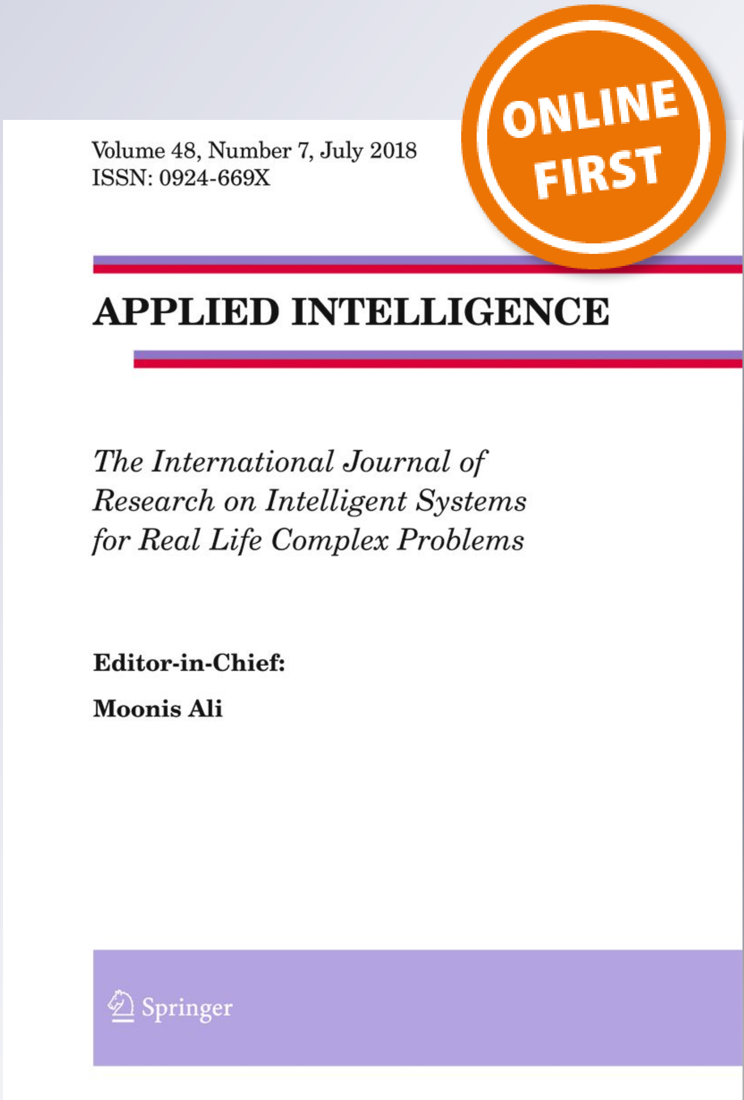
Incorporating reliable virtual ratings into social recommendation systems

Sajad Ahmadian, Majid Meghdadi & Mohsen Afsharchi

Applied Intelligence
The International Journal of Research on Intelligent Systems for Real Life Complex Problems

ISSN 0924-669X

Appl Intell
DOI 10.1007/s10489-018-1219-x



Springer

Applied Intelligence
https://doi.org/10.1007/s10489-018-1219-x

Incorporating reliable virtual ratings into social recommendation systems

Sajad Ahmadian[1] · Majid Meghdadi[1] · Mohsen Afsharchi[1]

Abstract

Social recommendation systems use social relations (such as trust, friendship, etc.) among users to find preferences and provide relevant suggestions to users. Historical ratings of items provided by the users are also used to predict unseen items in the systems. Therefore, it is an important issue to calculate the sufficient number of the historical ratings for each user to have a reliable prediction. In addition, providing a reliable mechanism to incorporate virtual ratings into the historical ratings of the users who have insufficient ratings can improve the performance of the rating prediction process. In this paper, a social recommendation system is proposed based on reliable virtual ratings to improve the accuracy of predicted ratings especially about the users with insufficient ratings. To this end, a probabilistic mechanism is used to calculate the minimum number of required ratings for each user to predict unseen items with high reliability. Then, a novel method is considered to predict the reliable virtual ratings based on users' reputation and clustering models. In addition, a noise detection method is used to detect noisy virtual ratings and prevent them from adding to the historical ratings. Then, the reliability, diversity and novelty of items are used to propose a selection mechanism for adding the remaining virtual ratings into historical ratings of the users with insufficient ratings. Therefore, the performance of the social recommendation systems can be improved through incorporating the reliable virtual ratings. Several experiments are performed based on three well-known datasets and the results show that the proposed method achieves higher performance than other state-of-the-art recommendation methods.

Keywords Social recommendation system · Virtual rating · Reputation · Noise detection · Reliability

1 Introduction

In recent years, the users who access to the Internet and also the amount of information on the web are exponentially increased. Therefore, processing and managing this information is a difficult and time-consuming process. Recommendation systems are powerful techniques which can help the users to find the information they need among a lot of choices by offering relevant suggestions to their preferences [1]. The recommendation problem can be formulated as a function $f : U \times I \rightarrow R$ which U is the set of all users, I is the set of all items, and R is the ratings provided by the users for the items. The main purpose of the recommendation systems is to predict the ratings of unseen items for the users by using their previous preferences. Users' preferences are generally represented as a matrix called user-item rating matrix. An example of the matrix with four users and five items is shown in Table 1. Recommendation systems attempt to predict the rating of item i_3 as an unseen item for user u_1 based on the available ratings which are provided by user u_1 for the other items. In this example, it is assumed that the ratings are in a specific range from 1 (bad) to 5 (excellent) with step 1.

Collaborative filtering (CF) is one of the most widely-used methods in recommendation systems which has made significant progress in exploiting user-item relations [2]. The main idea of the CF methods is that the preferences of a target user are likely to be similar to other users who have common preferences with the target user about items. These methods can be classified into two groups including memory-based and model-based methods. In memory-based CF methods, the historical ratings of the

✉ Majid Meghdadi
meghdadi@znu.ac.ir

Sajad Ahmadian
s.ahmadian@znu.ac.ir

Mohsen Afsharchi
afsharchim@znu.ac.ir

1 Department of Computer Engineering, University of Zanjan, Zanjan, Iran

Published online: 26 June 2018

Table 1 An example of user-item rating matrix

	i_1	i_2	i_3	i_4	i_5
u_1	2	1	?	4	3
u_2	-	2	3	1	-
u_3	-	-	1	4	5
u_4	4	3	5	-	1

users are directly used to calculate similarity values between them and find similar users as nearest neighbors for the target users. Then, the preferences of the nearest neighbors are used to predict ratings of unknown items for the target users [3–5]. It should be noted that, the nearest neighbors of the target users are typically the users whose preferences are most correlated to the preferences of the target users. On the other hand, model-based CF methods use historical ratings of the users to construct a model. Then, this constructed model can be used to predict unknown ratings for the target users [6]. Another type of recommendation methods is content-based technique which provides suggestions based on the items that are similar to the ones the target users have shown a preference for in the past rather than on the preferences of other users [7, 8]. The characteristics of the CF and content-based methods can be combined as hybrid methods to overcome certain limitations of these two types of recommendation methods [9, 10].

Social recommendation systems have been proposed based on social information among users which can be provided as different relations including friendship, trust, distrust, etc. The social relations are mainly classified into two types including explicit and implicit. The explicit relations are explicitly established by users which can be represented as a matrix to use in recommendation process [11–13]. An example of users' social relations is shown in Fig. 1 which it is assumed that the values of social relations can be established between 0 and 1. It should be noted that, the values of social relations are not symmetric in general [14]. Therefore, the value of relation between u_1 and u_2 may not be as the same value of relation between u_2 and u_1. The social relations in Fig. 1 can be represented as a user-user social relations matrix which is shown in Table 2. The implicit relations among the users can be calculated implicitly based on historical ratings of the users about existing items in the system. These implicit relations can improve the performance of the social recommendation systems for the condition that the users are not established explicit social relations [15, 16].

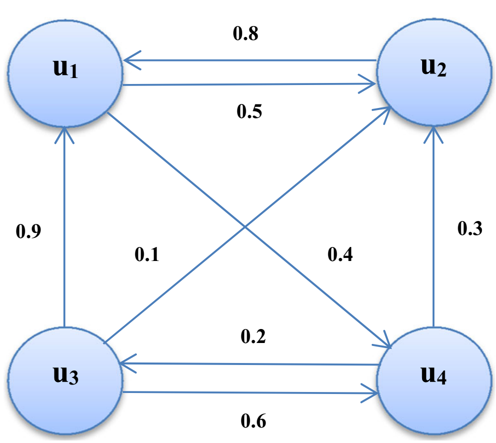


Fig. 1 An example of social relations among the users

CF-based recommendation methods suffer from several limitations and challenges due to traditional emphasis on calculating similarity values between the users. Therefore, these methods cannot provide suitable recommendations for the users who have insufficient ratings (i.e. cold start users) [17, 18]. In real applications, the users rate only a few number of items among a large number of choices available in the systems. The CF-based methods have a low performance about the cold start users because they cannot calculate the similarity values for this type of users, accurately. Social recommendation systems can improve the performance of the CF-based methods through incorporating social relations among the users into recommendation process. The main idea of these methods is to use additional information such as trust and friendship relations among the users to predict unknown ratings for the target users. In the systems, it is assumed that the preferences of the target users are likely to be similar to other users who have social relations with the target users [19–21]. However, the social relations among the users may not be enough to find nearest neighbors of the users. To this end, in addition to social relations, these systems use historical ratings assigned to items to provide suggestions for the users. Therefore, insufficient ratings for the users lead to reduce the performance of these systems in predicting unseen items. In other words, social recommendation systems cannot calculate reliable similarity values between the users who have insufficient ratings. Moreover, diversity and novelty of recommendations are important issues to increase the satisfaction of users in social recommendation systems.

Motivated by the above points and limitations, the main objective of this paper is to develop a novel social

Table 2 An example of user-user social relations matrix

	u_1	u_2	u_3	u_4
u_1	1	0.5	-	0.4
u_2	0.8	1	-	-
u_3	0.9	0.1	1	0.6
u_4	-	0.3	0.2	1

recommendation system based on reliable virtual ratings. To this end, a probabilistic method is used to calculate minimum number of required ratings for the users to have reliable predictions. Then, a novel mechanism is proposed to calculate virtual ratings and incorporate them into historical ratings of the users who have insufficient ratings. These virtual ratings are calculated based on users' reputation and clustering models. A noise detection mechanism is used in the proposed method to detect noisy virtual ratings and prevent them from adding to historical ratings of the users. Moreover, a selection mechanism is proposed to select a suitable subset of virtual ratings for the users which is based on the reliability, diversity and novelty of items. Therefore, the proposed method can improve the performance of social recommendation systems especially about the users with insufficient historical ratings.

The main contributions of the paper are summarized as follows:

- Evaluating the reliability of historical ratings provided by users in predicting unseen items is considered in this paper. To this end, a probabilistic model is applied to calculate the minimum number of required ratings for each user to have a reliable prediction.
- Improving the performance of the historical ratings is considered based on three important measures including reliability, diversity and novelty. For this purpose, a novel method is proposed to calculate virtual ratings based on reputation and clustering models. Moreover, a novel mechanism is proposed to select virtual ratings with considering reliability, diversity and novelty measures.
- Several experiments are performed to evaluate the performance of the proposed method in comparison with other state-of-the-art recommendation methods. The experiments results show that the proposed method significantly outperforms other methods based on different evaluation measures.

The rest of this paper is organized as follows. In Section 2, a brief description about the related works is presented. Section 3 presents the details of the proposed social recommendation method. Experimental results and performance comparisons between the proposed method and other recommendation methods are demonstrated in Section 4. Finally, Section 5 presents some important concluding remarks.

2 Related work

Several social recommendation systems have been proposed in the literature, which use social relations among users as additional information in recommendation process [15, 22–25]. In [23], a user model is proposed based on trust and distrust networks to identify trustworthy users and provide useful recommendations for cold start new users. An implicit social recommendation systems is proposed in [15] which is based on applying a new source of data to personalized recommendations by mining the short text posts of the users' friends. In [26], a graph-based method is proposed for social recommendations based on hyper edge and transitive closure. To this end, user-user and user-item connections are represented in the form of matrices to calculate the trust values between the users. Wu et al. [19] proposed a recommendation method for social media systems based on exploiting multi-sourced information to provide recommendations for the users. The main idea of the method is based on this assumption that the users' decisions on adopting item are affected both by their tastes and the favors of trusted friends. In [27], a social-based method is proposed for group recommendation systems in the tourism domain. For this purpose, a group profile is constructed by analyzing not only users' preferences, but also the social relations between members of a group. In [14], a recommendation approach is proposed which is based on collecting information from several social networking and social media platforms. A consolidated repository is formed based on the collected data that may become a valuable source for researchers. In [11], a fuzzy and argumentation based trust model is proposed which is also integrated within the practical reasoning of agents in the multi-agent recommendation systems.

