthermore, people in cluster/group tend to trust each other and share common preferences with each other more than those in outside the groups. Based on this intuition in this framework, architecture of trusted-community recommender system is proposed. User's preferences expressed by incorporating trusted neighbors within community of the target user are merged in order to find the similar preferences. In addition, the worth of merged ratings is measured by the confidence considering the number of ratings inside the community and the percentage of clashes between negative and positive views. Further, the rating confidence is incorporated into the computation of user similarity. The prediction for an unrated item is computed by aggregating the ratings of similar users within community. Experimental results on real-world data set validate that our method overtakes other complements in terms of accuracy.

Key words: Trust, collobarative recommenders, community detection, confidence, framework, architecture

INTRODUCTION

Millions of people are actively participating in online social networks and interacting with other users who they did not experience before. Forming trust among those indirectly unknown connected users plays a major function in implementing security and refining the character of net helps. "Friend of a friend is a friend" is a commonly used phrase in social networks used to identify reputation which in turn records, discovers and utilizes this information to form trusting and influence a user's behavior. These systems are seen as "soft security" mechanisms which use the collaborative approach for assessing the neighbor's behavior in the community, making it possible to identify who preaches and obey the norms.

The personalized recommendation system has grown rapidly since 1990's. Today, in that respect are many recommendation systems. For instance, e-Commerce stores advertise and recommend their new products, entertainment companies recommend movies, songs, travelling agencies recommend their restaurants and some websites provides personalized news information. The prominent approaches in recommendation systems are listed under:

Collaborative approach: The active user is recommended the appreciated items of that people with similar preferences or tastes in the past.

Content-based recommendation approach: The active user is recommended items similar to her preferred contents/documents.

Hybrid-based recommendation approach: The approach which combines both collaborative and content based approaches.

Traditional approaches usually depend on input called user rating information, however, this information may become vulnerable if the supplied users are

Corresponding Author: Satya Keerthi Gorripati, Department of IT, GMR Institute of Technology, Rajam, Andhra Pradesh, India

malicious. Attackers are either a target item by placing a false maximum value or demote a target item by setting the rating to a minimum value in order to deviate the recommendation. In order to affiliate with other lawful users in the system, the attacker profile will contain the ratings for unseen movies these may be collected based on the prior knowledge of the ratings. To come up to these matters and to model accurate preferences, additional information is supplied to CF algorithms like friendship and social confidence. Both implicit and explicit trust has been investigated in the literature. The old trust is derived from rating information of the user and the latter is at once supplied by users. However, explicit trust is more reliable than the implicit trust. Our approach focuses on explicit trust. Although, many trust-based attacks have been proposed, there is even a demand for a more accurate trust based approaches.

Hence, we propose a trusted community recommender system by incorporating the trusted neighbor in the community structure explicitly given by the target users in the system, aiming to improve the quality of recommender systems and to ease the usual problems of CF approaches. This quality is valued by the confidence considering the number of ratings and the ratio of engagements between positive and negative evaluations. These evaluations are then applied to map the active user's preference and to identify user similarity. Farther, the computation of user similarity is somewhat modified by adding the rating confidence. In conclusion, our method is integrated into a conventional community based CF approach to issue recommendations. Experiments are conducted in real-world data sets to determine the excellence and usefulness of our approach in terms of coverage and accuracy. The results approve that our framework achieves promising performance, especially in conditions of cold users.

MATERIALS AND METHODS

There are three major methods available in recommender systems namely content-based, collaborative-based and hybrid-based (Tang *et al*., 2013). Collaborative-based approach categorized into model-based and memory-based (Goldberg *et al*., 1992; Su and Khoshgoftaar , 2009).

Model-based collaborative approach: A user model is constructed in an offline phase using matrix factorization approaches like SVD (Koren *et al*., 2009), NNMF (Zhang *et al*., 2006) and so on and then the same model is used to generate recommendations in model-based CF.

Memory-based collaborative approach: This approach is further divided into User-oriented (UBCF) and Item-oriented (IBCF) approaches. User-oriented approaches predict the unrated item by using a weighted norm of all the similar users on the item while item-oriented approach predicts the unrated item by using a weighted norm of all similar items by the same user. There are many approaches to compute this similarity such as Pearson (Resnick *et al*., 1994), Cosine (Chowdhury, 2010) and probability based (Karypis, 2001), among which Pearson similarity is the commonly practiced one. Many approaches have proposed for specifying neighborhood by using specific methods such as threshold similarity, random neighbors and top-N neighbors. Another amendment to reduce this neighborhood is community detection methods for recommender systems. In the recent past, community algorithms have been applied to produce a group of users with similar users called communities and similarity measures are used to in order to find the nearest neighborhood of the active user. Several researchers explored the views of communities using popular algorithms like LabelRankT (Xie *et al*., 2013), Louvian (Blondel *et al*., 2008; Parimi and Caragea, 2014), Infomap (Rosvall and Bergstrom, 2008), iLCD (Alvari *et al*., 2016) and so on.

