Information Processing and Management 54 (2018) 707–725

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

A social recommendation method based on an adaptive neighbor selection mechanism

Sajad Ahmadian, Majid Meghdadi*, Mohsen Afsharchi

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

ARTICLE INFO

Article history:
Received 9 November 2016
Revised 30 January 2017
Accepted 10 March 2017
Available online 18 March 2017

Keywords:
Recommender systems
Adaptive neighbor selection
Confidence
Reliability
Trust

ABSTRACT

Recommender systems are techniques to make personalized recommendations of items to users. In e-commerce sites and online sharing communities, providing high quality recommendations is an important issue which can help the users to make effective decisions to select a set of items. Collaborative filtering is an important type of the recommender systems that produces user specific recommendations of the items based on the patterns of ratings or usage (e.g. purchases). However, the quality of predicted ratings and neighbor selection for the users are important problems in the recommender systems. Selecting suitable neighbors set for the users leads to improve the accuracy of ratings prediction in recommendation process. In this paper, a novel social recommendation method is proposed which is based on an adaptive neighbor selection mechanism. In the proposed method first of all, initial neighbors set of the users is calculated using clustering algorithm. In this step, the combination of historical ratings and social information between the users are used to form initial neighbors set for the users. Then, these neighbor sets are used to predict initial ratings of the unseen items. Moreover, the quality of the initial predicted ratings is evaluated using a reliability measure which is based on the historical ratings and social information between the users. Then, a confidence model is proposed to remove useless users from the initial neighbors of the users and form a new adapted neighbors set for the users. Finally, new ratings of the unseen items are predicted using the new adapted neighbors set of the users and the *top_N* interested items are recommended to the active user. Experimental results on three real-world datasets show that the proposed method significantly outperforms several state-of-the-art recommendation methods.

1. Introduction

In recent years, the volume of data on the web has been increasing at an unprecedented rate and the information overload creates difficulties for the Internet users. Therefore, it is difficult for the users to find useful information among the available choices. Recommender systems (RSs) are used to make personalized recommendations for the users to overcome the information overload problem. In other words, the RSs help the users to find items (e.g. books, movies, news, etc.) of interest from a plethora of available choices. In order to provide high quality recommendations by the RSs, these systems

* Corresponding author.
E-mail addresses: s.ahmadian@znu.ac.ir (S. Ahmadian), meghdadi@znu.ac.ir (M. Meghdadi), afsharchim@znu.ac.ir (M. Afsharchi).

http://dx.doi.org/10.1016/j.ipm.2017.03.002
0306-4573/

need to predict and compare the utility of items and then decide what items to recommend based on this comparison (Ricci, Rokach, Shapira, & Kantor, 2011).

Collaborative filtering (CF) is one of the recommender systems methods which is used to exploit information about the past behavior or the opinions of an existing user to predict unrated items and find the items that the current user of the system will most probably like or be interested in. The basic idea of the CF-based methods is that if the users shared the same interests in the past then they will have similar tastes in the future (Cechinel, Sicilia, Alonso, & Barriocanal, 2013; Domingues, Jorge, & Soares, 2013; Park, Park, Jung, & Lee, 2015). In these methods, a matrix of given user-item ratings is used as the input to calculate similarity values between the users. Moreover, these methods use the similarity values to determine the distance between a pair of the users and also form neighbors set of the active user. Therefore, the opinions of the users in the neighbors set of the active user are used to provide suitable recommendations for the active user in these methods. The CF-based methods are often classified into two groups including memory-based and model-based methods. In the memory-based methods, the original user-item ratings matrix is stored in memory and used directly for making the recommendations. On the other hand, in the model-based methods, the raw data are processed offline as item-based filtering or some dimensionality reduction approaches. Then, the pre-computed or learned model is used to make predictions at run time. Some of the model-based methods have been presented in the literature including probabilistic models (Javari & Jalili, 2015; Ma, Zhang, Liu, Li, & Yuan, 2016), clustering models (Bilge & Polat, 2013; Tsai & Hung, 2012), dimensionality reduction methods (Hernando, Bobadilla, & Ortega, 2016; Hong, Zheng, & Chen, 2016), pattern mining techniques (Tsai & Lai, 2015), latent semantic models (Hofmann, 2004), and Markov decision process models (Shani, Brafman, & Heckerman, 2005). Although the memory-based methods are more widely used than the model-based methods, such methods face problems of scalability of the systems with tens of millions of users and millions of items (Ricci et al., 2011).

The CF-based methods often suffer from several shortcomings which lead to reduce of the system performance. These methods use the user-rating matrix to compute the similarity values between the users and form the neighbors set of the active user (Kaššák, Kompan, & Bieliková, 2016; Liu, Hu, Mian, Tian, & Zhu, 2014). Therefore, the CF-based methods need the extensive data which contains exclusively the ratings made by the users over most of the items to make efficient recommendations. However, in real-world applications, the user-rating matrixes tend to be very sparse which means the users typically provide ratings for only a small number of the items. This problem is called data sparsity problem which makes to reduce the performance of the CF-based methods for finding the neighbors set of the active user. Moreover, the cold-start problem can be viewed as a special case of the data sparsity problem. This problem is about the new users that have not rated any item yet or the users who have rated a few items. In addition, recommending the items that have not been rated or bought yet is another challenge of the CF-based methods, as there are not available sufficient feedbacks on these items. Malicious attacks are another problem of the CF-based methods which can reduce the reliability of the systems (O'Mahony, Hurley, & Silvestre, 2005). A malicious attack occurs when a user tries to influence the functioning of the system intentionally. In other words, a malevolent user might try to influence the behavior of the recommender system in such a way that includes a certain item very often in its recommendation list.

To overcome the mentioned problems of the CF-based methods, several approaches have been proposed in the literature. Trust-aware recommender systems aim to exploit the social information from trust networks of the users to improve the system performance of the CF-based methods (Fang, Guo, & Zhang, 2015; Kim & Phalak, 2012; Yan, Zheng, Chen, & Wang, 2013). In these systems, the opinions of the trusted neighbors can be used as a starting point for predicting unrated items and making recommendations to the users. Therefore, these systems can be able to alleviate the cold-start problem and improve on the user coverage measure. Moreover, the trust statements between the users help to make the recommender systems more robust against malicious attacks. It is due to this fact that desired trust relationships to a fake profile cannot easily be injected into a recommender database.