Jamali et al. [28] proposed a model-based recommendation method in social networks by using matrix factorization techniques. To this end, the mechanism of trust propagation is incorporated into the model to provide more accurate recommendations for the users. In [25], a trust-based matrix factorization method is proposed to alleviate data sparsity and cold start problems in the recommendation systems. The main idea of the method is to use the explicit trust and also the implicit influence of both ratings and trust into recommendation process. In [29], a novel confidence-based recommendation method is proposed which uses four different confidence models with combining trust and certainty. These confidence models can derive the users' and items' confidence values from both of the local and global perspectives. A trust-based recommendation system is proposed in [30] for peer production services. In this method, the quality and veracity of peer production services are assessed by trust computing. Moreover, two fuzzy logic applications are used to support the decision of service choice. In [24], the authors proposed an ontology to describe trust relations among the users using fuzzy linguistic modeling. Then, these described trust relations are used to generate recommendations for the users. Ma et al. [31] introduced a factor analysis approach based on probabilistic matrix factorization to solve the data sparsity and poor prediction accuracy problems by employing both users' social network information and rating records. In [21], a model is proposed to provide personalized recommendations in social networks which is based on factors such as direct and indirect trust among the users, a mechanism for trust propagation, and user similarity.

Diversity and novelty are two important measures used in recommendation systems which can increase the satisfaction of users about provided suggestions. Therefore, several recommendation methods have been proposed in the literature to consider the diversity and novelty of recommendations [32–35]. In [36], a novel recommendation method is proposed which focuses on modeling user propensity toward selecting diverse items for users. To this end, the diversity measure is calculated using content-based item attributes. Moreover, an approach is proposed to re-arrange provided recommendation lists with the aim of fostering diversity in the final ranking. Gogna et al. [37] proposed a single stage optimization based solution to improve the diversity measure while maintaining acceptable levels of accuracy. For this purpose, additional diversity enhancing constraints are incorporated into matrix factorization based recommendation methods. A novel diversity-optimization method is proposed in [38] which is based on a time-sensitive semantic cover tree. To this end, a construction algorithm is defined for the proposed semantic tree model and then two supplement algorithms are considered to obtain a complete diversified recommendations list.

Hernando et al. [39] introduced the idea of incorporating a reliability measurement into the predictions made by collaborative filtering based recommendation systems. To this end, a general reliability measure is proposed which is suitable for any arbitrary recommendation system. In [40], a reliability measure is proposed which is based on the combination of trust relations among the users and also historical ratings of the users. This reliability measure is used to improve the performance of the trust networks in predicting unseen items through removing the users with low reliability among the selected nearest neighbors. In [41], the uncertainty of predictions is calculated which is based on two factors including posterior rating distribution and confidence level of predicted ratings to improve the accuracy of recommendations. A trust-based recommendation method is proposed in [42] which is based on merging the ratings of a user's trusted neighbors to complement and represent the preferences of the user. The quality of the merged ratings is evaluated using a confidence metric which is based on a positive and a negative factor. It is shown that, this method can improve the performance of social recommendation systems especially about cold start users. In [43], a probabilistic model is proposed to calculate the minimum number of required ratings for the products to produce reliable indicators on their qualities. Moreover, the effects of users' misbehavior are considered to evaluate the quality of a product and also the maximum fraction of misbehaving users that a rating suggestion rule can tolerate is derived in this method.

3 Proposed method

This section presents a novel social recommendation system based on reliable virtual ratings which is called SoRVR. The proposed method consists of five main steps. In the first step, the minimum number of ratings to predict unseen items with high reliability is calculated for each user based on a probabilistic method [43]. Then, in the second step a novel method is considered to calculate virtual ratings based on users' clustering and reputation models. In the third step, a noisy ratings detection mechanism [44] is used to detect noisy virtual ratings and prevent them from adding into historical ratings of users. In the fourth step, a selection mechanism is proposed to select a suitable subset of virtual ratings based on the reliability, diversity and novelty of items. Then, the selected virtual ratings can be added into historical ratings of users with insufficient number of ratings. Finally, the similarity values between the users can be calculated based on the improved historical ratings of users and then the ratings of unseen items can be predicted to suggest some relevant items as recommendation lists to users. The overall steps of the proposed method are shown in Fig. 2. In addition, the detailed discussions about the proposed method are presented in the following subsections.

3.1 Calculating minimum number of ratings

In this step, a probabilistic method is used to calculate the minimum number of required ratings for users to have predictions with high reliability. Suppose I is the set of items and U is the set of users in a recommender system. The ratings of items provided by the users can be selected from a set of $minR, \dots, maxR$ which $minR$ and $maxR$ are the minimum and maximum of ratings in the recommender system, respectively. Let I_u be a set of ratings provided by user u for the items and $n_{u,r}$ be the number of ratings for user u that are of rating level $r \in \{minR, \dots, maxR\}$. Therefore, user u can provide a rating r with probability $\alpha_{u,r}$ which is defined as follows:

$$\alpha_{u,r} = \Pr\left[I_{u,i} = r\right] = \frac{n_{u,r}}{|I_u|} \quad (1)$$

where, $u = 1, \dots, |U|$ and $i = 1, \dots, |I_u|$

Suppose l_u be the true rating provided by user u which can be calculated using the majority rule as $l_u = \{\alpha_{u,r}\}$. The main purpose of the method is to calculate the minimum

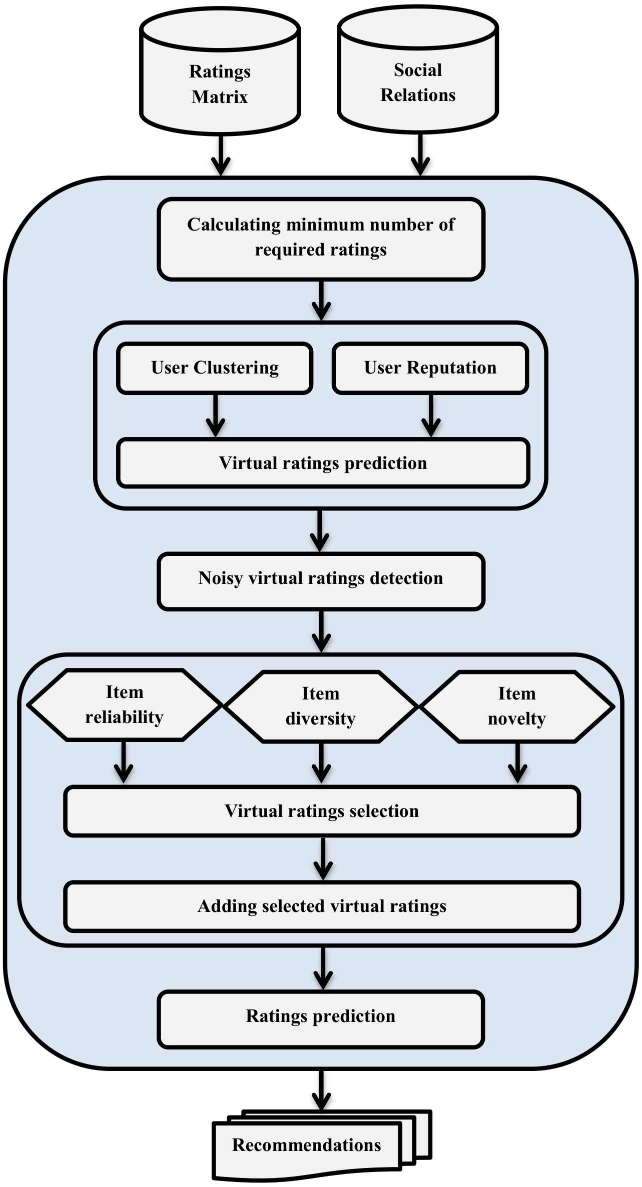


Fig. 2 Overview of the proposed method

number of ratings needed for user u so that the predicted rating $\hat{l}_u$ reveals the true rating l_u with high probability. Let $\tilde{\alpha}_u = \max\{\alpha_{u,r} | r \neq l_u\}$ be the second largest value of $\alpha_{u,r}$ and n'_u denotes the minimum number of ratings for user u to have a prediction with high reliability. The value of n'_u is calculated using the following equation:

$$n'_u = \left(2(\alpha_{u,l_u} + \tilde{\alpha}_u)(\alpha_{u,l_u} - \tilde{\alpha}_u)^{-2} - 2 + \frac{4(\alpha_{u,l_u} - \tilde{\alpha}_u)^{-1}}{3}\right) \times \ln(maxR - minR)\delta^{-1} \quad (2)$$

It is proved that if $|I_u| \geq n'_u$ then $\Pr[\hat{l}_u = l_u] \geq 1 - \delta$ [43]. In other words, if the number of ratings for user u (i.e. $|I_u|$) is at least equal to n'_u, then the predicted rating has the confidence $1-\delta$ in revealing the true rating. In the proposed method, we use this mechanism to determine the users who have insufficient ratings for predicting unseen items with high reliability. Moreover, an effective method is considered to improve the performance of rating prediction process by adding some virtual ratings into historical ratings of the users with insufficient ratings. Additional details about the next steps of the proposed method are presented in the following subsections.

3.2 Virtual rating prediction

The main idea of the proposed method is to improve the performance of social recommendation systems by incorporating reliable virtual ratings into rating prediction process. This mechanism can improve the accuracy of predicted ratings especially about the users who have insufficient historical ratings for predicting unseen items with high reliability. The virtual ratings are calculated using two user models including user clustering and user reputation. The overall steps of calculating the virtual ratings are presented in the following subsections.

3.2.1 User clustering

In this section, a graph clustering method is used to group users into appropriate clusters. To this end, the users' set are mapped into a graph $G = (V, E, W)$ which V is the set of users, E indicates the set of edges between the user pairs, and W denotes similarity weights between each pair of the users. The historical ratings of users and their trust relations are used to calculate the similarity weights between them. Therefore, Pearson correlation coefficient function is used to calculate the similarity values between the users based on their historical ratings as follows:

$$sim(u,v) = \frac{\sum_{i \in A_{u,v}} (r_i(u) - \bar{r}(u))(r_i(v) - \bar{r}(v))}{\sqrt{\sum_{i \in A_{u,v}} (r_i(u) - \bar{r}(u))^2}\sqrt{\sum_{i \in A_{u,v}} (r_i(v) - \bar{r}(v))^2}} \quad (3)$$

where, $r_i(u)$ is the rating of item i provided by user u, $\bar{r}(u)$ denotes the average of ratings provided by user u, and $A_{u,v}$ indicates common items which are rated by both of the users u and v.

On the other hand, the trust relations between the users are used as social information to calculate the similarity weights. These trust values can be calculated using (4) as follows:

$$T_{u,v} = \frac{d_{max} - d_{u,v} + 1}{d_{max}} \quad (4)$$

where, $d_{u,v}$ denotes the trust propagation distance between the users u and v [12], and d_{max} is the maximum allowable propagation distance between the users which can be calculated as follows:

$$d_{max} = \frac{\ln(n)}{\ln(k)} \tag{5}$$

where, n and k are the size and the average degree of the trust networks in a social recommendation system, respectively [45]

Finally, a combination of the similarity and trust values between the users is used as the final similarity weights W in the graph G. These similarity weights are calculated using (6) as follows:

$$w_{u,v} = \begin{cases} \frac{2\times sim(u,v)\times T_{u,v}}{sim(u,v)+T_{u,v}} & if\ sim(u,v)>0\ and\ T_{u,v}>0 \\ T_{u,v} & else\ if\ sim(u,v)\leq 0\ and\ T_{u,v}>0 \\ sim(u,v) & else\ if\ sim(u,v)>0\ and\ T_{u,v}\leq 0 \\ 0 & else \end{cases} \tag{6}$$

where, $sim(u,v)$ and $T_{u,v}$ are the similarity value and the trust value between the users u and v which can be calculated using (3) and (4), respectively.