Broadly, the CF approach suffers from the data sparsity and cold-start problems. Many advances have been suggested to address these matters. A feasible solution is to restrict the unrepresentative items in the user-item matrix to reduce the sparsity (Ramezani *et al*., 2013). Some other potential solution is towards building a recommender system for individual as well for a group. A group based RS focuses on a group of users in social networks (Fatemi and Tokarchuk, 2013), elaborated the definition of community membership by including degree utility set and ranked adjacencies. Another group approach proposed by Kim *et al*. (2010) is a two phased algorithm. Firstly, user's community is discovered and individual profiles are provided for the active user by using the other user's data in the community.

Another distinction in recommender systems to address the accuracy and flaws in the CF approach is to integrate the trust information along with similarity. In most social networks user can express his preferences and web of trust explicitly (i.e., ratings and their opinions). Many algorithms have been proposed in the literature to make use of the explicit trust information. One such method is TrustWalker where a user is represented as a node and a link established between trusted users and the power of the edge represents credibility (Jamali and Ester, 2009). The approaches (Massa and Avesani, 2007), addressed the failings of collaborative based approach and work out how to override them by incorporating trust. The algorithm MoleTrust uses depth-first search to indirect trust paths in the trusted networks. The researcher altered the search with breadth-first in TidalTrust (Golbeck, 2005) to calculate the trust value. The proposed framework is replica of Mergex (Guo *et al.*, 2014) approach in the sense of suggesting top recommendations to the active user based on his/her profile. However, there are few conflicts between these systems.

The goal of the proposed approach is to apply community detection algorithm (Angadi and Varma, 2015, 2016) on the whole graph to form user groups based on their preferences/tastes. Whereas, the goal of the Mergex approach is on individual in the trusted network.

The weighted trust values will be averaged by considering all user ratings within the trusted propagation distance. Whereas in our approach the weighted trusted considers only the users inside the target item's community.

Our approach: In this study, we will describe the trusted community method the basic step is to represent the tastes/preferences of target user using the ratings of trusted users. The architecture has three stairs to produce recommendations. First, aggregate the trusted neighbor's ratings of the target user inside the community. Trust propagation may be necessary to integrate more trusted neighbors, predominantly useful for the cold users. Second, the trusted neighbor's ratings within a community are then mixed into a single value for each unrated item if that item is rated by at least one neighbor. Hence, a new active user's profile of the target user gets created. Ultimately, a similarity metric is calculated based on user's preference profile and using traditional CF approach recommendations are generated.

Pre-processing: For simplicity, we use a number of representations to model the recommendation process. Formally, we represent the set of all users with U and all items with I and ratings with R. We denote u, v for users and i, j for the products/items. Then $r_{i,j}$ denotes rating given by user u on item i ranges from 1-5. Hence, the task of a recommendation is to predict the unrated item of every active user which is a triplet (u, i, j). In this framework for the active user a set of neighbors belongs to the community of the active user and those are reachable are identified as trusted neighbors and denoted as T_u. For each of these neighbors the active user specifies a trust value which lies in between [0, 1].

Aggregating trusted neighbors: Trust can be spread along with the web-of-trust. That is if user A trusts B and B trusts C, it can be inferred as A trusts C. Algorithms like MoleTrust and TidalTrust (Massa and Avesani, 2007; Golbeck, 2005) in literature are used inferring trust values. In order to best use the trust information it is necessary to identify trust inferences or indirect trusted neighbors. In this framework, we have implemented MoleTrust to discover the trust of indirect users. The trust values are binary, directly connected users are indicated with 1 and indirectly connected users are indicated with 0 and its value has to be computed. To reinforce the performance of trust-based approaches the frame work is not differentiating the shortest distance and longer distance users instead of that, we adopt a weighting factor (Guo *et al.*, 2014):

$$t_{u,v} = \frac{1}{d} \times \bar{t}_{u,v} \quad (1)$$

Where:

$\bar{t}_{u,v}$ = The inferred trust value
d = The number of hops between user u and v

In this regard directly connected users are more worthy than the indirectly connected users but it infers more trusted neighbors. However, the theory of six degree of separation says that any two users in the world (or network) can connect within <6 steps. If the network is personnel network the value of d will be restricted to 6 but in our research it is restrict to 2 to save the computational cost and unnecessary searching.