Neighbor selection for the users is one of the most important issues in recommender systems which it has a high impact on the accuracy of the predicted ratings in these systems. Most of the recommendation methods use the identified neighbors set of the users to predict unseen items and produce some recommendations to the users. However, these neighbors of the users may not be useful to predict all of the unseen items and it leads to reduce the accuracy of the predicted ratings in the recommendation process. In this paper, a social recommendation method is proposed which is based on an adaptive neighbor selection mechanism to improve the accuracy of ratings prediction and resolve neighbor selection problem in the recommender systems. To this end, the initial neighbors set of the users is calculated using clustering algorithm which is based on the combination of the similarity values and the trust information between each pair of the users. Then, the initial ratings of the unseen items are predicted using the initial neighbors set of the users and also the quality of the predicted ratings is evaluated using a reliability measure which is based on the similarity values and the trust statements between the users. Then, an adaptive neighbor selection mechanism is proposed to calculate a new neighbors set for the users who their initial neighbors sets have a low reliability measure to predict a target item. For this purpose, a novel confidence model between the users is proposed to identify and remove the useless users from the initial neighbors set of the users. Finally, the final ratings of the unseen items are predicted using the new adapted neighbors set of the users and the *top_N* interested items are recommended to the active user.

The remainder of this paper is organized as follows: Related studies are reviewed in Section 2. Section 3 introduces the proposed method. In Section 4, the proposed method is compared with the state-of-the-art methods by performing several experiments on three well-known datasets. Finally, some concluding comments are discussed in Section 5.

2. Related works

Most of the collaborative filtering approaches use a two-dimensional matrix which contains the ratings of the users to the specific items. However, in many real world applications, this matrix of information is not sufficient for the recommender systems to make accurate recommendations for the users (Lee, Kahng, & Lee, 2015; Moon, Kim, & Ryu, 2013). Therefore, using additional sources such as social factors or trust information can improve the performance of the recommender systems which these kinds of systems are called trust-aware recommender systems (Massa & Avesani, 2007). In trust-aware methods, there are two types of trust information which include explicit and implicit trust statements. In the explicit methods, the trust information can be explicitly collected from the users based on the pre-established social links among the users in the social networks (Bedi & Vashisth, 2014; Yan et al., 2013). On the other hand, in the implicit methods, the trust information can be implicitly inferred from users' rating information (Alahmadi & Zeng, 2015).

In Guo, Zhang, and Thalmann (2014), a novel method is proposed to incorporate social trust information in providing recommendations for the users. To this end, ratings of trusted neighbors for a given user are merged to complement and represent the preferences of the user and to find other users with similar preferences. Also, the number of ratings and the ratio of conflicts between positive and negative opinions are used as the confidence to measure the quality of merged ratings. In Deng, Huang, and Xu (2014), a recommendation method is proposed which is based on a matrix factorization method to assess the degree of trust between the users in a social network. Moreover, in this method, an extended random walk algorithm is proposed to obtain recommendation results. Yang, Guo, and Liu (2013) proposed a Bayesian-inference-based method for online social networks which the rating similarity between the users is measured by a set of conditional probabilities derived from their mutual rating history.

In addition, incorporating social information into matrix factorization based recommendation methods can be useful to improve the performance of these systems in predicting unseen items for target users. In Yang, Lei, Liu, and Liu (2013), a recommendation method is proposed to improve the performance of the collaborative filtering approach by means of integrating twofold sparse information including the historical ratings data given by the users and the social trust network among the same users. Moreover, a matrix factorization technique is used in this method to map the users into low-dimensional latent feature spaces in terms of their trust information for reflecting users' reciprocal influence on their own opinions more reasonably. A model-based method is proposed in Jamali and Ester (2010) for recommendation in social networks which is based on matrix factorization techniques. To this end, a trust propagation mechanism is incorporated into the matrix factorization based recommendation methods. The trust propagation mechanism has been shown to be a crucial phenomenon in the social recommendation methods. In Guo, Zhang, and Smith (2015b), a trust-based matrix factorization method is proposed which is based on both of the explicit and implicit influence of user trust and of item ratings. The authors have shown that not only the explicit but also the implicit influence of both ratings and trust should be taken into consideration in a recommendation method. Therefore, they proposed a method to involve the explicit and implicit influence of rated items, by further incorporating both the explicit and implicit influence of trusted users on the prediction of items for an active user.

Clustering algorithms can be used in the recommender systems to group the users/items into several clusters. Moreover, the recommender systems are demanded to elaborate a prediction taking into account the users/items in the same cluster that the users/items belong to. In Pereira and Hruschka (2015), a hybrid method is proposed to minimize system degradation which is based on an existing clustering algorithm. This method combines collaborative filtering with demographic information that can address the cold start problem where no ratings are available for new users. Moreover, in Birtolo and Ronca (2013), the authors presented a clustering framework which includes two clustering-based collaborative filtering algorithms. One of these algorithms (called IFCCF) is based on fuzzy clustering for items of the system and the other algorithm (called TRACCF) is based on trust-aware clustering collaborative filtering for users of the system. In Guo, Zhang, and Smith (2015a), a multiview clustering method is proposed which users are iteratively clustered from the views of both rating patterns and social trust relationships. In Moradi, Ahmadian, and Akhlaghian (2015), a recommendation method is proposed which is based on a novel graph clustering algorithm and also trust statements between the users. To this end, the identified clusters are used as a set of neighbors to predict unseen items and make recommendations to the current active user. Moreover, the initial center for each cluster is determined automatically, thus, the algorithm does not require setting the number of predefined clusters.

In all of the above recommendation methods, a static neighbor selection mechanism is used to predict the ratings of unseen items for the active user. In other words, the initial sets of neighbors for the users are directly used to predict the ratings of unseen items without considering the effectiveness of each user in these sets of neighbors. However, the initial constructed sets of neighbors for the users may not be useful to predict all of the unseen items and it leads to reduce the accuracy of the predicted ratings in the recommendation process. The main contribution of the proposed method in this paper is to use an adaptive neighbor selection mechanism to improve the accuracy of predicted ratings and resolve neighbor selection problem in the recommender systems. To this end, the initial ratings of the unseen items are predicted using the initial neighbors set of the users and then the quality of the predicted ratings is evaluated using a proposed reliability measure. Moreover, an adaptive neighbor selection mechanism based on a novel confidence model between the users is proposed to calculate a new neighbors set for the users who their initial neighbors sets have a low reliability measure to predict a target item.

3. Proposed method

In this section, a novel recommender system is proposed which is called Social Recommendation based on an Adaptive Neighbor Selection (in short SRANS). In the proposed method, first of all, the initial neighbors set of the users are calculated using a clustering algorithm (Moradi et al., 2015). Then, the identified initial neighbors set of the users are used to predict the initial ratings of the unseen items. Moreover, the quality of the initial predicted ratings is evaluated using a reliability measure which is based on the similarity values and the trust statements between the users. Then, an adaptive neighbor selection mechanism is used to select a subset of the initial neighbors set of the users for the predicted ratings with low reliability measures. To this end, a confidence model between the users is proposed to remove useless users from the initial neighbors set of the users. Finally, the new ratings of the unseen items are predicted using the new adapted neighbors set of the users and the *top_N* interested items are recommended to the active user. Additional details about the steps of the proposed method are discussed in following subsections.