After mapping the users' set into the graph G, the user clustering algorithm can be applied on the graph to find appropriate clusters of the users. This algorithm consists of two main steps which are shown in Fig. 3 (i.e. Algorithm 1). In the first step, a graph-based approach is used to find the sparsest subgraph for using as the initial centers set of the clusters [46]. To this end, the density of graph G (i.e. $\rho(G)$) is calculated. The density of a subgraph $S \subseteq V$ can be calculated as follows:

$$\rho(S) = \frac{\sum_{e\in E(S)} w_e}{|S|} \tag{7}$$

where, $E(S)$ is the edges set of subgraph S and w_e denotes the weight of edge e. Then, the candidate nodes $\tilde{A}(S)$ which can be removed from the graph are identified based on their weighted degrees using a threshold value. A portion of $\frac{\varepsilon}{1+\varepsilon} \times |\tilde{A}(S)|$ selected nodes with highest weighted degrees is removed from the candidate list. The weighted degree of node $i \in S$ is calculated using (8):

$$wd_S(i) = \sum_{e_{ij}\in E(S)} w_{e_{ij}} \tag{8}$$

where, e_{ij} denotes the edge between nodes i and j, and $w_{e_{ij}}$ is the weight of edge e_{ij}. The algorithm proceeds on the remaining graph if the resulted subgraph is non-empty. It should be noted that, the algorithm guarantees that the final subgraph contains at least k nodes. The main purpose of this step is to find a subgraph with minimum density which leads to form the centers set with maximum distances between them and it can be used as initial centers set of the clustering algorithm.

In the second step of the clustering algorithm, an iterative process is applied on the initial centers set to find final centers set for the clusters. To this end, each user is assigned to the nearest cluster center based on the initial centers set (i.e. $\tilde{S}$) which is formed by the first step of the clustering algorithm. Then, the new centers of the clusters are determined using an iterative process based on lines 2.5-2.6 of algorithm 1. In addition, those of the clusters whose associated members are less than a threshold value (i.e. m) will be merged with the other clusters to form clusters with higher performance. This merging process is necessary because the clusters with a small number of users may lead to reduction of rating prediction accuracy. Finally, the resulted clusters are used as final clusters of the users.

3.2.2 User reputation

In this section, a new model is proposed to calculate reputation values of the users. Two types of information including historical ratings and social relations of the users are used to calculate this reputation model. The proposed user reputation model is used to measure the influences of the users in predicting virtual ratings. In other words, the users who have high reputation values can affect more than the users with lower reputation values on predicting the virtual ratings. To this end, the well-known PageRank algorithm [47] is used to calculate the user reputation model based on social relations (i.e. trust statements). The main idea of this method is that the users who trusted by a large number of trusted users have a higher value of the reputation. A recursive function is used to calculate the trust-based reputation value for user u as follows:

$$TR_u = \omega\frac{1}{|U|} + (1-\omega)\sum_{T_{v,u}\neq 0}\frac{TR_v}{\deg(v)} \tag{9}$$

where, ω is a constant value which is set to $\omega = 0.15$ as suggested by [47], $|U|$ denotes the number of all users in the system, TR_v is the trust-based reputation value for user v, $T_{v,u}$ is the trust value between the users v and u which can be calculated using (4), and $\deg(v)$ indicates the out degree of user v in his/her trust network. It should be noted that, a recursive function is presented in (9) because the reputation value of each user depends on the reputation values of his/her trusted users. It is proved that this approach will converge to a unique stationary distribution without depending on the choice of initialized vector [48]. Therefore, the initial vector of the reputation values is randomly initialized by a set of non-negative values. In addition, the following equation is used as termination condition of the recursive function:

$$TR^{(n)} - TR^{(n-1)} = 0 \tag{10}$$

where, $TR^{(n)}$ and $TR^{(n-1)}$ are the trust-based reputation vectors of the users in the iterations n and $n-1$, respectively.

Algorithm 1. User clustering algorithm.

Inputs: $G = (V, E, W)$, $k > 0$, $\varepsilon > 0$, and $m > 0$.
Output: Users' clusters.
Begin algorithm:
1: **First step: Finding initial centers set.**
1.1: ***Set*** $S = V$ and $\tilde{S} = V$;
1.2: **if** $S \neq \emptyset$ then **go to** step 1.3 **else go to** step 2.1;
1.3: ***Calculate*** the density of S (i.e. $\rho(S)$) using Eq. (7);
1.4: ***Set*** $\tilde{A}(S) = \emptyset$;
1.5: **for** all $i \in S$ **do**
1.6: ***Calculate*** the weighted degree of node i (i.e. $wd_S(i)$) using Eq. (8);
1.7: **if** $wd_S(i) \geq (2 + 2\varepsilon) * \rho(S)$ **then**
1.8: $\tilde{A}(S) = \tilde{A}(S) \cup \{i\}$;
1.9: **end if**
1.10: **end for**
1.11: ***Sort*** all $i \in \tilde{A}(S)$ descending based on their $wd_S(i)$;
1.12: ***Set*** $r = \frac{\varepsilon}{1+\varepsilon} \times |\tilde{A}(S)|$;
1.13: ***Select*** top_r nodes from $\tilde{A}(S)$ as $A(S)$;
1.14: ***Set*** $S = S - A(S)$;
1.15: **if** $|S| \geq k$ and $\rho(S) < \rho(\tilde{S})$ **then**
1.16: $\tilde{S} = S$;
1.17: **end if**
1.18: ***Go*** **to** step 1.2;
2: **Second step: Finding final clusters.**
2.1: ***Set*** $k' = |\tilde{S}|$;
2.2: ***Set*** $p_j = \tilde{S}_j, \forall j = 1, \ldots, k'$;
2.3: ***Let*** $p_j, \forall j = 1, \ldots, k'$ be initial center corresponding to **j**-th cluster C_j;
2.4: ***Associate*** each non-selected user to nearest cluster;
2.5: ***Select*** new centers $p'_j = \arg\max_{v_i \in C_j} sum(v_i), j = 1, \ldots, k'$, where $sum(v_i) = \sum_{v_t \in C_j, v_t \neq v_i} w(v_i, v_t)$;
2.6: **if** $p_j = p'_j, \forall j = 1, \ldots, k'$ **then go to** line 2.7**, else** $p_j = p'_j, \forall j = 1, \ldots, k'$ and **go to** line 2.4;
2.7: **for** all $C_j, j = 1, \ldots, k'$ **do**
2.8: **if** $|C_j| < m$ **then**
2.9: ***Merge*** the members of C_j to other clusters;
2.10: **end if**
2.11: **end for**
2.12: **for** all users $u \in V$ **do**
2.13: ***Let*** C_u be the cluster that user u belong to;
2.14: **end for**
End algorithm.

Fig. 3 Pseudo code of the user clustering algorithm

On the other hand, a method described as user reputation in [49] is used to calculate the user reputation model based on historical ratings of the users. The main idea of this method is to calculate the correlation coefficient between the historical ratings of users and quality vector of items. This rating-based reputation model consists of four steps which are summarized as follows:

Step 1: The initial value for the rating-based reputation of user u can be calculated using (11) as follows:

$$RR_u = \frac{|I_u|}{|I|} \tag{11}$$

where, RR_u denotes the rating-based reputation value for user u, $|I_u|$ is the number of ratings provided by user u and $|I|$ is the number of all items in the system.

Step 2: The quality of item i can be calculated using (12) as follows:

$$Q_i = \frac{\sum_{u \in U_i} RR_u r_i(u)}{\sum_{u \in U_i} RR_u} \tag{12}$$

where, U_i is the set of users who rated item i, and $r_i(u)$ is the rating of item i provided by user u

Step 3: The Pearson coefficient function is used to calculate correlation value between user u and quality vector of items as follows:

$$C_u = \frac{\sum_{i \in I_u} (r_i(u) - \bar{r}(u))(Q_i - \bar{Q}_u)}{\sqrt{\sum_{i \in I_u} (r_i(u) - \bar{r}(u))^2} \sqrt{(Q_i - \bar{Q}_u)^2}} \tag{13}$$

where, $\bar{r}(u)$ is the average of ratings provided by user u, I_u is a set of items that have ratings given by user u, and $\bar{Q}_u$ is the average value of qualities for all items rated by user u. The range of rating-based reputation value for user u is bounded into [0,1] using the following equation:

$$RR_u = \frac{C_u + 1}{2} \tag{14}$$

Step 4: The algorithm is iterated using steps 2 and 3 until the results satisfy the following termination condition:

$$\frac{1}{|I|} \sum_{i=1}^{|I|} |Q_i^{(n)} - Q_i^{(n-1)}| \leq \varepsilon \tag{15}$$

where, $|I|$ is the number of all items in the system, $Q_i^{(n)}$ is the quality of item i in iteration n, and ε is a constant value which is set to $\varepsilon = 10^{-6}$ as suggested by [49].

Finally, a novel user reputation model is proposed which is based on combination of the rating-based and trust-based reputation models. To this end, we use the harmonic mean of the rating-based and trust-based reputations to combine these models into a final user reputation model as follows:

$$CR_u = \frac{2 \times TR_u \times RR_u}{TR_u + RR_u} \tag{16}$$

where, CR_u is the proposed reputation model for user u, TR_u and RR_u are the trust-based and rating-based reputation values for user u which can be calculated using (9) and (14), respectively.

3.2.3 Prediction

In this section, the calculated user clustering and reputation models are used to predict some virtual ratings for users. To this end, the user clustering model is used to select some items for a specific user u which are rated by the users who exist in cluster of user u. Let I_u be the set of items which are rated by user u and I_{C_u} be the set of items which are rated by the users who exist in cluster of user u. The main idea of this mechanism is to select relevant items for each user to calculate virtual ratings based on the preferences of neighbors in his/her cluster. It should be noted that, the historical ratings of users (i.e. I_u) are used without any changes as real ratings in predicting process. Therefore, the set of items which is selected to calculate their virtual ratings is $I'_u = I_{C_u} - I_u$. The user reputation model is used to calculate the virtual ratings for determining the effect of each user on predicting virtual ratings. The virtual rating of item $i \in I'_u$ for user u can be calculated as follows:

$$VR_i(u) = \frac{\sum_{v \in C_{u_i}, v \neq u} CR_v r_i(v)}{\sum_{v \in C_{u_i}, v \neq u} CR_v} \tag{17}$$

where, C_{u_i} is the set of users in cluster of user u that have a rating for item i, CR_v denotes the user reputation value for user v which is calculated using (16), and $r_i(v)$ is the rating of item i provided by user v

3.3 Noisy virtual rating detection

In this step, a noise detection method is used to detect noisy virtual ratings and prevent them from adding into historical ratings of users. This step is necessary for the proposed method because incorporating noisy virtual ratings into historical ratings of users leads to reduce the performance of rating prediction process. To this end, each rating $r_i(u)$ for user u and item i is classified according its value into three classes including $Weak$ if $r_i(u) < LB$, $Average$ if $LB \leq r_i(u) < UB$, and $Strong$ if $r_i(u) \geq UB$ which LB and UB are constant values for the lower bound and upper bound of ratings, respectively. These constant values can be calculated as follows [44]:

$$LB = minR + round(\frac{1}{3} \times (maxR - minR)) \tag{18}$$

$$UB = maxR - round(\frac{1}{3} \times (maxR - minR)) \tag{19}$$

where, $minR$ and $maxR$ are the minimum and maximum of ratings in the recommender system, respectively.

Let W_u, A_u, and S_u be the sets of ratings provided by user u with classes of $Weak$, $Average$, and $Strong$, respectively. In addition, suppose W_i, A_i, and S_i are the sets of ratings assigned to item i with classes of $Weak$, $Average$, and $Strong$, respectively. Therefore, a user u can be classified into three classes as follows:

- **Critical user:** if $|W_u| \geq |A_u| + |S_u|$
- **Average user:** if $|A_u| \geq |W_u| + |S_u|$
- **Benevolent user:** if $|S_u| \geq |W_u| + |A_u|$

Moreover, an item i can be classified into three classes as follows:

- **Weakly-preferred:** if $|W_i| \geq |A_i| + |S_i|$
- **Averagely-preferred:** if $|A_i| \geq |W_i| + |S_i|$
- **Strongly-preferred:** if $|S_i| \geq |W_i| + |A_i|$

Finally, a noise detection mechanism is used which is based on the above classifications. A virtual rating $VR_i(u)$ for user u and item i which is calculated using (17) can be detected as a noisy rating if one of the following conditions is satisfied:

- **Condition 1:** if u is critical, i is weakly-preferred, and $VR_i(u) \geq LB$
- **Condition 2:** if u is average, i is averagely-preferred, and $(VR_i(u) < LB or VR_i(u) \geq UB)$
- **Condition 3:** if u is benevolent, i is strongly-preferred, and $VR_i(u) < UB$

The detected noisy virtual ratings are removed from the set of calculated virtual ratings for user u (i.e. I'_u) in (17).