A set of users having trust value greater than threshold are identified as trusted neighbors. In addition, every active user trust himself and observed as one of the trusted neighbor in his/her neighborhood, i.e., $t_{v,v} = 1$ and $v \in Nei_v$. In other words, we preserve the ratings given by the user and predict the ratings of unknown items.

Merging the ratings of trusted neighbors: The candidate items for the active user can be identified after finding the trust neighborhood inside the community:

$$L_i = \{i \mid r_{v,i} \in R, \exists v \in Nei_u, \wedge v \in C, i \in I\} \quad (2)$$

That is, L_i is the list of items that have been given rating for the target item and resides in the same community as the active user u. All the ratings given by the trusted users are mixed into a single value using Eq. 3:

$$\overline{r_{u,j}} = \frac{\sum_{v \in C \wedge v \in Nei_u} t_{u,v} \times r_{vj}}{\sum_{v \in C \wedge v \in Nei_u} t_{u,v}} \quad (3)$$

Where:

$\overline{r_{u,j}}$ = The merged value
$t_{u,v}$ = The inferred weighted trust value
r_{vj} = The rating given by the user with a constraint that both the users belongs to the same community and should in the trusted neighbors list

Equation 3 can be modified slightly when the researchers considers rating similarity, social similarity and trust value into consideration. In our approach, we are not considering social similarity our trust value is calculated by taking only rating similarity and trust value using Eq. 4:

$$t_{u,v} = \alpha\times S_{u,v}+\beta\times t_{u,v}+(1-\alpha-\beta)\times j_{u,v} \quad (4)$$

According to Ray and Mahanti (2010), users have positive influence when they are highly similar to each other. Therefore, it is required to consider both trust and value rating similarity. We prefer to use Pearson correlation coefficient to compute user's rating similarity:

$$S_{u,v} = \frac{\sum_{i\in I_{u,v}} (r_{u,i}-\bar{r}_u)(r_{v,i}-\bar{r}_v)}{\sqrt{\sum_{i\in I_{u,v}} (r_{u,i}-\bar{r}_u)^2}\sqrt{\sum_{i\in I_{u,v}} (r_{v,i}-\bar{r}_v)^2}} \quad (5)$$

In particular, if $S_{u,v}>0$, if users are positively correlated and negatively correlated if $S_{u,v}<0$ and since, every active user be a part of his trusted neighbors list, we denote that with $S_{u,u} = 1$.

RESULTS AND DISCUSSION

Finding the confidence of merged ratings: A merged rating for a target user on an unrated item can be computed using Eq. 3 based on the ratings of trusted neighbors belongs to the same community. Nevertheless, the accuracy of this rating is unknown. To validate these following factors are to be considered, one is the number of ratings and the second is the differences between the positive and negative ratings among all users.

More specifically, if an item is cold item (i.e., it has received ratings from less number of users) the merged value tends to be unreliable and noisy. It is reliable if that item gets more ratings. In this framework, we consider the rating as positive opinion when it is greater than the median and negative opinion otherwise:

$$\begin{aligned} &r_{med} = (n+1)/2 \\ &n = r_{max} \\ &+ve: r_{u,i} > r_{med} \\ &-ve: \text{otherwise} \end{aligned} \quad (6)$$

where, r_{max} is the maximum rating in the rating's dataset. The less is the difference between positive and a negative opinion, the more is the efficiency of the merged trust. This can be managed using the measure confidence which contemplates the differences in the number of neighbors and the difference in the conflicts between positive and negative opinions (Wang and Singh, 2007):

$$C_{u,j} = c(p_o, n_o) = \frac{1}{2}\int_0^1 \left| \frac{x^{p_0}(1-x)^{n_0}}{\int_0^1 x^{p_0}(1-x)^{n_0}\,dx} - 1 \right| dx \quad (7)$$

where, p_o, n_o refers to positive and negative opinions of the trusted neighbor and the value of $C_{u,j}$ lies in between (0, 1). The framework produces output of the form $(\overline{r_{u,j}}, C_{u,j})$ which will constitute a new profile of the target user based on which item predictions can be generated.