3.1. Initial neighbor selection

In the first step of the proposed method, nearest neighbors of the users are initialized by using a user clustering approach. The main idea behind user clustering is to group the original users set into several clusters with similar properties. Therefore, the users in the same cluster are similar to each other and can be used as the nearest neighbors set in rating prediction process. Additional details about the steps of the initial neighbor selection are discussed in following subsections.

3.1.1. Graph construction

In this step, a given users set are mapped into a graph $G = (V, E, W)$, where V denotes the set of the users, E is the set of edges between the user pairs, and W indicates similarity weight between each pair of the users. In the proposed method, a combination of the similarity values and also the trust statements is used to compute the final similarity weights between the users. The similarity values between the users are calculated by means of Pearson correlation coefficient function as Eq. (1):

$$sim(u,v) = \frac{\sum_{i\in A_{u,v}} (r_i(u) - \bar{r}(u))(r_i(v) - \bar{r}(v))}{\sqrt{\sum_{i\in A_{u,v}} (r_i(u) - \bar{r}(u))^2}\sqrt{\sum_{i\in A_{u,v}} (r_i(v) - \bar{r}(v))^2}} \tag{1}$$

where, $r_i(u)$ is the rate of item i given by user u, $\bar{r}(u)$ is the average of the rates given by user u, and $A_{u,\,v}$ is the set of items which are rated by both of the users u and v.

Moreover, the trust values between the users can be calculated using Eq. (2):

$$T_{u,v} = \frac{d_{max} - d_{u,v} + 1}{d_{max}} \tag{2}$$

where, $d_{u,\,v}$ is the trust propagation distance between the users u and v (Massa & Avesani, 2007), and d_{max} represents the maximum allowable propagation distance between the users which is calculated as Eq. (3):

$$d_{max} = \frac{\ln(n)}{\ln(k)} \tag{3}$$

where, n and k are respectively the size and the average degree of the trust networks in a specific recommender system (Yuan, Guan, Lee, Lee, & Hur, 2010).

Finally, the combination of the similarity and trust values between the users is calculated as the final similarity weight using Eq. (4):

$$w_{u,v} = \begin{cases} \dfrac{2 \times sim(u,v) \times T_{u,v}}{sim(u,v) + T_{u,v}} & if\ sim(u,v) + T_{u,v} \neq 0\ and\ sim(u,v) \times T_{u,v} \neq 0 \\ T_{u,v} & else\ if\ sim(u,v) = 0\ and\ T_{u,v} \neq 0 \\ sim(u,v) & else\ if\ sim(u,v) \neq 0\ and\ T_{u,v} = 0 \\ 0 & else \end{cases} \tag{4}$$

where, $sim(u,\,v)$ and $T_{u,\,v}$ are the similarity value and the trust value between the users u and v which are calculated using Eqs. (1) and (2), respectively.

3.1.2. Finding initial centers set

In this step, the initial center for each cluster is found automatically thus, the clustering method does not require setting the number of predefined clusters. To this end, a graph-based approach (Bahmani, Kumar, & Vassilvitskii, 2012) is used to find a subgraph as initial centers set of the clustering method. In the proposed method, the graph-based approach is modified to find sparsest subgraph for using as initial centers set of the clustering method. Let $G = (V, E)$ be an undirected

Algorithm 1. Finding initial centers set.

Inputs: $G = (V, E)$, $k > 0$, and $\varepsilon > 0$.
Output: Initial centers set $\tilde{S}$.
Algorithm:
1: ***Set*** $S = V$ and $\tilde{S} = V$;
2: **if** $S \neq \emptyset$ then **go to** step 3 **else go to** step 19;
3: ***Calculate*** the density of S (i.e. $\rho(S)$) using Eq. (5);
4: ***Set*** $\tilde{A}(S) = \emptyset$;
5: **for** all $i \in S$ **do**
6: ***Calculate*** the weighted degree of node i (i.e. $wd_S(i)$) using Eq. (6);
7: **if** $wd_S(i) \geq (2 + 2\varepsilon) * \rho(S)$ **then**
8: $\tilde{A}(S) = \tilde{A}(S) \cup \{i\}$;
9: **end if**
10: **end for**
11: ***Sort*** all $i \in \tilde{A}(S)$ descending based on their $wd_S(i)$;
12: ***Set*** $r = \frac{\varepsilon}{1+\varepsilon} \times |\tilde{A}(S)|$;
13: ***Select*** top_r nodes from $\tilde{A}(S)$ as $A(S)$;
14: ***Set*** $S = S - A(S)$;
15: **if** $|S| \geq k$ and $\rho(S) < \rho(\tilde{S})$ **then**
16: $\tilde{S} = S$;
17: **end if**
18: ***Go*** **to** step 2;
19: ***Return*** $\tilde{S}$;

Fig. 1. The pseudo code of the algorithm for finding initial centers set.

weighted graph which is constructed in the graph construction step (See Section 3.1.1). The density of the subgraph $S \subseteq V$ can be defined as Eq. (5):

$$\rho(S) = \frac{\sum_{e \in E(S)} w_e}{|S|} \tag{5}$$

where, $E(S)$ is the edges set of the subgraph S and w_e is the weight of edge e. Moreover, the weighted degree of node $i \in S$ is calculated using Eq. (6):

$$wd_S(i) = \sum_{e_{ij} \in E(S)} w_{e_{ij}} \tag{6}$$

where, e_{ij} is the edge between nodes i and j, and $w_{e_{ij}}$ is the weight of edge e_{ij}.

The procedure of finding initial centers set of the clusters is represented in Fig. 1 (i.e. Algorithm 1). This algorithm starts with a given graph G and computes the current density $\rho(G)$. Then, the candidate nodes that can be removed from the graph are identified based on a threshold value (i.e. $(2 + 2\varepsilon) * \rho(S)$) which the candidate nodes are denoted as $\tilde{A}(S)$. To this end, the nodes with highest weighted degrees are removed which are a portion of $\frac{\varepsilon}{1+\varepsilon} \times |\tilde{A}(S)|$ selected nodes from the candidate list. The algorithm guarantees that at least one of the subgraphs under consideration contains approximately k nodes. Finally, the algorithm proceeds on the remaining graph if the resulted subgraph is non-empty. It should be noted that, the main idea of this step is to find a subgraph with minimum density. Because, this idea guarantees that a centers set with maximum distances between them is found and it can be used as initial centers set of the clustering algorithm. Therefore, the output of this step is a subset of vertices $\tilde{S} \subseteq V$ having a size of at least k with minimum density.