3.4 Virtual rating selection

In this section, a novel selection mechanism is proposed to select an appropriate subset of virtual ratings to add into historical ratings of users. To this end, three different measures are considered including items' reliability, diversity and novelty. The main purpose of this mechanism is to improve the reliability of virtual ratings and also enhance the performance of recommendations to users through the diversity and novelty measures. It should be noted that, a subset of virtual ratings are determined by the used noise detection method in Section 3.3. Therefore, the proposed selection mechanism is applied on this subset of virtual ratings to select final subset of them for adding into historical ratings of users.

The reliability measure for item i is based on three different factors as follows:

Factor 1 This factor is based on the number of ratings which have been assigned to item i. The more ratings for item i indicate the higher value of reliability. Therefore, this factor has a positive effect on the reliability value of item i and is calculated using (20) as follows:

$$f_i(I_i) = 1 - \frac{\bar{i}}{\bar{i} + |I_i|} \tag{20}$$

where, $|I_i|$ denotes the number of ratings which have been assigned to item i, and $\bar{i}$ is the median of the values for $|I_i|$.

Factor 2 The standard deviation of ratings that have been assigned to item i is the second factor to calculate the item reliability measure. A higher value of this factor makes a lower value for reliability of item i. Therefore, this factor has a negative effect on the item reliability measure and can be calculated using a decreasing function as follows:

$$f_{sd}(stdev(I_i)) = \frac{max - stdev(I_i)}{max - min} \tag{21}$$

where, $stdev(I_i)$ denotes the standard deviation of ratings that have been assigned to item i, max and min are the maximum and minimum of values for $stdev(I_i)$, respectively.

Factor 3 The summation of user reputation values for the users who have assigned a rating to item i is the third factor to calculate item reliability measure. This factor has a positive effect on the item reliability measure. In other words, the higher values of this factor lead to increase the values of item reliability measure. Therefore, this positive factor can be calculated using (22) as follows:

$$f_c(C_i) = 1 - \frac{\bar{c}}{\bar{c} + C_i} \tag{22}$$

where,

$$C_i = \sum\nolimits_{u \in U_i} CR_u \tag{23}$$

where, U_i denotes the set of users who have assigned a rating to item i, CR_u is the user reputation value for user u which can be calculated using (16), and $\bar{c}$ is the median of values for C_i.

After calculating the mentioned factors, the final value of item reliability measure can be calculated as the geometric average of these factors. The higher values of $f_i(I_i)$ and $f_c(C_i)$ as positive factors lead to increase the final value of reliability measure. In addition, the higher values of $f_{sd}(stdev(I_i))$ as a negative factor created lower values for the item reliability. It should be noted that, the values of positive factors in (20) and (22) are calculated based on the median concept to bound these values into the range of [0,1]. The reliability measure for item i can be defined based on the geometric average of three calculated factors as follows:

$$IR_i = [f_i(I_i).f_{sd}(stdev(I_i))^{f_i(I_i)}.f_c(C_i)^{f_i(I_i)}]^{\frac{1}{1+2f_i(I_i)}} \tag{24}$$

It should be noted that, the weight of first factor (i.e. $f_i(I_i)$) is determined as a constant value of 1 in (24) because this factor is not dependent on any other factors. On the other hand, the weights of $stdev(I_i)$ and $f_c(C_i)$ are determined as the value of $f_i(I_i)$ in (24). Because, the values of $stdev(I_i)$ and C_i are dependent on the value of $|I_i|$. In other words, the higher values of $|I_i|$ lead to increase the values of $stdev(I_i)$ and C_i, and vice versa.

The second measure which is used in the proposed selection mechanism is diversity measure of virtual ratings. The diversity measure is defined based on the internal differences within the set of items recommended to each user. Therefore, the aim of using this measure is to improve the diversity of recommendation lists provided for users by the recommender system. To this end, the intra-list diversity measure is used to calculate diversity of each item using (25) as follows:

$$D_i = \frac{1}{|I_a|} \sum\nolimits_{j \in I_a} (1 - s(i, j)) \tag{25}$$

where, D_i is the diversity value of item i for active user a, I_a denotes the set of items that have been rated by active user a, and $s(i, j)$ is the similarity value between items i and j which can be calculated using the cosine similarity function as follows:

$$s(i, j) = \frac{\sum_{u \in U} r_i(u).r_j(u)}{\sqrt{\sum_{u \in U} r_i(u)^2 \sum_{u \in U} r_j(u)^2}} \tag{26}$$

where, U denotes the set of all users in the system, and $r_i(u)$ is the rating of item i provided by user u.

The third measure which is used in the proposed selection mechanism is the novelty measure of virtual ratings that indicates the degree of difference between the items recommended to and known by the user. To this end, the inverse user frequency (IUF) measure [50] is used to calculate the novelty value of each item as follows:

$$N_i = -P(i) \log_2 P(i) \tag{27}$$

where, N_i denotes the novelty value of item i, and $P(i)$ is the probability of item i being drawn from the recommendation lists which can be calculated using Eq. (28) as follows:

$$P(i) = \frac{|I_i|}{|U|} \tag{28}$$

where, $|I_i|$ denotes the number of ratings provided for item i, and $|U|$ is the number of all users in the system.

Finally, a combination of the items' reliability, diversity and novelty measures is calculated as the final measure of the proposed selection mechanism. To this end, the final measure of item i is calculated using the harmonic mean of the item's reliability, diversity and novelty measures as follows:

$$F_i = \frac{3}{IR_i^{-1} + D_i^{-1} + N_i^{-1}} \tag{29}$$

where, IR_i, D_i, and N_i are the reliability, diversity, and novelty measures for item i which can be calculated using (24), (25) and (27), respectively.

The final measure of items (i.e. F_i) is used to select a subset of virtual ratings for adding into historical ratings of users. It should be noted that, the minimum number of required ratings for user u (i.e. n'_u) to have reliable predictions is calculated using (2). On the other hand, let I_u be the set of items which are rated by user u. Therefore, the number of virtual ratings which can be added into historical ratings of user u is calculated as $n'_u - |I_u|$. To this end, the virtual ratings are sorted in descending order based on their final measures (i.e. F_i). Then, a specified number of virtual ratings (i.e. $n'_u - |I_u|$) from the beginning of the sorted list is selected to add into historical ratings of user u. It should be noted that, the added virtual ratings for user u are completely different from her/his historical ratings. In other words, the real ratings of users in their historical ratings are not changed or replaced with virtual ratings.

3.5 Recommendation

In this step, the final similarity weights between users can be calculated using (6) based on the improved historical ratings of users. Then, a set of nearest neighbors for active user a is calculated as follows:

$$K_a = \{u \in U | w_{a,u} \geq \theta\} \tag{30}$$

where, U is the set of all users in the system, $w_{a,u}$ denotes the similarity weight between users a and u which is calculated using (6), and θ is a threshold value for the similarity weights. In addition, the rating of an unseen item i for active user a is calculated using (31) as follows:

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u \in K_{a,i}} w_{a,u}(r_i(u) - \bar{r}_u)}{\sum_{u \in K_{a,i}} w_{a,u}} \tag{31}$$

where, $\bar{r}_a$ denotes the average of ratings for user a, $K_{a,i}$ is the set of neighbors for user a who have a rating for item i, $r_i(u)$ is the rating of item i provided by user u, and $w_{a,u}$ is the similarity weight between users a and u which is calculated using (6). Finally, a subset of items with higher ratings is selected for recommending to active user as a recommendation list. The pseudo code of the proposed method is represented in Fig. 4 (i.e. Algorithm 2).

4 Experiments

In this section, the performance of the proposed method (i.e. SoRVR) is evaluated based on three well-known datasets including Epinions,[1] Flixster,[2] and FilmTrust.[3] To this end, the proposed method is compared with several recommendation methods including User-based CF (UCF), Item-based CF (ICF) [51], SlopeOne [52], SVD++ [53], TARS [12], SoRec [31], SoReg [54], TrustMF [22], SocialMF [28], TrustSVD [25], IPG [26], CETrust [21], and 2DGC [55]. In the proposed method, some parameters need to be initialized for performing the experiments. The parameter δ is used in (2) which is set to $\delta = 0.4$ for the FilmTrust dataset and $\delta = 0.3$ for the Epinions and Flixster datasets. Moreover, the parameter θ is a threshold value in (30) to calculate nearest neighbors of users which is set to $\theta = 0.6$ for the Epinions, Flixster, and FilmTrust datasets. In the user clustering method (i.e. Algorithm 1), three parameters are used including k, ε, and m. The parameters k and ε are respectively set to $k = 5$ and $\varepsilon = 1$ for all of the used datasets as

[1] http://www.trustlet.org/datasets/download_epinions.

[2] http://www.cs.sfu.ca/~sja25/personal/datasets/

[3] http://trust.mindswap.org/FilmTrust.

Algorithm 2. Social recommendation system based on reliable virtual ratings (SoRVR)

Input: Parameters $\boldsymbol{\delta}$ and $\boldsymbol{\theta}$.
Output: Predicted ratings for users.
Begin algorithm:

Let $\boldsymbol{U}$ be the set of all users;
Let $\boldsymbol{I}$ be the set of all items;
for all $\boldsymbol{u} \in \boldsymbol{U}$ **do**
    Calculate the minimum number of required ratings for user $\boldsymbol{u}$ (i.e. $\boldsymbol{n'_u}$) using Eq. (2);
end for
Apply Algorithm 1 to cluster users into appropriate clusters;
for all $\boldsymbol{u} \in \boldsymbol{U}$ **do**
    Calculate the user reputation value for user $\boldsymbol{u}$ (i.e. $\boldsymbol{CR_u}$) using Eq. (16);
end for
for all $\boldsymbol{u} \in \boldsymbol{U}$ **do**
    Let $\boldsymbol{I_u} \subseteq \boldsymbol{I}$ be the set of items which are rated by user $\boldsymbol{u}$;
    if $(|\boldsymbol{I_u}| < \boldsymbol{n'_u})$ **then**
        Let $\boldsymbol{I_{C_u}} \subseteq \boldsymbol{I}$ be the set of items that are rated by the users who exist in cluster of user $\boldsymbol{u}$;
        Set $\boldsymbol{I'_u} = \boldsymbol{I_{C_u}} - \boldsymbol{I_u}$;
        for all $\boldsymbol{i} \in \boldsymbol{I'_u}$ **do**
            Calculate the virtual rating of item $\boldsymbol{i}$ (i.e. $\boldsymbol{VR_i(u)}$) using Eq. (17);
            Apply the noise detection method for $\boldsymbol{VR_i(u)}$ based on Section 3.3;
            if $\boldsymbol{VR_i(u)}$ is not a noisy virtual rating **then**
                Add $\boldsymbol{VR_i(u)}$ into the set of the virtual ratings of user $\boldsymbol{u}$ (i.e. $\boldsymbol{VR(u)}$);
            end if
        end for
        for all $\boldsymbol{i} \in \boldsymbol{VR(u)}$ **do**
            Calculate the final measure $\boldsymbol{F_i}$ for $\boldsymbol{VR_i(u)}$ using Eq. (29);
        end for
        Sort $\boldsymbol{VR(u)}$ descending based on their final measures $\boldsymbol{F_i}$;
        Set $\boldsymbol{s} = \boldsymbol{n'_u} - |\boldsymbol{I_u}|$;
        Select $\boldsymbol{top\_s}$ virtual ratings from $\boldsymbol{VR(u)}$ as $\boldsymbol{\tilde{I}_u}$;
        Add $\boldsymbol{\tilde{I}_u}$ into historical ratings of user $\boldsymbol{u}$;
    end if
end for
for all $\boldsymbol{a} \in \boldsymbol{U}$ **do**
    Calculate the similarity weights between the active user $\boldsymbol{a}$ and other users using Eq. (6);
    Calculate the nearest neighbors set of the active user $\boldsymbol{a}$ using Eq. (30);
end for
Predict the unseen items for the active users using Eq. (31);

End algorithm.

Fig. 4 Pseudo code of the proposed method

default values which give acceptable results. In addition, the parameter m is set to $m = 40$ for the Epinions dataset and $m = 30$ for the Flixster and FilmTrust datasets. In all of the experiments, the 5-fold cross validation method is used to evaluate the recommendation methods. Finally, the parameters of the other recommendation methods are set based on the optimal values which are reported in their corresponding papers to make a fair comparison with the proposed method.