Integrating with traditional collaborative filtering: We apply a traditional CF approach to predict the unrated item of the active user after which a user profile can be generated. Like in item based CF approach, we go over a set of neighbors who have rated that item and having similarity with the active user are selected. These ratings are then aggregated to produce a single value. The Pearson correlation coefficient is slightly changed in accordance with the confidence in order to check the measure of similarity between users defined Eq. 8:

$$\bar{S}_{u,v} = \frac{\sum_{i\in I_{u,v}} C_{u,i}(r_{u,i}-\bar{r}_u)(r_{v,i}-\bar{r}_v)}{\sqrt{\sum_{i\in I_{u,v}} C_{u,i}^2(r_{u,i}-\bar{r}_u)^2}\sqrt{\sum_{i\in I_{u,v}} C_{u,i}^2(r_{v,i}-\bar{r}_v)^2}} \quad (8)$$

Where:

$\bar{r}_u$, $\bar{r}_v$ = The average ratings for the users u, v, respectively
$C_{u,i}$ = The confidence measurement after the merge method (Xue *et al*., 2005)

This framework uses thresholding method in order to select nearest neighborhood with a group of similar users. However, in traditional CF approaches top-k similar users will be used. For performance reasons some researcher have shown that top-k method is less effective than thresholding method (Guo *et al*., 2014). Therefore, we prefer to use the thresholding method rather than top-k method to find the nearest neighborhood of the target user:

$$\text{Nei}_u = \{v \mid \bar{S}_{u,v} > \theta,\ v \in U \wedge u,\ v \in C\} \tag{9}$$

where θ is a predefined similar value. Finally, to predict the unrated item of active user, the ratings of nearest neighborhood are aggregated. To compute this, we use the average method defined as follows:

$$\overline{r_{u,j}} = \frac{\sum_{v \in \text{Nei}_u \wedge v \in C} \bar{S}_{u,v} \times r_{v,j}}{\sum_{v \in \text{Nei}_u \wedge v \in C} \left|\bar{S}_{u,v}\right|} \tag{10}$$

where, $\overline{r_{u,j}}$ represents the predicted value of an unrated item.

Algorithm 1:
Input: User-item rating matrix and a directed trust graph
I: Number of Items
U: Number of Users
θ: Threshold
au: Active user
d_{max}: Propagation distance to be considered (i.e., 2)
Output: Predictions of unseen items
Begin
1: Build a network with a list of co-rated interactions
2: Apply community detection algorithm to get the communities
3: For each user u compute trust inference using Eq. 1
// Step 1
4: For each user u ϵ U do
5: For each item i ϵ I do
6: If i is an unrated item then
7: Get the list of users rated this item
8: If the user belongs to the same community of the active user and within propagation distance using Eq. 2
9: Predict the rating using Eq. 3
10: End If
11: End If
12: End for
13: End for
// Step 2
14: Identify the confident users for the active user u using Eq. 9
15: Compute the similarity using Eq. 8
16: Predict the ratings using Eq. 10
End algorithm

A running example: In this study, we propose to demonstrate step by step process of trusted community method to generate a prediction for a given unrated item. Suppose there are 15 users and 15 items in a certain system. Every user rate a few interested items by giving a rating between 1-5 as shown in Table 1.

Table 1: User rating table

User ratings	1	2	3	4	5	7	8	9	10	11	12	13	14	15	16
1	-	-	5	-	-	-	-	-	-	-	-	-	4	5	-
2	5	-	-	4	-	3	-	2	-	-	-	3	-	-	5
3	-	4	-	-	-	-	-	1	-	-	4	-	3	-	-
4	-	-	5	-	2	-	-	-	-	-	2	-	-	-	4
5	-	4	4	-	3	-	-	3	-	4	-	3	5	-	-
7	-	3	3	5	5	-	-	-	-	-	-	-	-	-	-
8	3	-	-	-	-	-	5	-	4	-	-	-	-	-	2
9	-	-	4	2	-	-	-	5	-	4	1	3	-	-	5
10	-	-	4	5	-	-	-	-	-	2	-	-	-	-	-
11	4	-	-	-	-	-	4	-	-	-	-	-	-	-	-
12	-	2	4	-	-	3	-	3	4	-	-	4	4	-	-
13	3	-	-	-	4	-	-	-	-	-	-	2	-	-	-
14	-	-	-	-	5	-	-	2	-	-	2	-	-	-	-
15	2	-	-	-	-	-	4	-	-	-	-	3	-	5	-
16	-	-	3	-	-	-	-	-	-	-	2	-	-	-	-