3.1.3. Initial neighbors set calculation for users

In this step, an effective mechanism is applied to calculate the initial neighbors set for the users. To this end, an iterative process is applied on the initial centers set to form a better centers set for the clusters. This step is necessary because the identified initial centers set may not be the best result to the clustering algorithm. Moreover, those of the clusters whose associated members are less than a threshold value will be merged with the other clusters. It should be noted that, the clusters with a small number of users may lead to reduction of the rating prediction accuracy. Therefore, the members of a merged cluster that is a cluster with few members will be assigned to their next nearest centers. The procedure of the initial neighbors set calculation for the users is represented in Fig. 2 (i.e. Algorithm 2). In this algorithm, first of all, each user is assigned to the nearest cluster center based on the initial centers set (i.e. $\tilde{S}$) which is formed by Algorithm 1. The similarity value between a user and a center is used to find the nearest cluster center for the user which this value is calculated using Pearson correlation coefficient function. Then, the new centers of the clusters are calculated based on Lines 5–6 of algorithm 2. In addition, the merging process is performed to merge the clusters which their cardinality is less than the parameter m with the others and to form better clusters for the users (i.e. Lines 7–11). Finally, the members of the resulted clusters are considered as the initial neighbors set of the users that are belonging to their clusters (i.e. Lines 12–15).

Algorithm 2. Initial neighbors set calculation for users.

Inputs: $\boldsymbol{G = (V, E, W)}$, $\boldsymbol{m > 0}$, and initial centers set $\boldsymbol{\tilde{S}}$.
Output: Initial neighbors set for each user.
Algorithm:
1: ***Set*** $\boldsymbol{k' = |\tilde{S}|}$;
2: ***Set*** $\boldsymbol{p_j = \tilde{S}_j}$, $\boldsymbol{\forall j = 1, \ldots, k'}$;
3: ***Let*** $\boldsymbol{p_j}$, $\boldsymbol{\forall j = 1, \ldots, k'}$ be initial center corresponding to **j**-th cluster $\boldsymbol{C_j}$;
4: ***Associate*** each non-selected user to nearest cluster;
5: ***Select*** new centers $\boldsymbol{p'_j = arg\,max_{v_i \in C_j}\, sum(v_i)}$, $\boldsymbol{j = 1, \ldots, k'}$, where
$\boldsymbol{sum(v_i) = \sum_{v_t \in C_j, v_t \neq v_i} w(v_i, v_t)}$;
6: **if** $\boldsymbol{p_j = p'_j}$, $\boldsymbol{\forall j = 1, \ldots, k'}$ **then go to** line 7**, else** $\boldsymbol{p_j = p'_j}$, $\boldsymbol{\forall j = 1, \ldots, k'}$ and **go to** line 4;
7: **for** all $\boldsymbol{C_j}$, $\boldsymbol{j = 1, \ldots, k'}$ **do**
8: **if** $\boldsymbol{|C_j| < m}$ **then**
9: ***Merge*** the members of $\boldsymbol{C_j}$ to other clusters;
10: **end if**
11: **end for**
12: **for** all users $\boldsymbol{u \in V}$ **do**
13: ***Let*** $\boldsymbol{C_u}$ be the cluster that user $\boldsymbol{u}$ belong to;
14: ***Set*** the members of cluster $\boldsymbol{C_u}$ as initial neighbors set for user $\boldsymbol{u}$;
15: **end for**

Fig. 2. The pseudo code of the algorithm for calculating initial neighbors set of the users.

3.2. Adaptive neighbor selection mechanism

In this section, a novel neighbor selection mechanism is proposed to improve the accuracy of the predicted ratings in the prediction process. It should be noted that, in most of the existing recommendation methods the identified initial neighbors set of the users are directly used as the active user's neighbors to predict unseen items without considering the quality of the predicted ratings. However, some of the initial neighbors may not be useful for predicting all of the unseen items. In other words, it is important to infer and refine the identified initial neighbors set of the users to make better recommendations to the users. Therefore, the proposed adaptive neighbor selection mechanism is used to select a subset of the initial neighbors set that is useful to predict an unseen item. To this end, a reliability measure is used to evaluate the quality of the initial predicted ratings. Moreover, a confidence model between the users is proposed to remove useless users from the initial neighbors set of the users. The overall steps of the proposed adaptive neighbor selection mechanism are described in the following subsections.

3.2.1. Calculating reliability measure

In this step, the reliability values of initial predicted ratings are calculated using a reliability measure (Hernando, Bobadilla, Ortega, & Tejedor, 2013). These initial ratings are predicted based on the initial neighbors set of the users which is formed in the initial neighbor selection step of the proposed method (See Section 3.1). Therefore, the initial rating of item i for active user a can be calculated using Eq. (7):

$$P_{a,i} = \overline{r_a} + \frac{\sum_{u \in K_{a,i}} w_{a,u}(r_{u,i} - \overline{r_u})}{\sum_{u \in K_{a,i}} w_{a,u}} \tag{7}$$

where, $\overline{r_a}$ is the average of the ratings for active user a, $K_{a,i}$ refers to a subset of the initial neighbors set of user a that have rated item i, and $w_{a,u}$ represents the similarity weight between the users a and u which is calculated using Eq. (4).

On the other hand, the proposed reliability measure is based on combination of the similarity values and the trust statements between the users which is used to evaluate the quality of the initial predicted ratings. To this end, two positive and negative factors are used to calculate the reliability measure. The overall procedure of calculating this reliability measure is described in the following.

Step 1: The positive factor of the reliability measure is calculated using Eq. (8):

$$f_s(S_{a,i}) = 1 - \frac{\bar{S}}{\bar{S} + S_{a,i}} \tag{8}$$

where, $\bar{S}$ is the median of the values of $S_{a,i}$ in a specific recommender system, and $S_{a,i}$ is the summation of the similarity weights between user a and her/his initial neighbors set which is calculated using Eq. (9):

$$S_{a,i} = \sum_{u \in K_{a,i}} w_{a,u} \tag{9}$$

where, $K_{a,i}$ is a set of neighbors for user a who have rated item i, and $w_{a,u}$ represents the similarity weight between users a and u that is calculated using Eq. (4).