4.1 Datasets

In the experiments, the Epinions, Flixster, and FilmTrust datasets are used to compare the proposed method with other recommendation methods. In the Epinions dataset, the opinions of users about existing items are used as numerical ratings in the range of 1 (min) to 5 (max). In addition, the Epinions dataset includes 49,290 users who rated at least once among 139,738 items. The trust relations among the users are used as social information in the Epinions dataset which the values of them are 0 or 1. On the other hand, the Flixster and FilmTrust datasets contain the ratings of users about existing items in the range of 0.5 (min) to 4.0 (max) with step 0.5. Moreover, the friend relationships and link information among the users are used as social information in the Flixster and FilmTrust datasets, respectively. There are 1,986 users, 2,071 items, and 35,497 ratings in the

Table 3 The statistics of the evaluation datasets

Dataset	#Users	#Items	#Ratings	#Trust	Sparsity (%)
Epinions	10K	117K	385K	288K	99.97
Flixster	10K	6K	55K	89K	99.92
FilmTrust	1986	2071	35,497	1853	99.14

FilmTrust dataset. Moreover, two subsets of the Epinions and Flixster datasets are randomly sampled for simplicity by selecting 10K users with their corresponding ratings and trust statements. The statistics of the evaluation datasets is presented in Table 3.

4.2 Evaluation measures

To compare the proposed method with other recommendation methods, several evaluation measures are used in the experiments including mean absolute error (MAE), diversity, novelty, precision, recall, and F1 measures. The MAE measure is used to calculate the accuracy of the recommendation method in predicting unseen items and can be defined as follows:

$$MAE = \frac{\sum_{i=1}^{n} |r_i - p_i|}{n} \tag{32}$$

where, r_i and p_i are respectively the real and predicted ratings of item i, and n is the total number of ratings in the test set. The lower value of the MAE measure shows a better performance for the recommendation methods.

Another measure which is used to evaluate the recommendation methods is the diversity measure. This measure refers to the differences between items which are suggested to a target user as recommendations list. To this end, the intra-list diversity is used to calculate the diversity measure as (33) [34]:

$$Diversity = \frac{\sum_{u=1}^{|U|} diversity_u}{|U|} \tag{33}$$

where,

$$diversity_u = \frac{1}{|L_u|(|L_u|-1)} \sum_{i \in L_u} \sum_{j \in L_u, j \neq i} [1 - s(i,j)] \tag{34}$$

and U denotes the set of all users in the system, L_u is the recommendations list for user u, and $s(i, j)$ is the cosine similarity between items i and j which can be calculated using (26). The novelty measure of a target item refers to the difference between it and other items which are previously experienced by the active user. To this end, the novelty measure can be calculated based on Shannon entropy [56] as follows:

$$Novelty = -\sum_{i \in I} p(i|s) \log_2 p(i|s) \tag{35}$$

where, $p(i|s)$ refers to the probability of item i being drawn from the recommendation lists of users generated by system s. Therefore, (36) can be used to calculate this probability value as follows:

$$p(i|s) = \frac{|\{u \in U | i \in L_u\}|}{\sum_{j \in I} |\{u \in U | j \in L_u\}|} \tag{36}$$

where, U and I are respectively the set of users and items in the system, and L_u denotes the recommendations list for user u.

Furthermore, the precision and recall measures can be used as exactness and completeness of the recommendation methods. To this end, the recommendation methods are trained based on training set and then a set of recommendations (i.e. L) is provided for each user based on predicted ratings of test set. Also, the items which appear in both of the recommendations and test sets are members of a special set called the hit set [57]. Therefore, the precision measure can be defined as the ratio of the hit set size to the recommendations set size and the recall measure can be defined as the ratio of hit set size to the test set size. The following equations can be used to calculate the precision and recall measures:

$$Precision = \frac{|test \cap L|}{|L|} \tag{37}$$

$$Recall = \frac{|test \cap L|}{|test|} \tag{38}$$

Finally, the harmonic mean of the precision and recall measures can be calculated as F1 measure using (39):

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{39}$$

4.3 Results

In this section, the results of experiments are reported based on the used datasets and calculated evaluation measures. To this end, two different views of data including all users and cold start users (i.e. the users with less than five ratings) are used to compare the proposed method with the other recommendation methods. Table 4 shows the results of experiments based on the MAE measure for all of the Epinions, Flixster, and FilmTrust datasets. The results indicate that the proposed method obtains the best MAE value for both of the all users and cold start users' views based on the Epinions dataset. The MAE values for the proposed method are 0.561 and 0.605 for the all users and cold start users, respectively. The IPG method obtains the second best MAE value for the all users which its value is 0.781. Moreover, the second best result for MAE measure is obtained by the TrustSVD method for the cold start users which its value is 0.825. Therefore, it can be concluded that the proposed method significantly outperforms other

Table 4 Experiment results on the *Epinions, Flixster*, and *FilmTrust datasets* for MAE measure

Algorithms	All users			Cold users		
	Epinions	Flixster	FilmTrust	Epinions	Flixster	FilmTrust
UCF	0.865	0.956	0.703	1.063	1.213	0.744
ICF	0.824	0.912	0.698	1.102	1.268	0.786
SlopeOne	0.815	0.908	0.632	0.894	0.942	0.676
SVD+ +	0.807	0.781	0.611	0.881	0.846	0.677
TARS	0.826	0.873	0.763	0.864	0.952	0.827
SoRec	0.874	0.735	0.628	0.832	0.863	0.670
SoReg	0.926	0.762	0.668	1.104	0.937	0.771
TrustMF	0.804	0.864	0.631	0.835	0.891	0.674
SocialMF	0.813	0.756	0.638	0.846	0.877	0.680
TrustSVD	0.786	0.719	0.607	0.825	0.823	0.661
IPG	0.781	0.735	0.684	0.839	0.862	0.726
CETrust	0.812	0.657	0.626	0.851	0.796	0.653
2DGC	0.796	0.725	0.659	0.838	0.864	**0.288**
SoRVR	**0.561**	**0.586**	**0.513**	**0.605**	**0.728**	0.594

The best results are presented in boldface

methods based on the MAE measure for the all users and cold start users. Moreover, the performance of the proposed method is better than other methods based on the MAE measure for the Flixster dataset. For example, the MAE value of the proposed method is 0.586 for the all users while the MAE value of the second best method (i.e. CETrust) is 0.657. The results demonstrate that the proposed method has the best and second best results based on the FilmTrust dataset for the all users and cold start users, respectively. The 2DGC method obtains the best MAE value for the cold start users based on the FilmTrust dataset. The MAE value of the proposed method is 0.594 for the cold start users while the MAE value of the 2DGC method is 0.288. On the other hand, the MAE values for the proposed method and the 2DGC method based on the all users are 0.513 and 0.659, respectively.

Moreover, several experiments are performed to compare the proposed method with the other recommendation methods based on the diversity and novelty measures. Tables 5-7 report the results of performed experiments based on the diversity and novelty measures for all users view and also different lengths of recommendation lists (i.e. $L =$ 5, 10, 15). The results of experiments based on the diversity and novelty measures for the Epinions dataset are shown in Table 5. These results indicate that the proposed method outperforms the other recommendation methods based on both of the diversity and novelty measures and also different lengths of recommendation lists. The values of diversity measure for the proposed method are 0.564, 0.612, and 0.657 for $L = 5$, $L = 10$, and $L = 15$, respectively. In these cases, the proposed method obtains the best results. The second best results based on the diversity measure are 0.479, 0.508, and 0.523 for $L = 5$, $L = 10$, and $L = 15$, respectively. Moreover, the novelty values of the proposed method are 8.973, 9.215, and 9.479 for $L = 5$, $L = 10$, and $L = 15$, respectively. The 2DGC method obtains the second best results based on the novelty measure for all of the recommendations list lengths. The novelty values of the 2DGC method are 7.923, 8.104, and 8.217 for $L = 5$, $L =$ 10, and $L = 15$, respectively. These results demonstrate that the proposed method can significantly outperform other recommendation methods in terms of the diversity and novelty measures. It can be seen from the results that the values of diversity and novelty measures for the proposed method increase when the lengths of recommendation lists are increased. Therefore, the length of recommendations list has a positive effect on the diversity and novelty measures of the proposed method.

The results of experiments for the diversity and novelty measures based on the Flixster dataset are represented in Table 6. As you can see from the results, the proposed method significantly outperforms other recommendation methods in terms of the diversity and novelty measures. The diversity values of the proposed method are 0.475, 0.486, and 0.498 for different recommendations list lengths. Moreover, the novelty values of the proposed method are 5.927, 6.458, and 6.873 for different recommendations list lengths. The 2DGC method obtains the second best results based on the both diversity and novelty measures for all of the recommendations list lengths. The diversity values of the 2DGC method are 0.382, 0.407, and 0.441 for $L = 5$, $L = 10$, and $L = 15$, respectively. On the other hand,

Table 5 Experiment results on the *Epinions dataset* for Diversity and Novelty measures of all users and different lengths of recommendations list (L)

Algorithms	L=5		L=10		L=15	
	Diversity	Novelty	Diversity	Novelty	Diversity	Novelty
UCF	0.379	6.545	0.386	6.671	0.392	6.815
ICF	0.382	6.728	0.394	6.861	0.405	6.981
SlopeOne	0.414	6.925	0.417	6.936	0.423	6.942
SVD+ +	0.418	6.936	0.422	6.951	0.424	6.957
TARS	0.401	6.906	0.415	7.068	0.419	7.127
SoRec	0.413	6.917	0.415	6.935	0.421	6.942
SoReg	0.421	6.965	0.425	6.973	0.428	6.982
TrustMF	0.426	7.105	0.435	7.236	0.439	7.354
SocialMF	0.437	7.334	0.441	7.459	0.445	7.612
TrustSVD	0.473	7.894	0.482	7.968	0.493	8.027
IPG	0.415	7.025	0.423	7.218	0.431	7.462
CETrust	0.454	7.219	0.467	7.382	0.481	7.497
2DGC	0.479	7.923	0.508	8.104	0.523	8.217
SoRVR	**0.564**	**8.973**	**0.612**	**9.215**	**0.657**	**9.479**

The best results are presented in boldface

the novelty values of the 2DGC method are 5.314, 5.873, and 6.109 for $L = 5$, $L = 10$, and $L = 15$, respectively. The results indicate that the diversity and novelty values increase when the lengths of the recommendations list are increased. Therefore, it can be concluded that the higher values of the recommendations list lengths lead to improve the performance of the proposed method based on the diversity and novelty measures. Table 7 shows the results of experiments for the diversity and novelty measures based on the FilmTrust dataset. It can be seen from the results that the proposed method obtains the best diversity and novelty values for all of the used recommendations list lengths in comparison with the other recommendation methods. The 2DGC is the second best method based on the diversity measure for different lengths of the recommendations list. Moreover, the CETrust method obtains the second best novelty values for $L = 5$ and $L = 15$ while the IPG method obtains the second best novelty value for $L = 10$. The higher values of the recommendations list lengths lead to increase the novelty values of the proposed method.

Table 6 Experiment results on the *Flixster dataset* for Diversity and Novelty measures of all users and different lengths of recommendations list (L)

Algorithms	L=5		L=10		L=15	
	Diversity	Novelty	Diversity	Novelty	Diversity	Novelty
UCF	0.221	4.203	0.236	4.354	0.258	4.623
ICF	0.243	4.486	0.262	4.784	0.287	4.913
SlopeOne	0.268	4.983	0.282	5.156	0.298	5.324
SVD+ +	0.271	4.992	0.287	5.178	0.305	5.428
TARS	0.265	4.979	0.283	5.230	0.301	5.472
SoRec	0.276	4.995	0.293	5.214	0.314	5.396
SoReg	0.282	4.997	0.295	5.247	0.319	5.468
TrustMF	0.291	5.011	0.302	5.312	0.331	5.593
SocialMF	0.309	5.105	0.318	5.394	0.352	5.638
TrustSVD	0.358	5.237	0.374	5.689	0.412	5.927
IPG	0.337	5.114	0.352	5.476	0.395	5.819
CETrust	0.375	5.283	0.391	5.786	0.437	6.014
2DGC	0.382	5.314	0.407	5.873	0.441	6.109
SoRVR	**0.475**	**5.927**	**0.486**	**6.458**	**0.498**	**6.873**

The best results are presented in boldface

Table 7 Experiment results on the *FilmTrust dataset* for Diversity and Novelty measures of all users and different lengths of recommendations list (L)

Algorithms	L=5		L=10		L=15	
	Diversity	Novelty	Diversity	Novelty	Diversity	Novelty
UCF	0.201	4.267	0.187	4.378	0.176	4.397
ICF	0.214	4.624	0.198	4.748	0.184	4.813
SlopeOne	0.228	4.889	0.212	4.893	0.198	4.896
SVD+ +	0.231	4.897	0.227	4.921	0.224	4.943
TARS	0.222	4.979	0.219	5.043	0.211	5.084
SoRec	0.235	4.907	0.234	4.935	0.232	4.961
SoReg	0.246	5.032	0.242	5.068	0.235	5.104
TrustMF	0.248	5.217	0.235	5.279	0.227	5.304
SocialMF	0.256	5.358	0.251	5.362	0.246	5.412
TrustSVD	0.271	5.847	0.267	5.897	0.262	5.968
IPG	0.269	5.793	0.265	5.991	0.258	6.013
CETrust	0.263	5.926	0.261	5.983	0.257	6.024
2DGC	0.285	5.714	0.279	5.905	0.268	5.984
SoRVR	**0.319**	**6.451**	**0.317**	**6.492**	**0.314**	**6.528**

The best results are presented in boldface

However, the performance of the proposed method based on the diversity measure decreases when the length of the recommendations lists is increased.