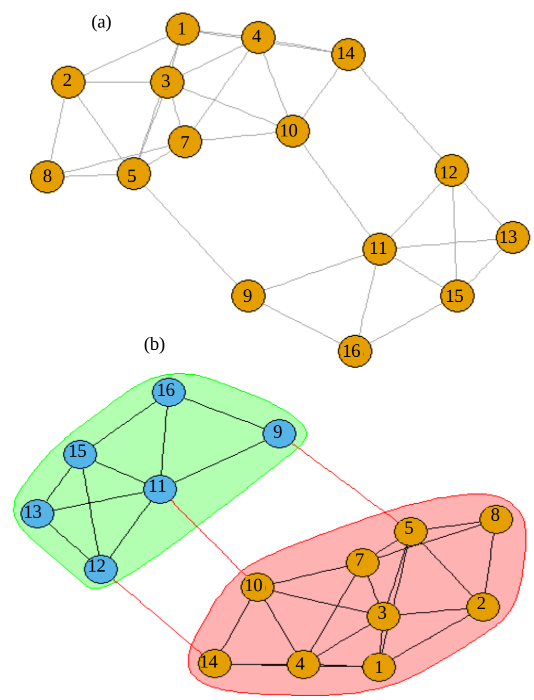

Fig. 1: a) A sample network with co-ratings and b) After applying community algorithm

In addition, users may specify his immediate/adjacent neighbors as trusted neighbors; where an entry 1 indicates that both are trusted in each other. Before going for predicting unrated items, apply community algorithm on Fig. 1. The approach has identified two communities and they are listed here. User 6 has not provided any ratings and has no trusted neighbors (i.e., no adjacencies) in our framework we treat such users as outliers:

$$L_{C1} = \{1, 2, 3, 4, 5, 7, 8, 10, 14\}$$
$$L_{C2} = \{9, 11, 12, 13, 15, 16\}$$

Table 2: Propagation distance for 1st user

Factors	[1,]	[2,]	[3,]	[4,]	[5,]	[7,]	[8,]	[10,]	[14,]
d	1	1	1	1	1	2	2	3	1
$t_{1,k}$	1	1	1	1	1	0.5	0.5	0.33	1

Implicit trusted neighbors in first community:

[1,]	.	1	1	1	1	.	.	.	1
[2,]	1	.	1	.	1	.	1	.	.
[3,]	1	1	.	1	1	1	.	1	.
[4,]	1	.	1	.	.	1	.	1	1
[5,]	1	1	1	.	.	1	1	.	.
[6,]	.	.	1	1	1	.	1	1	.
[7,]	.	1	.	.	1	1	.	.	.
[8,]	.	.	1	1	.	1	.	.	1
[9,]	1	.	.	1	.	.	.	1	.

Implicit trusted neighbors in 2nd community:

[1,]	.	1	.	.	.	1
[2,]	1	.	1	1	1	1
[3,]	.	1	.	1	1	.
[4,]	.	1	1	.	1	.
[5,]	.	1	1	1	.	1
[6,]	1	1	.	.	1	.

In this case, we are concerned with generating a prediction for a target user 1 on a target item 1. User 1 has given ratings for only 3 items in order to build the profile of user 1, we have to predict remaining all unrated items $<u_1, i_1>$, $<u_1, i_2>$, $<u_1, i_4>$, …. The framework supposes that all adjacent nodes are implicit trusted neighbors, i.e., items {2, 3, 4, 5, 14} as his neighbors. This trusted information is in symmetric in nature.

The first step is to see the trusted neighbors of the target user from the network shown in Fig. 1. According to that user 1 belongs to community 1 and its trust values with the others in the same community are shown in sets L_{C1} and L_{C2}. In particular, user 1 as a target user he trust himself and hence, $t_{1,1} = 1$. Since, the users 7 and 8 are separated by one hop in between with user 1 and its trust value is $t_{1,7} = 0.5$ and the shortest path distance between user 7 and user 1 is d = 2 (Table 2). The users with distance (i.e., $d \geq 3$) and outside the community are not regarded as inferred trusted neighbors. Hence, a list of users Nei_1 = {2, 8} are the trusted neighbors for the target user 1 though the users 11, 13 and 15 are rated the target item and their shortest path distance from user 1 is also within the constraint they are not considered as neighbors since they belongs to other community. The calculated trust values between user 1 and other users in the community (Table 3 and 4).