Step 2: The negative factor of the reliability measure is calculated using Eq. (10):

$$f_v(V_{a,i}) = \left(\frac{max - min - V_{a,i}}{max - min}\right)^{\gamma} \tag{10}$$

where,

$$\gamma = \frac{\ln 0.5}{\ln \frac{max-min-\bar{V}}{max-min}} \tag{11}$$

and $\bar{V}$ is the median of the values of $V_{a,\,i}$ in a specific recommender system, max and min are the maximum and minimum values of the ratings for items in a specific recommender system, respectively. Moreover, $V_{a,\,i}$ can be defined as Eq. (12):

$$V_{a,i} = \frac{\sum_{u\in K_{a,i}} w_{a,u}.(r_{u,i} - \overline{r_u} - P_{a,i} + \overline{r_a})^2}{\sum_{u\in K_{a,i}} w_{a,u}} \tag{12}$$

where, $\overline{r_a}$ is the average of ratings for active user *a*, $K_{a,\,i}$ refers to a set of neighbors for user *a* that have rated item *i*, $P_{a,\,i}$ is the initial rating of item *i* for user *a* that is calculated by Eq. (7), and $w_{a,\,u}$ is the similarity weight between users *a* and *u* that is calculated using Eq. (4).

Step 3: Finally, the reliability measure of the initial rating of item *i* for user *a* (i.e. $P_{a,\,i}$) is calculated using Eq. (13):

$$R_{a,i} = \left(f_s(S_{a,i}).f_v(V_{a,i})^{f_s(S_{a,i})}\right)^{\frac{1}{1+f_s(S_{a,i})}} \tag{13}$$

where, $f_s(S_{a,\,i})$ and $f_v(V_{a,\,i})$ are the positive and negative factors of the reliability measure which are calculated using Eqs. (8) and (10), respectively.

3.2.2. Neighborhood adaptation

In the initial rating prediction step (i.e. Eq. (7)), all of the users in the initial neighbors set are used as the nearest neighbors of the active user to predict unseen items. These initial neighbors are determined based on the identified clusters of the users which are calculated in the initial neighbor selection step of the proposed method (see Section 3.1). However, some of the initial neighbors may not be useful to predict some of the unseen items for a given active user. Therefore, a method with ability of evaluating the initial predicted ratings can be useful to recommend the efficient items to the active user.

In this step, an adaptive neighbor selection mechanism is proposed which is based on the proposed reliability measure to improve the accuracy of the initial predicted ratings. Therefore, the reliability measure is used to evaluate the quality of the initial predicted ratings of the unseen items. To this end, the reliability measure $R_{a,\,i}$ for the active user *a* and the item *i* is calculated using Eq. (13) and if its value is lower than a threshold value (r), the initial neighbors set of the active user will be replaced with a new set of neighbors. Then, this adapted set of the neighbors is used to predict a new rating for the initial predicted rating. To calculate a new adapted set of the neighbors for the active user *a* and the item *i*, a confidence model between the users is proposed which is based on the proposed reliability measure (i.e. Eq. (13)). The proposed confidence model between the users can be calculated using Eq. (14):

$$C_{a,v} = \frac{\sum_{i\in A_{a,v}} R_{a,i}(r_i(a) - \bar{r}(a))R_{v,i}(r_i(v) - \bar{r}(v))}{\sqrt{\sum_{i\in A_{a,v}} R^2_{a,i}(r_i(a) - \bar{r}(a))^2}\sqrt{\sum_{i\in A_{a,v}} R^2_{v,i}(r_i(v) - \bar{r}(v))^2}} \tag{14}$$

where, $C_{a,\,v}$ is the confidence value between users *a* and *v*, $R_{a,\,i}$ is the reliability measure for the active user *a* and the item *i* which is calculated using Eq. (13), $r_i(a)$ is the rate of item *i* given by user *a*, $\bar{r}(a)$ is the average of the rates given by user *a*, and $A_{a,\,v}$ is the set of items which are rated by both of the users *a* and *v*.

After calculating the confidence values between the active user *a* and the users of her/his initial neighbors set, the useless users are removed from the initial neighbors set of the user *a*. To this end, a threshold value (θ) is used to identify those of the useless users. In other words, the user *v* will be removed from the initial neighbors set of the active user *a*, if the confidence value $C_{a,\,v}$ is lower than the threshold θ. Therefore, the initial neighbors set of the users is adapted based on the proposed confidence model between the users for the initial predicted ratings that their reliability measures are less than a threshold value. In other words, the proposed neighbor selection mechanism makes to form adaptive neighbors set for the users which leads to improve the accuracy and efficiency of the prediction process.

3.3. Recommendation

In the neighborhood adaptation step of the proposed method (see Section 3.2.2), a new set of the neighbors for the active user is obtained which has a higher performance than the initial neighbors set. In the recommendation step, the new

Algorithm 3. Social Recommendation based on an Adaptive Neighbor Selection (SRANS).

Inputs: Parameters $\boldsymbol{r}$, $\boldsymbol{\theta}$, and $\boldsymbol{top\_N}$.
Output: Top-N recommendation list.
Algorithm:

Split dataset into train set $\boldsymbol{Tr}$ and test set $\boldsymbol{Te}$;
Map all users to a graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{W})$ based on the train set $\boldsymbol{Tr}$ by using Eqs. (1)-(4);
Apply Algorithm 1 on graph $\boldsymbol{G}$ to find initial centers set of the clusters;
Apply Algorithm 2 on graph $\boldsymbol{G}$ to calculate initial neighbors set for users;
for all $\boldsymbol{r_{a,i}} \in \boldsymbol{Te}$ **do**
    Predict the initial rating $\boldsymbol{P_{a,i}}$ of the item $\boldsymbol{i}$ for the active user $\boldsymbol{a}$ using Eq. (7);
    Calculate the reliability measure $\boldsymbol{R_{a,i}}$ of the item $\boldsymbol{i}$ for the active user $\boldsymbol{a}$ using Eq. (13);
end for
for all $\boldsymbol{R_{a,i}} \in \boldsymbol{Te}$ **do**
    if $(\boldsymbol{R_{a,i}} < \boldsymbol{r})$ **then**
        Let $\boldsymbol{K_a}$ be the initial neighbors set for user $\boldsymbol{a}$;
        Let $\boldsymbol{K_a^{new}}$ be the adapted neighbors set for user $\boldsymbol{a}$;
        Set $\boldsymbol{K_a^{new}} = \boldsymbol{K_a}$;
        for all $\boldsymbol{v} \in \boldsymbol{K_a^{new}}$ **do**
            Calculate confidence value between users $\boldsymbol{a}$ and $\boldsymbol{v}$ using Eq. (14);
            if $\left(\boldsymbol{C_{a,v}} < \boldsymbol{\theta}\right)$ **then**
                Set $\boldsymbol{K_a^{new}} = \boldsymbol{K_a^{new}} - \{\boldsymbol{v}\}$;
            end if
        end for
        Predict the new rating $\boldsymbol{P_{a,i}^{new}}$ based on the new set of neighbors $\boldsymbol{K_a^{new}}$ using Eq. (15);
    end if
end for
Recommend $\boldsymbol{top\_N}$ of items as the recommendation list to the active user $\boldsymbol{a}$;