The experiments are repeated for the cold start users based on the diversity and novelty measures and the results are reported in Table 8 for the Epinions, Flixster, and FilmTrust datasets. It should be noted that, the length of recommendation lists is set to 5 (i.e. $L = 5$) because the cold start users in the experiments are the users who have less than 5 ratings. The results indicate that the proposed method outperforms other methods based on the diversity and novelty measures for the Epinions dataset. The obtained results for the proposed method based on the Epinions dataset are 0.362 and 8.814 for the diversity and novelty measures, respectively. Moreover, it can be seen that the proposed method obtains the best results based on the diversity and novelty measures for the Flixster and FilmTrust datasets. The diversity values of the proposed method are 0.335 and 0.098 for the Flixster and FilmTrust datasets, respectively. In addition, the proposed method

Table 8 Experiment results on the *Epinions, Flixster*, and *FilmTrust datasets* for Diversity and Novelty measures of cold start users: Length of recommendations list is equal to 5 (L=5)

Algorithms	Epinions		Flixster		FilmTrust	
	Diversity	Novelty	Diversity	Novelty	Diversity	Novelty
UCF	0.209	5.278	0.195	3.259	0.055	3.255
ICF	0.228	5.545	0.209	3.436	0.063	3.369
SlopeOne	0.234	5.625	0.212	3.512	0.061	4.287
SVD+ +	0.239	5.718	0.218	3.524	0.065	4.316
TARS	0.242	5.912	0.228	3.559	0.069	3.636
SoRec	0.241	5.856	0.231	3.573	0.076	4.346
SoReg	0.245	6.029	0.235	3.611	0.072	4.285
TrustMF	0.247	6.243	0.245	3.712	0.075	4.124
SocialMF	0.253	6.489	0.263	3.768	0.079	4.318
TrustSVD	0.289	7.125	0.296	3.976	0.084	4.956
IPG	0.257	7.059	0.274	3.914	0.078	4.223
CETrust	0.249	6.713	0.278	3.815	0.082	4.694
2DGC	0.262	7.352	0.286	4.012	0.081	4.857
SoRVR	**0.362**	**8.814**	**0.335**	**4.713**	**0.098**	**5.627**

The best results are presented in boldface

Table 9 Experiment results on the *Epinions dataset* for Precision, Recall, and F1 measures of all users and different lengths of recommendations list (L)

Algorithms	Precision			Recall			F1		
	L=5	L=10	L=15	L=5	L=10	L=15	L=5	L=10	L=15
UCF	0.104	0.045	0.041	0.148	0.195	0.215	0.122	0.074	0.069
ICF	0.126	0.044	0.041	0.174	0.245	0.253	0.146	0.075	0.071
SlopeOne	0.107	0.052	0.023	0.170	0.223	0.238	0.131	0.085	0.043
SVD+ +	0.142	0.076	0.053	0.213	0.254	0.287	0.170	0.116	0.089
TARS	0.146	0.078	0.061	0.234	0.263	0.284	0.179	0.120	0.100
SoRec	0.182	0.089	0.074	0.350	0.361	0.392	0.239	0.142	0.124
SoReg	0.188	0.093	0.086	0.381	0.397	0.401	0.251	0.150	0.141
TrustMF	0.253	0.231	0.204	0.489	0.497	0.543	0.333	0.315	0.296
SocialMF	0.298	0.283	0.265	0.512	0.572	0.596	0.376	0.378	0.366
TrustSVD	0.367	0.352	0.324	0.548	0.597	0.613	0.439	0.442	0.423
IPG	0.346	0.327	0.309	0.538	0.572	0.604	0.421	0.416	0.409
CETrust	0.337	0.319	0.281	0.497	0.536	0.581	0.402	0.399	0.379
2DGC	0.354	0.342	0.317	0.529	0.581	0.608	0.424	0.431	0.417
SoRVR	**0.439**	**0.412**	**0.398**	**0.635**	**0.672**	**0.722**	**0.519**	**0.511**	**0.513**

The best results are presented in boldface

obtains the novelty values 4.713 and 5.627 for the Flixster and FilmTrust datasets, respectively. The TrustSVD method obtains the second best results for the diversity measure based on all of the used datasets. The values of the diversity measure for the TrustSVD method are 0.289, 0.296, and 0.084 based on the Epinions, Flixster, and FilmTrust datasets, respectively. Moreover, the 2DGC method obtains the second best values for the novelty measure based on the Epinions and Flixster datasets. The novelty values of the 2DGC method are 7.352 and 4.012 for the Epinions and Flixster datasets, respectively. Moreover, the TrustSVD method obtains the second best result for the novelty measure based on the FilmTrust dataset. It can be concluded from the results of Table 8 that the proposed method can obtain better diversity and novelty values in comparison with other recommendation methods for the cold start users. Therefore, the proposed method can alleviate the cold start problem in the recommendation systems based on the diversity and novelty measures.

Tables 9–11 report the results of experiments based on the precision, recall, and F1 measures for all users' views and also different lengths of recommendation lists. The

Table 10 Experiment results on the *Flixster dataset* for Precision, Recall, and F1 measures of all users and different lengths of recommendations list (L)

Algorithms	Precision			Recall			F1		
	L=5	L=10	L=15	L=5	L=10	L=15	L=5	L=10	L=15
UCF	0.321	0.262	0.214	0.402	0.422	0.465	0.356	0.323	0.293
ICF	0.354	0.241	0.205	0.439	0.473	0.498	0.391	0.319	0.290
SlopeOne	0.327	0.274	0.229	0.425	0.461	0.496	0.369	0.343	0.313
SVD+ +	0.375	0.291	0.253	0.477	0.489	0.517	0.419	0.364	0.339
TARS	0.406	0.297	0.261	0.493	0.521	0.531	0.445	0.378	0.349
SoRec	0.468	0.304	0.291	0.511	0.576	0.583	0.488	0.397	0.388
SoReg	0.496	0.325	0.305	0.538	0.594	0.598	0.516	0.420	0.403
TrustMF	0.584	0.452	0.438	0.607	0.642	0.681	0.595	0.530	0.533
SocialMF	0.628	0.503	0.487	0.687	0.701	0.724	0.656	0.585	0.582
TrustSVD	0.673	0.531	0.512	0.716	0.754	0.786	0.693	0.623	0.620
IPG	0.679	0.546	0.537	0.723	0.768	0.791	0.700	0.638	0.640
CETrust	0.691	0.567	0.542	0.735	0.786	0.814	0.712	0.659	0.651
2DGC	0.684	0.539	0.521	0.718	0.746	0.765	0.701	0.626	0.619
SoRVR	**0.725**	**0.613**	**0.581**	**0.793**	**0.836**	**0.875**	**0.757**	**0.707**	**0.698**

The best results are presented in boldface

Table 11 Experiment results on the *FilmTrust dataset* for Precision, Recall, and F1 measures of all users and different lengths of recommendations list (L)

Algorithms	Precision			Recall			F1		
	L=5	L=10	L=15	L=5	L=10	L=15	L=5	L=10	L=15
UCF	0.409	0.289	0.195	0.428	0.581	0.598	0.418	0.386	0.294
ICF	0.431	0.289	0.203	0.471	0.553	0.563	0.450	0.380	0.299
SlopeOne	0.456	0.282	0.205	0.512	0.599	0.613	0.482	0.383	0.308
SVD+ +	0.440	0.286	0.196	0.464	0.581	0.601	0.451	0.383	0.296
TARS	0.458	0.292	0.208	0.473	0.556	0.589	0.465	0.382	0.307
SoRec	0.489	0.298	0.214	0.506	0.598	0.630	0.497	0.398	0.320
SoReg	0.491	0.301	0.216	0.502	0.601	0.613	0.496	0.401	0.319
TrustMF	0.478	0.284	0.217	0.490	0.592	0.627	0.484	0.384	0.322
SocialMF	0.478	0.315	0.212	0.508	0.634	0.643	0.493	0.421	0.318
TrustSVD	0.471	0.302	0.213	0.517	0.614	0.620	0.493	0.405	0.317
IPG	0.465	0.297	0.226	0.489	0.582	0.615	0.477	0.393	0.331
CETrust	0.483	0.361	0.234	0.505	0.572	0.609	0.494	0.443	0.338
2DGC	0.496	0.375	0.247	0.524	0.628	0.656	0.509	0.469	0.359
SoRVR	**0.526**	**0.416**	**0.295**	**0.648**	**0.691**	**0.724**	**0.581**	**0.519**	**0.419**

The best results are presented in boldface

results of experiments for the Epinions dataset are shown in Table 9. As you can see from these results, the proposed method outperforms other recommendation methods in terms of the precision, recall, and F1 measures for all of the used recommendations list lengths. The TrustSVD method obtains the second best results for all of the precision, recall, and F1 measures and also all of the recommendations list lengths. For example, the precision values of the proposed method are 0.439, 0.412, and 0.398 for $L = 5$, $L = 10$, and $L = 15$, respectively. Moreover, the TrustSVD method obtains the precision values 0.367, 0.352, and 0.324 for $L = 5$, $L = 10$, and $L = 15$, respectively. The values of precision measure for the proposed method decrease when the length of recommendations list is increased. However, increasing the length of recommendations list leads to increase the value of recall measure for the proposed method. Therefore, higher values for the length of recommendations list have negative and positive effects on the precision and recall of the proposed method, respectively. Table 10 shows the results of experiments based on the Flixster dataset. It can be seen from the results that the proposed method obtains the best precision, recall, and F1 values based on different lengths of the recommendations list. The precision values of the proposed method are 0.725, 0.613, and 0.581 for $L = 5$, $L = 10$, and $L = 15$, respectively. The CETrust method obtains the second best results based on the precision measure which its values are 0.691, 0.567, and 0.542 for $L = 5$, $L = 10$, and $L = 15$, respectively. Moreover, the CETrust is the second best method based on the recall and F1 measures for all lengths of the recommendations list. The results of experiments based on the FilmTrust dataset are shown in Table 11. The results indicate that the performance of the proposed method is better than other recommendation methods in terms of the precision, recall, and F1 measures. The precision values of the proposed method are 0.526, 0.416, and 0.295 for $L = 5$, $L = 10$, and $L = 15$, respectively. The 2DGC method obtains the second best results for the precision measure based on all lengths of the recommendations list. The obtained precision values for the 2DGC method are 0.496, 0.375, and 0.247 for $L = 5$, $L = 10$, and $L = 15$, respectively. Therefore, it can be concluded that the proposed method significantly outperforms other methods based on the precision measure. Moreover, the result values of the proposed method based on the recall measure are 0.648, 0.691, and 0.724 for $L = 5$, $L = 10$, and $L = 15$, respectively. On the other hand, the second best results for the recall measure are 0.524, 0.634, and 0.656 for $L = 5$, $L = 10$, and $L = 15$, respectively. The precision values of the proposed method decrease when the lengths of recommendations list (i.e. L) are increased for the FilmTrust dataset. In addition, higher values for the length of recommendations list lead to increase the recall values of the proposed method. The conducted experiments in Tables 9–11 indicate that the proposed method significantly improves the quality of recommendations based on the precision, recall, and F1 measures in comparison with other methods.