Table 3: Propagation distance

Factors	[9,]	[11,]	[12,]	[13,]	[15,]	[16,]
d	1	1	2	2	2	1
$t_{1,k}$	1	1	0.5	0.5	0.5	1

Second, trust values are considered as user weights and these weights and ratings are taken to compute weighted average of the target user. The following is the prediction of user 1's target item 1 by considering the neighbors of 2nd, 8th user's ratings and their trust values:

$$r_{1,1} = \frac{5\times1+3\times0.5}{1+0.5} = 4.33$$

$$r_{4,9} = \frac{2\times0.5+1\times1+3\times0.5+2\times1}{0.5+1+0.5+1} = 1.8$$

According to equation all the ratings above median rating are observed as positive otherwise observed as negative. The confidence is derived by:

$$C_{u_1, i_1} = C(1, 1) = 0.19$$

This process continues until all unrated items by active user are predicted. A new user profile is formed for every active user as shown in Table 4. We are leaving the actual ratings of the user since he believes his own rating than anyone else.

In the second step, another sample rating is shown in order to show the improvement of our approach in calculating the predictions using confidence based approach. For doing this, we have chosen first user's fourth item for prediction. According to Mergex approach, 10th user is not considered in the prediction process, since its propagation distance is not in the specified range but users 2, 7, 9, 10 are rated item 4. The same user is considered in similarity calculation since, it has the positive correlation with the first user. According to the proposed approach user 9 is not considered since this user does not belongs to the community of the active user and user 10 is not considered with the same reason mentioned above. The prediction using confidence based similarity is calculated similarly to Mergex approach as shown in Table 5 and 6.

Pearson correlation approach is used to get the similar users using Eq. 5, it is slightly altered in order to include the confidence as shown in Eq. 8. The Pearson correlation and confidence based similarities for the first user is shown in Table 5.

Predictions for unrated items for the first user using both approaches (Step 1 and 2 in pseudo code) were shown in Table 6. The computed values (4.58, 0.47) are

Table 4: The computed predictions and confidence for the first user

Factors	1	2	3	4	5	7	8	9	10	11	12	13	14	15	16
r(1, j)	4.3	3.8	5	4.5	3.5	3	5	2	4	3.3	2.6	3	4	5	4
t(1, j)	0.19	0.27	0.4	0.47	0.29	0.25	0.25	0.53	0.25	0.19	0.27	0.38	0.27	0.25	0.27

Table 5: Pearson correlation coefficient and confidence-based correlation values for the first user

Values	Pearson correlation	Confidence based
1	1	1
2	0.89108211	0.7476
3	0.824315308	0.568178
4	0.916357835	0.58269
5	0.639769057	0.51787
7	0.685105113	0.597386
8	0.76181953	0.664213
9	0.214763224	0.128591
10	0.776996597	0.465232
11	0.331065395	0.3311
12	0.382325604	0.32079
13	0.378111915	0.26741
14	0.748248488	0.627743
15	0.270804773	0.236125
16	0.138152254	0.115894

Table 6: Computational differences between mergex method and the proposed approach

For an active user, user 1	Mergex
Mergex approach	
Merged rating profile without considering community	Confidence
$r_{1,4} = \frac{4\times1+5\times0.5+2\times0.5}{1+0.5+0.5} = 3.75$	$C(2,1) = \frac{1}{2}\int_0^1 \left\lvert \frac{x^2(1-x)}{\int_0^1 x^2(1-x)dx} - 1 \right\rvert dx = 0.27$
Merged rating profile by combining confidence based similarity	Confidence
$r_{1,4} = \frac{4\times0.7476+5\times0.597+5\times0.465232+2\times0.1285}{0.7476+0.597+0.465232+0.1285} = 4.41$	$C(3,1) = \frac{1}{2}\int_0^1 \left\lvert \frac{x^3(1-x)}{\int_0^1 x^3(1-x)dx} - 1 \right\rvert dx = 0.34$
Proposed approach	
Merged rating profile by considering community	Confidence
$r_{1,4} = \frac{4\times1+5\times0.5}{1+0.5} = 4.33$	$C(2,0) = \frac{1}{2}\int_0^1 \left\lvert \frac{x^2}{\int_0^1 x^2dx} - 1 \right\rvert dx = 0.38$
Merged rating profile by combining confidence based similarity	Confidence
$r_{1,4} = \frac{4\times0.7476+5\times0.597+5\times0.465232}{0.7476+0.597+0.465232} = 4.58$	$C(3,0) = \frac{1}{2}\int_0^1 \left\lvert \frac{x^3}{\int_0^1 x^3dx} - 1 \right\rvert dx = 0.47$

Table 7: The predictions for the first user using confidence-based approach

Factor	1	2	3	4	5	7	8	9	10	11	12	13	14	15	16
1	4.05	3.64	5	4.58	3.36	3	5	1.9	4	3.05	2.63	3	4	5	3.70

higher compared with the values (4.33, 0.38) when consider the confidence based similarity than the mixed ratings of trusted neighbors and within community shown in Table 6. Finally, the predictions for the first user using Eq. 10 shown in Table 7 and Fig. 2.