Fig. 3. The pseudo code of the proposed method.

set of the neighbors is used for the active user to predict the new ratings of the unseen items using Eq. (15):

$$P_{a,i}^{new} = \overline{r_a} + \frac{\sum_{u \in K_{a,i}^{new}} w_{a,u}(r_{u,i} - \overline{r_u})}{\sum_{u \in K_{a,i}^{new}} w_{a,u}} \tag{15}$$

where, $P_{a,i}^{new}$ is the new rating of item i for the active user a, $r_{u,\,i}$ is the rating of item i given by user u, $\overline{r_a}$ is the average of the ratings for the active user a, $K_{a,i}^{new}$ refers to a subset of the adapted neighbors set of user a that have rated item i, and $w_{a,\,u}$ represents the similarity weight between the users a and u which is calculated using Eq. (4). After predicting the new ratings of the unseen items for the active user, the algorithm selects the top_N items as recommendation list to suggest to the active user. The pseudo code of the proposed method is shown in Fig. 3 (i.e. Algorithm 3).

4. Experimental results

In this section, the performance of the proposed method (i.e. SRANS) is evaluated empirically upon three well-known datasets and the results are compared with the other recommendation methods based on different evaluation metrics. The brief descriptions of the compared methods are provided in the following:

- **KCF** is a baseline method which users are clustered based on their historical ratings (i.e. Eq. (1)) by using K-Means clustering algorithm. Moreover, the identified clusters are used as nearest neighbors of the users to predict the ratings of unseen items using collaborative filtering method.
- **TKCF** is a variant of the KCF method which uses the combination of the similarity values and trust statements as the similarity weights between the users (i.e. Eq. (4)) for ratings prediction.
- **TRACCF** is a model-based collaborative filtering method which takes into account both of the clustering and trustworthy information to predict unseen items for the users (Birtolo & Ronca, 2013).
- **IFCCF** is a method that items are clustered based on their obtained ratings by using fuzzy C-Means clustering algorithm. In addition, the resulted clusters are used to predict the ratings of the unseen items for active users (Birtolo & Ronca, 2013).
- **MV** is a multiview K-medoids method which clusters users based on both of the historical ratings and trust information (Guo et al., 2015a).
- **DGCTARS** is a recommendation method which uses a graph clustering algorithm to group users into several clusters and then these identified clusters are considered to predict ratings for active users in recommendation process (Moradi et al., 2015).

- **TrustSVD** is a trust-based matrix factorization technique which involves the explicit and implicit influence of rated items, by further incorporating both of the explicit and implicit influence of trusted users on the prediction of items (Guo et al., 2015b).
- **SocialMF** is a recommendation method based on incorporating trust propagation into a matrix factorization model for providing suggestions in social networks (Jamali & Ester, 2010).
- **TrustMF** is a model-based method which uses matrix factorization technique to map users into low-dimensional latent factors spaces in terms of their trust relationship (Yang, Lei et al., 2013).
- **SRANS** is the proposed method which is based on an adaptive neighbor selection mechanism to provide recommendations to users in social networks.

The description of the datasets used in the experiments, the parameter setting, the evaluation metrics, the performance comparison, the sensitivity analysis of the parameters, and the statistical analysis are presented in the following subsections.

4.1. Datasets

Three well-known datasets are used in the experiments to show the effectiveness of the proposed method. These datasets include Epinions[1], Flixster[2], and FilmTrust[3]. The Epinions dataset has been extracted from Epinions.com website which its users will be able to review items and also assign them numeric ratings in the range 1 (min) to 5 (max). Furthermore, the users can express their web of trust with the other users. The values of the trust statements between the users in this dataset are 0 or 1 which mean the trust relationship between two users exists or not, respectively. The number of users in the Epinions dataset is 49,290 users who rated at least once among 139,738 different items. For simplicity, we randomly sample 10 K users from the original dataset as well as the user ratings and trust information.

The Flixster dataset is a social movie site in which the users have friendship relations with the other users and can share their ratings on the existing movies where are scaled from 0.5 (min) to 4.0 (max) with step 0.5. The values of the trust statements in the Flixster dataset are not available; therefore, we used friend relationships between the users as the trust statements. In the experiments, we sampled a subset of the Flixster dataset by randomly choosing 10 K users with their corresponding ratings on the items and the trust statements.

Finally, the FilmTrust dataset is a social site which its users are able to rate the existing movies. The values of the ratings are in the range of 0.5 (min) to 4.0 (max) with step 0.5. Moreover, the values of the trust statements are in the range of 1 to 10 which are not available due to the sharing policy. Therefore, in the experiments we used the link information between the users as the trust statements. This dataset consists of 1986 users, 2071 movies, 35,497 movie ratings, and 1853 trust ratings (Guo et al., 2014).

4.2. Parameter setting

In the proposed method, several parameters need to be initialized in accordance with the compared methods. These parameters are r, θ, and top_N which respectively denote the reliability threshold, the confidence threshold, and the size of the recommendation list. It should be noted that, the parameters r and θ are used as the threshold values in the proposed method for the calculated reliability and confidence measures, respectively (see Section 3.2.2). In the experiments, the values of the parameters r, θ, and top_N are set to $r = 0.7$, $\theta = 0.6$, and $top\_N = 5, 10, 15$ for all datasets. The parameters k and ε are used in Algorithm 1 and we used the values $k = 5$ and $\varepsilon = 1$ as default values which give good results in general. In addition, the parameter m is used in Algorithm 2 and we used the values $m = 30$, $m = 40$, and $m = 30$ for Flixster, Epinions, and FilmTrust datasets, respectively. For the rest of the methods in the experiments, there are parameters to be set. To make a fair comparison, the values of these parameters are set based on the optimal values which are reported in their corresponding papers.

4.3. Evaluation metrics

The conventional leave-one-out procedure is used in current work to compare the performance of the proposed method with the other methods (Massa & Avesani, 2007). To this end, the 5-fold cross-validation approach is applied for comparing the results of the recommendation methods. In other words, each dataset is divided into five folds and in each run four folds are used as the train set and the remaining fold as the test set. Five runs are performed for testing all of the folds and the average results of these runs are reported as the final result. In the leave-one-out method, a rated review in the test set is used as an unseen item and a rating is predicted for it by using the information about the remaining rated reviews. Then, the predicted rating is compared with the real rating and the difference between them is computed as prediction error. It should be noted that, the procedure is repeated for all ratings in the test set to compute evaluation metrics. In

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

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

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

Table 1
Experiment results on the *Epinions* dataset for MAE measure, different neighborhood sizes (N), and 5-fold cross validation.