The experiments are repeated for the cold start users view based on the precision, recall, and F1 measures which the results are reported in Table 12. It should be noted that, the length of recommendation lists for the cold start view is set to 5 because the maximum number of ratings for the cold start users is equal to 5. The results indicate that the

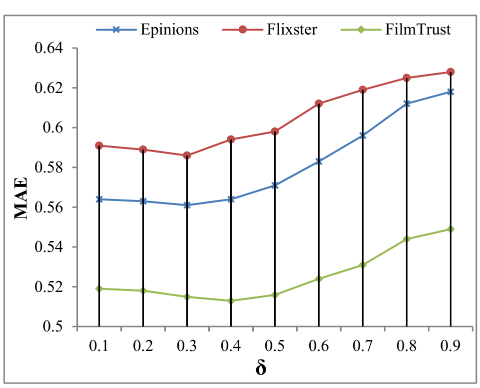
Table 12 Experiment results on the *Epinions*, *Flixster*, and *FilmTrust datasets* for Precision, Recall, and F1 measures of cold start users: Length of recommendations list is equal to 5 (L = 5)

Algorithms	Epinions			Flixster			FilmTrust		
	P@5	R@5	F1@5	P@5	R@5	F1@5	P@5	R@5	F1@5
UCF	0.056	0.112	0.074	0.254	0.315	0.281	0.122	0.390	0.186
ICF	0.082	0.149	0.105	0.296	0.376	0.331	0.147	0.416	0.217
SlopeOne	0.067	0.152	0.093	0.271	0.342	0.302	0.159	0.489	0.240
SVD+ +	0.085	0.195	0.118	0.343	0.405	0.371	0.169	0.541	0.257
TARS	0.097	0.201	0.130	0.379	0.436	0.405	0.171	0.449	0.247
SoRec	0.126	0.327	0.181	0.397	0.488	0.437	0.187	0.454	0.265
SoReg	0.159	0.368	0.222	0.435	0.511	0.469	0.193	0.469	0.273
TrustMF	0.198	0.472	0.278	0.571	0.591	0.580	0.215	0.678	0.327
SocialMF	0.215	0.493	0.299	0.594	0.637	0.614	0.223	0.703	0.338
TrustSVD	0.321	0.517	0.396	0.625	0.685	0.653	0.224	0.615	0.328
IPG	0.254	0.462	0.328	0.581	0.628	0.604	0.211	0.572	0.308
CETrust	0.284	0.459	0.351	0.611	0.652	0.631	0.208	0.556	0.303
2DGC	0.317	0.486	0.384	0.619	0.673	0.645	0.246	0.662	0.359
SoRVR	**0.386**	**0.592**	**0.467**	**0.674**	**0.745**	**0.708**	**0.308**	**0.714**	**0.430**

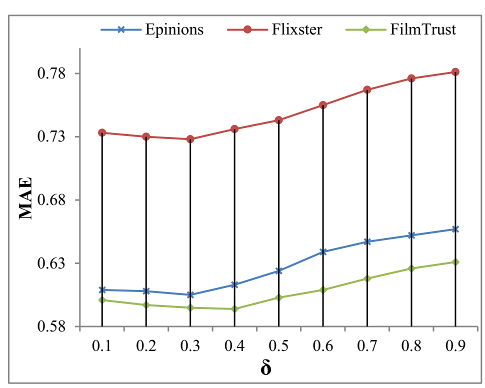
The best results are presented in boldface

proposed method obtains the best precision, recall, and F1 values for the Epinions, Flixster, and FilmTrust datasets. The TrustSVD is the second best method based on the Epinions and Flixster datasets for all of the precision, recall, and F1 measures. As you can see from Table 12, the values of the precision, recall, and F1 measures based on the Epinions dataset for the proposed method are 0.386, 0.592, and 0.467, respectively. On the other hand, the second best results based on the Epinions dataset are 0.321, 0.517, and 0.396 for the precision, recall, and F1 measures, respectively. Moreover, the precision, recall, and F1 values of the proposed method based on the Flixster dataset are 0.674, 0.745, and 0.708, respectively. The TrustSVD method obtains the values 0.625, 0.685, and 0.653 respectively for the precision, recall, and F1 measures based on the Flixster dataset. It can be concluded that the proposed method significantly improves the quality of the recommendations for the cold start users based on the precision, recall, and F1 measures. Therefore, the proposed method can be useful to alleviate cold start problem in the recommendation systems.

The parameter δ is an important parameter for the proposed method which is used in (2) to calculate the minimum number of required ratings for users. The performance of the proposed method depends on the value of the parameter δ. Therefore, several experiments are performed to evaluate the effect of different values of the parameter δ on the performance of the proposed method based on the used evaluation measures. Figure 5 shows the results of different δ values for the proposed method based on the MAE



(a)



(b)

Fig. 5 The effect of parameter δ on the system performance: **a** MAE for All users, **b** MAE for Cold users measure in the views of all users and cold start users. As you can see from these results, the value of the MAE measure increases when the value of the parameter δ is higher than a specific value. For example, the MAE values based on the Epinions and Flixster datasets increase for both of the all users and cold start users when the parameter δ is higher than 0.3. Moreover, the MAE values based on the FilmTrust dataset increase when the parameter δ is higher than 0.4. Therefore, it can be concluded that the higher values of the parameter δ have a negative effect on the performance of the proposed method based on the MAE measure.

Figure 6 shows the results of experiments based on different δ values for the diversity and novelty measures in the views of all users and cold start users. The results indicate that the diversity values for all the users and cold start users decrease when the value of parameter δ is increased from 0.1 to 0.9. It can be concluded that the performance of the proposed method based on the diversity measure will be reduced with increasing the value of parameter δ. Therefore, the higher values of the parameter δ have a negative effect on the diversity measure of the proposed method for the views of all users and cold start users. Moreover, the results demonstrate that the novelty values of the proposed method decrease when the value of parameter δ is increased for the views of all users and cold start users. Therefore, the performance of the proposed method for the novelty measure depends on the value of the parameter δ. In other words, the higher values of the parameter δ lead to reduce the quality of recommendations for the proposed method based on the novelty measure. The experiments are repeated for the precision, recall and F1 measures and the results are reported in Fig. 7. As you can see from Fig. 7, the values of the precision measure for the views of all users and cold start users decrease when the value of the parameter δ is increased. Therefore, the higher values of the parameter δ have a negative effect on the performance of the proposed method based on the precision measure. In most cases, increasing the value of the parameter δ from 0.1 to 0.9 leads to decrease the values of the recall measure for the proposed method. It can be seen that the F1 values of the proposed method will be decreased when the value of the parameter δ increases from 0.1 to 0.9. The value of parameter δ is set to $\delta = 0.3$ for the Epinions and Flixster datasets and $\delta = 0.4$ for the FilmTrust dataset to compare the proposed method with the other recommendation methods.

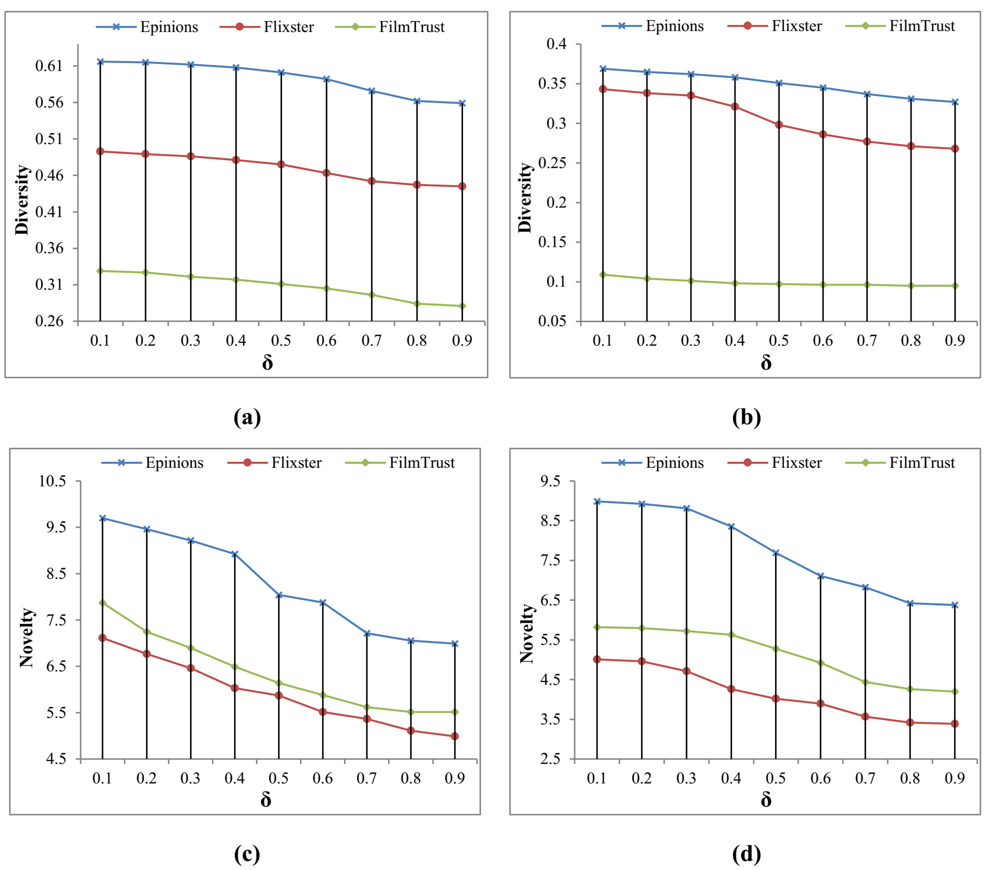


Fig. 6 The effect of parameter δ on the system performance: **a** diversity for All users, **b** diversity for Cold users, **c** novelty for All users, **d** novelty for Cold users

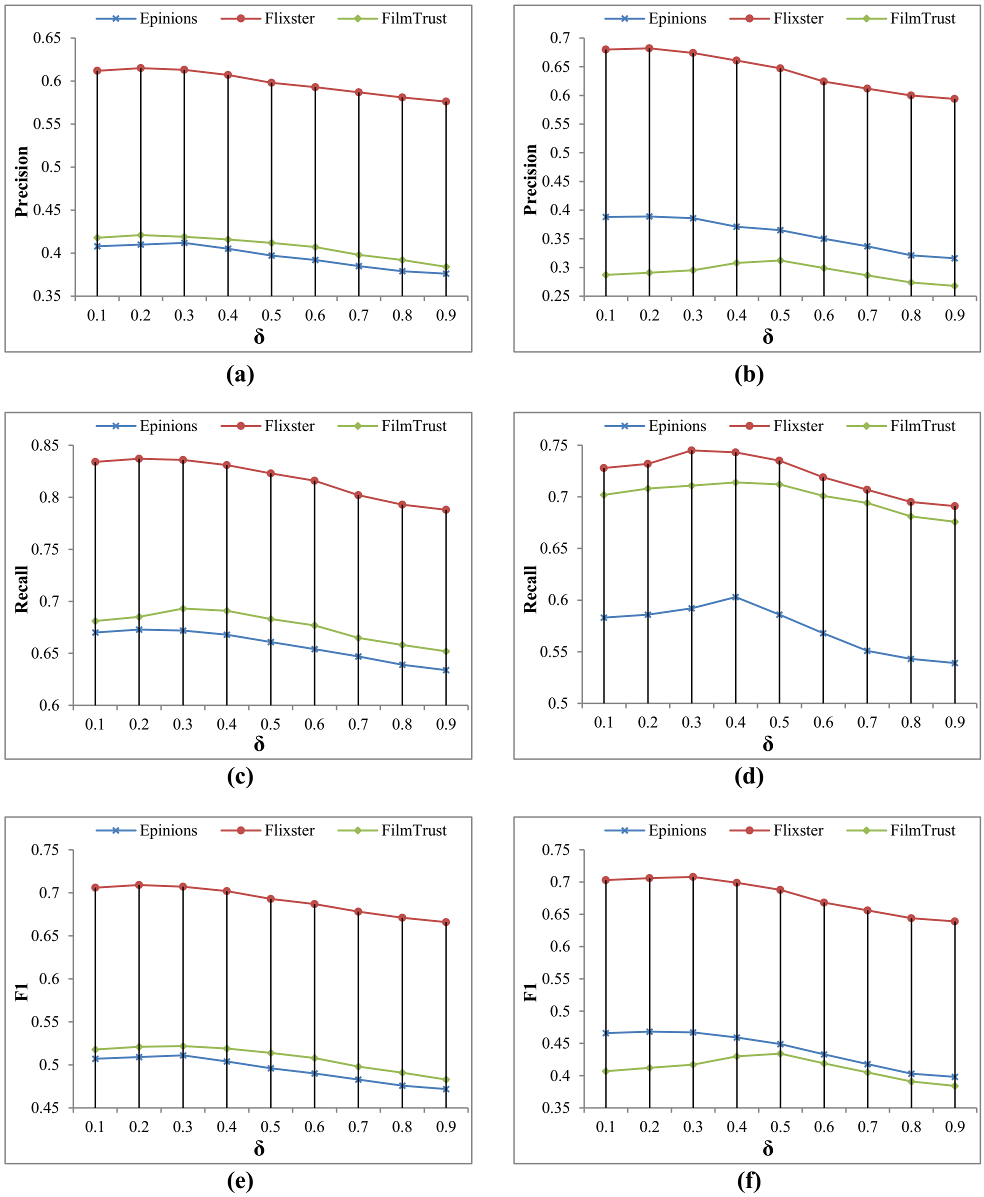


Fig. 7 The effect of parameter δ on the system performance: **a** precision for All users, **b** precision for Cold users, **c** recall for All users, **d** recall for Cold users, **e** F1 for All users, **f** F1 for Cold users