Experiments: To verify the performance, we have worked out on real-world dataset namely Epinions dataset contains both ratings data and explicit trust statements. Epinion is a social web site, where users can share their opinion by giving ratings to each movie and discover new

Table 8: Accuracy comparisions

Algorithms	MAE	RC	F-measure
UBCF	0.62	0.41	93.82
IBCF	0.69	0.48	94.12
Mergex	0.71	0.77	94.64
Trusted-community	0.77	0.81	94.73

Fig. 2: Recommended movies for the 1st user

movies. We implement the data set from the website http://www.epinions.com. The ratings hold real values between 0.5-4.0 and the trust statements are ranged from 1-10. Hence, these ratings are converted into binary where a value 1 represents a trusted neighbor and 0 otherwise. We compare the performance of our framework with conventional CF approaches.

IBCF computes and selects neighbors using the Pearson correlation measure, whose similarity is above threshold and give predictions based on selected user's ratings. Mergex denotes an approach to integrate both similarity and trust to improve the performance of traditional collaborative filtering. The proposed method integrates both similarity and trust within community increases the performance compared to IBCF and Mergex method.

Our approach: The performance is measured using metrics like accuracy and coverage. Using the leave-one-out method the actual user's ratings are hidden and the values are predicted using some approach and these errors are accumulated. The metrics are defined as follows. MAE Mean Absolute Error deals with the degree to which a prediction is closed to the truth:

$$MAE = \frac{\sum_u \sum_i \left| \overline{r_{u,i}} - r_{u,i} \right|}{N} \quad (11)$$

Where:

N = The number of users ratings

$\overline{r_{u,i}}, r_{u,i}$ = The predicted and actual ratings

Coverage deals with the degree to which the testing ratings can be predicted and covered relative to the actual ratings:

$$RC = \frac{M}{N}$$

where, M, N are predicted and actual ratings. F-measure: is the overall performance of both merging accuracy and coverage and computed using Eq. 12:

$$F = \frac{2 \times iMAE \times RC}{iMAE + RC} \quad (12)$$

CONCLUSION

This study proposed an innovative method to integrate trusted neighbors into traditional collaborative filtering techniques, aiming to avoid the malicious ratings and to resolve the core problems of traditional recommender systems. Precisely, the weighted average of the trusted neighbors represents the preferences of the active users, based on which related users can be recognized and recommendations are produced. The quality of this approach was computed by the confidence considering the number of ratings involved and the encounters between positive and negative opinions (i.e., ratings). New similarity approach is introduced by integrating confidence into traditional Pearson correlation. The prediction of an unrated is generated by averaging the ratings of similar users with in community weighted by their importance. Experiments on three real-world data set were conducted and the results showed that significant improvements against other methods were obtained both in accuracy and coverage as well as the overall performance.

REFERENCES

Alvari, H., A. Hajibagheri, G. Sukthankar and K. Lakkaraju, 2016. Identifying community structures in dynamic networks. Soc. Netw. Anal. Min., 6: 77-77.

Angadi, A. and P.S. Varma, 2015. Overlapping community detection in temporal networks. Indian J. Sci. Technol., 8: 1-6.

Angadi, A. and P.S. Varma, 2016. Finding hubs and outliers in temporal networks. Indian J. Sci. Technol., 9: 1-5.

Blondel, V.D., J.L. Guillaume, R. Lambiotte and E. Lefebvre, 2008. Fast unfolding of communities in large networks. J. Stat. Mech. 10.1088/1742-5468/2008/10/P10008

Chowdhury, G.G., 2010. Introduction to Modern Information Retrieval. 3rd Edn., Facet Publishing, London, England, ISBN:978-1-85604-694-7, Pages: 514.

Fatemi, M. and L. Tokarchuk, 2013. A community based social recommender system for individuals and groups. Proceedings of the 2013 International Conference on Social Computing (SocialCom'13), September 8-14, 2013, IEEE, Alexandria, Virginia, ISBN:978-0-7695-5137-1, pp: 351-356.