Algorithms	All users					Cold users				
	N = 20	N = 40	N = 60	N = 80	N = 100	N = 20	N = 40	N = 60	N = 80	N = 100
KCF	1.167	1.144	1.098	1.082	1.057	1.384	1.312	1.297	1.253	1.239
TKCF	1.142	1.103	1.086	1.069	1.054	1.293	1.276	1.249	1.225	1.208
TRACCF	1.025	1.002	0.984	0.972	0.956	1.175	1.159	1.143	1.128	1.107
IFCCF	0.986	0.963	0.941	0.923	0.897	1.108	1.087	1.052	1.034	1.009
MV	1.003	0.995	0.973	0.957	0.929	1.096	1.082	1.041	1.017	0.995
DGCTARS	0.954	0.932	0.916	0.885	0.864	0.965	0.952	0.934	0.918	0.896
TrustSVD	0.866	0.839	0.817	0.805	0.793	0.898	0.882	0.864	0.845	0.838
SocialMF	0.921	0.908	0.887	0.862	0.839	0.943	0.927	0.895	0.887	0.864
TrustMF	0.893	0.871	0.856	0.839	0.817	0.928	0.905	0.876	0.862	0.849
SRANS	**0.772**	**0.743**	**0.698**	**0.672**	**0.664**	**0.813**	**0.786**	**0.753**	**0.742**	**0.735**

the experiments, several evaluation metrics including mean absolute error (MAE), root mean square error (RMSE), Precision, Recall, and F1 are used to evaluate the proposed method.

The MAE and RMSE measures are two metrics to evaluate the accuracy of the predicted ratings. To this end, a predicted rating is compared with the real rating and the difference between them is considered as prediction error. This process is repeated for all of the predicted ratings and then an average of all values is computed as the final prediction error. The MAE and RMSE metrics are calculated respectively using Eqs. (16) and (17):

$$MAE = \frac{\sum_{i=1}^{N} |r_i - p_i|}{N} \tag{16}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(r_i - p_i)^2} \tag{17}$$

where, r_i and p_i are respectively actual and predicted ratings of item i. Also, N denotes the total number of ratings that are predicted by a recommendation method.

On the other hand, to compute the Precision, Recall, and F1 metrics, the ratings are divided into a binary scale including relevant and non-relevant items (Herlocker, Konstan, Terveen, & Riedl, 2004). The Epinions dataset provides 5-point scale ratings. Furthermore, the Flixster and FilmTrust datasets consist of ratings in the range of 0.5 to 4.0 with step 0.5. Therefore, in the experiments, the rating values which are larger than 3 (i.e. $r_{u,i} \geq 3$) are considered as relevant items and the remaining ones are considered as non-relevant items for all of the used datasets. The Precision metric is defined as a measure of exactness or accuracy which is computed as the fraction of the relevant items retrieved out of all items using Eq. (18):

$$Precision = \frac{|relevant\ items\ recommended|}{|all\ items\ retrieved\ and\ recommended|} \tag{18}$$

Moreover, the Recall metric is a measure of completeness which is computed as the fraction of relevant items retrieved out of all relevant items that were retrieved or not. This metric is defined as Eq. (19):

$$Recall = \frac{|relevant\ items\ recommended|}{|all\ relevant\ items\ retrieved\ and\ not\ recommended|} \tag{19}$$

It should be noted that, the Precision and Recall metrics are clearly conflicting in nature (Herlocker et al., 2004). In other words, if the number of *top_N* recommendations increases then the number of relevant items and also the Recall metric will be increased. On the other hand, at the same time the Precision measure decreases. To obtain an appropriate weighted combination of the Precision and Recall measures, the F1 metric is defined as Eq. (20):

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{20}$$

4.4. Performance comparison

In this section, a series of experiments are performed on three real-world datasets to demonstrate the performance of the proposed method relative to others based on two views of data including: *All users* and *Cold users*. In the *All users* view, we used all of the data in datasets while in the *Cold users* view, the cold start users with lower than 5 ratings are considered to evaluate the recommendation methods. Tables 1–6 show the results of the comparison between the proposed SRANS method and the other state-of-the-art methods based on the MAE and RMSE measures. It should be noted that, these results are based on different neighborhood sizes (i.e. $N = 20, 40, 60, 80, 100$) for all of the recommendation methods. In the TrustSVD, SocialMF, and TrustMF methods, we used trust networks of the users to form neighborhood sets with different

Table 2
Experiment results on the *Epinions* dataset for RMSE measure, different neighborhood sizes (N), and 5-fold cross validation.

Algorithms	All users					Cold users				
	N = 20	N = 40	N = 60	N = 80	N = 100	N = 20	N = 40	N = 60	N = 80	N = 100
KCF	1.594	1.571	1.537	1.506	1.487	1.653	1.645	1.627	1.612	1.596
TKCF	1.564	1.529	1.509	1.488	1.472	1.634	1.632	1.625	1.609	1.593
TRACCF	1.343	1.317	1.294	1.265	1.228	1.372	1.346	1.325	1.297	1.245
IFCCF	1.243	1.201	1.189	1.154	1.125	1.287	1.248	1.216	1.185	1.159
MV	1.275	1.253	1.231	1.207	1.192	1.295	1.274	1.253	1.219	1.216
DGCTARS	1.119	1.107	1.098	1.069	1.055	1.272	1.221	1.195	1.163	1.151
TrustSVD	1.113	1.101	1.087	1.053	1.041	1.233	1.188	1.165	1.127	1.123
SocialMF	1.217	1.189	1.146	1.115	1.072	1.258	1.208	1.182	1.153	1.146
TrustMF	1.134	1.112	1.105	1.086	1.069	1.241	1.201	1.174	1.138	1.129
SRANS	**1.052**	**1.033**	**1.012**	**1.002**	**0.993**	**1.083**	**1.065**	**1.042**	**1.027**	**1.018**

Table 3
Experiment results on the *Flixster* dataset for MAE measure, different neighborhood sizes (N), and 5-fold cross validation.