5 Conclusions

Social recommendation systems mainly use both of the social information and historical ratings to provide relevant suggestions for the users. The performance of the systems depends on the quality of the used resources for the prediction process. The systems cannot provide reliable predictions for the users who have insufficient ratings in their profiles. Therefore, one of the most important issues in the recommendation systems is to determine the minimum number of required ratings for providing reliable predictions. This paper proposed a probabilistic method to calculate the minimum number of required ratings for the users to predict reliable ratings for the unseen items. Moreover, a novel mechanism is considered to improve the performance of the rating profiles of the users with insufficient ratings. The mechanism is based on incorporating reliable virtual ratings into historical ratings of the users who have unreliable rating profiles. On the other hand, a novel selection mechanism for the virtual ratings is proposed which considers the reliability, diversity, and novelty measures. One of the main advantages of the proposed method is to alleviate the cold start and data sparsity problems in the recommendation systems by adding the reliable virtual ratings. In other words, the proposed method makes a denser user-item rating matrix in comparison with the original one. Experimental results on three well-known datasets demonstrated that the proposed method significantly outperformed other recommendation methods in terms of several evaluation measures for both of the all users and cold start users' views.

Future works will be focused on considering other social information such as distrust relations among the users into the recommendation process. Moreover, the temporal information of the ratings can be used in the proposed selection mechanism to consider the changes of users' preferences over time. On the other hand, the proposed method can be used in other types of recommendation systems such as context-aware recommendation systems to improve their performance.

References

1. Ortega F, Hernando A, Bobadilla J, Kang JH (2016) Recommending items to group of users using matrix factorization based collaborative filtering. Inf Sci 345:313–324
2. Zheng L, Zhu F, Huang S, Xie J (2017) Context Neighbor Recommender: Integrating contexts via neighbors for recommendations. Inf Sci 414:1–18
3. Ghazarian S, Nematbakhsh MA (2015) Enhancing memory-based collaborative filtering for group recommender systems. Expert Syst Appl 42:3801–3812
4. Luo X, Xia Y, Zhu Q, Li Y (2013) Boosting the K-Nearest-Neighborhood based incremental collaborative filtering. Knowl-Based Syst 53:90–99
5. Shi Y, Larson M, Hanjalic A (2013) Unifying rating-oriented and ranking-oriented collaborative filtering for improved recommendation. Inf Sci 229:29–39
6. Moradi P, Ahmadian S, Akhlaghian F (2015) An effective trust-based recommendation method using a novel graph clustering algorithm. Physica A: Stat Mech Appl 436:462–481
7. Boratto L, Carta S, Fenu G, Saia R (2017) Semantics-aware content-based recommender systems: Design and architecture guidelines. Neurocomputing 254:79–85
8. Bagher RC, Hassanpour H, Mashayekhi H (2017) User trends modeling for a content-based recommender system. Expert Syst Appl 87:209–219
9. Paradarami TK, Bastian ND, Wightman JL (2017) A hybrid recommender system using artificial neural networks. Expert Syst Appl 83:300–313
10. Ning X, Karypis G (2012) Sparse linear methods with side information for top-n recommendations. In: Proceedings of the sixth ACM conference on Recommender systems, RecSys '12. ACM, New York, pp 155–162
11. Bedi P, Vashisth P (2014) Empowering recommender systems using trust and argumentation. Inf Sci 279:569–586
12. Massa P, Avesani P (2007) Trust-aware recommender systems. Paper presented at the ACM conference on recommender systems, RecSys'07, Minneapolis
13. Zhang Z, Liu H (2015) Social recommendation model combining trust propagation and sequential behaviors. Appl Intell 43(3):695–706
14. Ozsoy MG, Polat F, Alhajj R (2016) Making recommendations by integrating information from multiple social networks. Appl Intell 45(4):1047–1065
15. Alahmadi DH, Zeng XJ (2015) ISTS: Implicit Social trust and sentiment based approach to recommender systems. Expert Syst Appl 42:8840–8849
16. Gao Q, Gao L, Fan J, Ren J (2017) A preference elicitation method based on bipartite graphical correlation and implicit trust. Neurocomputing 237:92–100
17. Son LH (2016) Dealing with the new user cold-start problem in recommender systems: a comparative review. Inf Syst 58:87–104
18. Lika B, Kolomvatsos K, Hadjiefthymiades S (2014) Facing the cold start problem in recommender systems. Expert Syst Appl 41:2065–2073
19. Wu H, Yue K, Pei Y, Li B, Zhao Y, Dong F (2016) Collaborative Topic Regression with social trust ensemble for recommendation in social media systems. Knowl-Based Syst 97:111–122
20. Seo YD, Kim YG, Lee E, Baik DK (2017) Personalized recommender system based on friendship strength in social network services. Expert Syst Appl 69:135–148
21. Zhang Z, Xu G, Zhang P, Wang Y (2017) Personalized recommendation algorithm for social networks based on comprehensive trust. Appl Intell 47(3):659–669
22. Yang B, Lei Y, Liu D, Liu J (2013) Social collaborative filtering by trust. In: Proceedings of the 23th international joint conference on Artificial Intelligence (IJCAI), pp 2747–2753
23. Chen CC, Wan YH, Chung MC, Sun YC (2013) An effective recommendation method for cold start new users using trust and distrust networks. Inf Sci 224:19–36
24. Martinez-Cruz C, Porcel C, Bernabé-Moreno J, Herrera-Viedma E (2015) A model to represent users trust in recommender systems using ontologies and fuzzy linguistic modeling. Inf Sci 311:102–118
25. Guo G, Zhang J, Smith NY (2015) TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence, pp 123–129
26. Bathla G, Aggarwal H, Rani R (2017) A graph-based model to improve social trust and influence for social recommendation. The Journal of Supercomputing. https://doi.org/10.1007/s11227-017-2196-2
27. Christensen I, Schiaffino S, Armentano M (2016) Social group recommendation in the tourism domain. J Intell Inform Syst 47(2):209–231
28. Jamali M, Ester M (2010) A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the 4th ACM conference on Recommender systems (RecSys), pp 135–142
29. Gohari FS, Aliee FS, Haghighi H (2018) A new confidence-based recommendation approach: Combining trust and certainty. Inf Sci 422:21–50
30. Li YM, Kao CP (2009) TREPPS: A Trust-based Recommender System For Peer Production Services. Expert Syst Appl 36(2):3263–3277
31. Ma H, Yang H, Lyu MR (2008) King I SoRec: social recommendation using probabilistic matrix factorization. In: Proceedings of the 17th ACM conference on Information and knowledge management
32. Servajean M, Akbarinia R, Pacitti E, Yahia SA (2015) Profile diversity for query processing using user recommendations. Inf Syst 48:44–63
33. Zhang M, Hurley N (2008) Avoiding monotony: Improving the diversity of recommendation lists. In: 2nd ACM Conference on Recommender Systems, Recsys 2008. ACM, New York, pp 123–130
34. Hurley N, Zhang M (2011) Novelty and diversity in top-N recommendations analysis and evaluation. ACM Trans Internet Technol 10:1–29

35. Lee KI, Lee K (2015) Escaping your comfort zone: a graph-based recommender system for finding novel recommendations among relevant items. Expert Syst Appl 42(10):4851–4858
36. Noia TD, Rosati J, Tomeo P, Sciascio ED (2017) Adaptive multi-attribute diversity for recommender systems. Inf Sci 234-253:382–383
37. Gogna A, Majumdar A (2017) DiABlo: Optimization based design for improving diversity in recommender system. Inf Sci 378:59–74
38. Gu L, Yang P, Dong Y (2017) Diversity optimization for recommendation using improved cover tree. Knowl-Based Syst 135:1–8
39. Hernando A, Bobadilla J, Ortega F, Tejedor J (2013) Incorporating reliability measurements into the predictions of a recommender system. Inf Sci 218:1–16
40. Moradi P, Ahmadian S (2015) A reliability-based recommendation method to improve trust-aware recommender systems. Expert Syst Appl 42:7386–7398
41. Zhang M, Guo X, Chen G (2016) Prediction uncertainty in collaborative filtering: Enhancing personalized online product ranking. Decis Support Syst 83:10–21
42. Guo G, Zhang J, Thalmann D (2014) Merging trust in collaborative filtering to alleviate data sparsity and cold start. Knowl-Based Syst 57:57–68
43. Xie H, Lui JCS (2015) Mathematical modeling and analysis of product rating with partial information. ACM Trans Knowl Discov Data 9(4):1–33
44. Toledo RY, Mota YC, Martínez L (2015) Correcting noisy ratings in collaborative recommender systems. Knowl-Based Syst 76:96–108
45. Yuan W, Guan D, Lee YK, Lee S, Hur SJ (2010) Improved trust-aware recommender system using small-worldness of trust networks. Knowl-Based Syst 23(3):232–238
46. Bahmani B, Kumar R, Vassilvitskii S (2012) Densest subgraph in streaming and mapreduce. Proc VLDB Endowment 5(5):454–465
47. Page L, Brin S, Motwani R, Winograd T (1999) The PageRank Citation Ranking: Bringing Order to the Web. Stanford InfoLab, Stanford
48. Erkan G, Radev DR (2004) Lexrank: Graph-based Lexical Centrality as Salience in Text Summarization. J Artif Intell Res 22:457–479
49. Zhou Y, Lei T, Zhou T (2011) A robust ranking algorithm to spamming. Europhysics Letters 94:1–6
50. Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence, pp 43–52
51. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: 10th International Conference on World Wide Web. ACM, New York, pp 285–295
52. Lemire D, Maclachlan A (2005) Slope one predictors for online rating-based collaborative filtering. In: Proceedings of the 2005 SIAM International Conference on Data Mining, pp 471–475
53. Koren Y (2008) Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. Las Vegas, pp 426–434
54. Ma H, Zhou D, Liu C, Lyu MR, King I (2011) Recommender systems with social regularization. In: Proceedings of the fourth ACM international conference on Web search and data mining. ACM, Hong Kong, pp 287–296
55. Sheugh L, Alizadeh SH (2018) A novel 2D-Graph clustering method based on trust and similarity measures to enhance accuracy and coverage in recommender systems. Inf Sci 432:210–230
56. Shannon CE (2001) A mathematical theory of communication. ACM SIGMOBILE Mob Comput Commun Rev 5:3–55
57. Herlocker JL, Konstan JA, Terveen LG, Reidl J (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst 22(1):5–53

Sajad Ahmadian received his B.S. degree in computer engineering from Razi University, Kermanshah, Iran, in 2011, and the M.S. degree in computer engineering from University of Kurdistan, Sanandaj, Iran in 2014. Since 2015, he is a PhD student at University of Zanjan, Zanjan, Iran. His research interests include recommender systems, social networks analysis, and machine learning.

Majid Meghdadi received the B.S. degree in Mathematics and Applied Computer from University of Tehran, Iran in 1991 and the M.S. degree in Computer Engineering from University of Sharif, Iran in 1994 and the PhD degree in Electronic and Computer Education in 2010 from Gazi University, Turkey. His current research interests include data mining, recommender systems, and social networks analysis. He is as an Assistant Professor of the Department of Computer Engineering, University of Zanjan, Zanjan, Iran.

Mohsen Afsharchi received the B.S. degree in computer engineering from University of Isfahan, Isfahan, Iran in 1994 and the M.S. degree in computer architecture from University of Science and Technology, Tehran, Iran in 1997 and the PhD degree in artificial intelligence from University of Calgary, Calgary, Canada in 2006. His current research interests include recommender systems, social networks analysis, and multi-agent systems. He is as an Associate Professor of the Department of Computer Engineering, University of Zanjan, Zanjan, Iran.