Golbeck, J.A., 2005. Computing and applying trust in web-based social networks. Ph.D Thesis, University of Maryland Libraries, College Park, Maryland.

Goldberg, D., D. Nichols, B.M. Oki and D. Terry, 1992. Using collaborative filtering to weave an information tapestry. Commun. ACM., 35: 61-70.

Guo, G., J. Zhang and D. Thalmann, 2014. Merging trust in collaborative filtering to alleviate data sparsity and cold start. Knowl. Based Syst., 57: 57-68.

Jamali, M. and M. Ester, 2009. Trustwalker: A random walk model for combining trust-based and item-based recommendation. Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, June 28-July 01, 2009, ACM, Paris, France, ISBN:978-1-60558-495-9, pp: 397-406.

Karypis, G., 2001. Evaluation of item-based top-n recommendation algorithms. Proceedings of the 10th International Conference on Information and Knowledge Management, October 05-10, 2001, ACM, Atlanta, Georgia, USA., ISBN:1-58113-436-3, pp: 247-254.

Kim, J.K., H.K. Kim, H.Y. Oh and Y.U. Ryu, 2010. A group recommendation system for online communities. Int. J. Inform. Manage., 30: 212-219.

Koren, Y., R. Bell and C. Volinsky, 2009. Matrix factorization techniques for recommender systems. Computer, 42: 30-37.

Massa, P. and P. Avesani, 2007. Trust-aware recommender systems. Proceedings of the 2007 ACM Conference on Recommender Systems, October 19-20, 2007, ACM, Minneapolis, Minnesota, USA, ISBN: 978-1-59593-730-8, pp: 17-24.

Parimi, R. and D. Caragea, 2014. Community detection on large graph datasets for recommender systems. Proceedings of the 2014 IEEE International Conference on Data Mining Workshop (ICDMW'14), December 14, 2014, IEEE, Shenzhen, China, ISBN:978-1-4799-4273-2, pp: 589-596.

Ramezani, M., P. Moradi and F.A. Tab, 2013. Improve performance of collaborative filtering systems using backward feature selection. Proceedings of the 5th Conference on Information and Knowledge Technology (IKT'13), May 28-30, 2013, IEEE, Shiraz, Iran, ISBN:978-1-4673-6490-4, pp: 225-230.

Ray, S. and A. Mahanti, 2010. Improving prediction accuracy in trust-aware recommender systems. Proceedings of the 2010 43rd Hawaii International Conference on System Sciences (HICSS), January 5-8, 2010, IEEE, Honolulu, Hawaii, ISBN: 978-1-4244-5509-6, pp: 1-9.

Resnick, P., N. Iacovou, M. Suchak, P. Bergstrom and J. Riedl, 1994. GroupLens: An open architecture for collaborative filtering of netnews. Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, October 22-26, 1994, ACM, Chapel Hill, North Carolina, USA, ISBN: 0-89791-689-1, pp: 175-186.

Rosvall, M. and C.T. Bergstrom, 2008. Maps of random walks on complex networks reveal community structure. Proc. National Acad. Sci., 105: 1118-1123.

Su, X. and T.M. Khoshgoftaar, 2009. A survey of collaborative filtering techniques. Adv. Artificial Intell., 10.1155/2009/421425

Tang, J., X. Hu and H. Liu, 2013. Social recommendation: A review. Social Network Anal. Min., 3: 1113-1133.

Wang, Y. and M.P. Singh, 2007. Formal trust model for multiagent systems. Proceedings of the 20th International Joint Conference on Artifical Intelligence Vol. 7, January 06-12, 2007, ACM, Hyderabad, India, pp: 1551-1556.

Xie, J., M. Chen and B.K. Szymanski, 2013. LabelrankT: Incremental community detection in dynamic networks via label propagation. Proceedings of the 2013 Workshop on Dynamic Networks Management and Mining, June 22-27, 2013, ACM, New York, USA., ISBN:978-1-4503-2209-6, pp: 25-32.

Xue, G.R., C. Lin, Q. Yang, W. Xi and H.J. Zeng et al., 2005. Scalable collaborative filtering using cluster-based smoothing. Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 2005, ACM, Salvador, Brazil, ISBN:1-59593-034-5, pp: 114-121.

Zhang, S., W. Wang, J. Ford and F. Makedon, 2006. Learning from incomplete ratings using non-negative matrix factorization. Proceedings of the 2006 SIAM International Conference on Data Mining, April 20-22, 2006, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, ISBN:978-0-89871-611-5, pp: 549-553.