Algorithms	All users					Cold users				
	N = 20	N = 40	N = 60	N = 80	N = 100	N = 20	N = 40	N = 60	N = 80	N = 100
KCF	1.195	1.146	1.105	1.038	0.996	1.242	1.218	1.175	1.163	1.148
TKCF	1.167	1.115	1.063	0.998	0.965	1.219	1.183	1.161	1.147	1.127
TRACCF	1.183	1.124	1.085	1.001	0.979	1.201	1.178	1.153	1.125	1.094
IFCCF	1.144	1.098	1.057	0.953	0.924	1.196	1.172	1.146	1.098	1.052
MV	1.135	1.072	1.004	0.946	0.912	1.192	1.155	1.108	1.062	1.014
DGCTARS	1.128	1.012	0.973	0.885	0.857	1.188	1.143	1.102	1.031	0.985
TrustSVD	1.097	0.963	0.875	0.817	0.798	1.172	1.105	1.031	0.964	0.913
SocialMF	1.109	0.985	0.916	0.852	0.815	1.175	1.112	1.053	0.987	0.945
TrustMF	1.131	1.055	0.997	0.931	0.897	1.183	1.119	1.078	1.021	0.976
SRANS	**0.975**	**0.883**	**0.791**	**0.748**	**0.709**	**1.007**	**0.941**	**0.885**	**0.843**	**0.805**

Table 4
Experiment results on the *Flixster* dataset for RMSE measure, different neighborhood sizes (N), and 5-fold cross validation.

Algorithms	All users					Cold users				
	N = 20	N = 40	N = 60	N = 80	N = 100	N = 20	N = 40	N = 60	N = 80	N = 100
KCF	1.395	1.345	1.312	1.304	1.293	1.468	1.432	1.397	1.371	1.367
TKCF	1.376	1.342	1.305	1.271	1.258	1.445	1.415	1.392	1.364	1.353
TRACCF	1.353	1.324	1.287	1.252	1.235	1.376	1.359	1.317	1.285	1.258
IFCCF	1.294	1.275	1.246	1.203	1.187	1.344	1.306	1.262	1.237	1.216
MV	1.253	1.228	1.197	1.185	1.164	1.352	1.314	1.287	1.255	1.232
DGCTARS	1.217	1.193	1.164	1.136	1.125	1.276	1.249	1.213	1.194	1.175
TrustSVD	1.175	1.129	1.075	1.004	0.975	1.203	1.176	1.134	1.117	1.095
SocialMF	1.196	1.154	1.105	1.047	0.996	1.221	1.197	1.168	1.149	1.126
TrustMF	1.242	1.217	1.182	1.168	1.152	1.255	1.228	1.211	1.191	1.173
SRANS	**0.995**	**0.963**	**0.941**	**0.925**	**0.897**	**1.104**	**1.041**	**1.007**	**1.003**	**0.998**

sizes. Tables 1 and 2 reveal that the proposed method obtains the best results in terms of the MAE and RMSE measures for the *All users* and *Cold users* views compared to the other methods on the Epinions dataset. For example, the SRANS method obtains the MAE value 0.772 and the RMSE value 1.052 for the *All users* view and the MAE value 0.813 and the RMSE value 1.083 for the *Cold users* view based on $N = 20$ while in the same case the second best method (i.e. TrustSVD method) obtains the MAE value 0.866 and the RMSE value 1.113 for the *All users* view and the MAE value 0.898 and the RMSE value 1.233 for the *Cold users* view.

Furthermore, the experiments based on the MAE and RMSE measures are repeated on the Flixster and FilmTrust datasets and the corresponding results are shown in Tables 3–6. The results of Tables 3 and 4 illustrate that the proposed method outperforms the other methods under both of the MAE and RMSE measures for the *All users* and *Cold users* views. In addition, Tables 5 and 6 show the results of the experiments for the MAE and RMSE measures on the FilmTrust dataset. It can be observed from Table 5 that, the proposed method obtains better performance than the other recommendation methods except for the TrustSVD method which obtains the best MAE values for the *All users* view. In these cases, the proposed method obtains the second best MAE values. Moreover, the results of Table 6 show that, the proposed method obtains the third best RMSE values compared to the other methods for the *All users* view. However, the results of the Tables 5 and 6 also reveal that the proposed method acquires the better MAE and RMSE values compared to the other methods for the *Cold users* view.

Table 5
Experiment results on the *FilmTrust* dataset for MAE measure, different neighborhood sizes (N), and 5-fold cross validation.

Algorithms	All users					Cold users				
	N = 20	N = 40	N = 60	N = 80	N = 100	N = 20	N = 40	N = 60	N = 80	N = 100
KCF	0.965	0.897	0.824	0.776	0.745	1.069	1.002	0.945	0.923	0.889
TKCF	0.983	0.912	0.878	0.816	0.759	1.086	1.021	0.964	0.937	0.895
TRACCF	0.854	0.808	0.751	0.685	0.648	1.024	0.955	0.902	0.846	0.803
IFCCF	0.789	0.774	0.721	0.668	0.637	0.942	0.896	0.828	0.781	0.736
MV	0.931	0.875	0.819	0.752	0.734	1.008	0.951	0.895	0.842	0.798
DGCTARS	0.815	0.795	0.739	0.673	0.643	0.997	0.928	0.862	0.819	0.764
TrustSVD	**0.769**	**0.748**	**0.693**	**0.637**	**0.611**	0.896	0.853	0.806	0.761	0.706
SocialMF	0.876	0.823	0.765	0.698	0.651	0.965	0.907	0.845	0.784	0.742
TrustMF	0.837	0.797	0.746	0.679	0.645	0.923	0.882	0.821	0.774	0.715
SRANS	0.775	0.752	0.706	0.643	0.614	**0.872**	**0.805**	**0.753**	**0.698**	**0.662**

Table 6
Experiment results on the *FilmTrust* dataset for RMSE measure, different neighborhood sizes (N), and 5-fold cross validation.

Algorithms	All users					Cold users				
	N = 20	N = 40	N = 60	N = 80	N = 100	N = 20	N = 40	N = 60	N = 80	N = 100
KCF	1.118	1.045	1.008	0.976	0.941	1.289	1.242	1.191	1.148	1.094
TKCF	1.134	1.088	1.021	0.986	0.954	1.294	1.243	1.196	1.154	1.105
TRACCF	1.009	0.997	0.954	0.918	0.881	1.265	1.236	1.175	1.112	1.067
IFCCF	0.992	0.938	0.891	0.857	0.839	1.184	1.147	1.034	0.978	0.942
MV	1.125	1.063	1.014	0.982	0.949	1.218	1.164	1.095	1.022	0.987
DGCTARS	1.002	0.991	0.943	0.906	0.875	1.242	1.201	1.164	1.108	1.016
TrustSVD	**0.986**	**0.927**	**0.884**	**0.851**	**0.827**	1.141	1.108	1.005	0.953	0.912
SocialMF	1.103	1.007	0.962	0.923	0.885	1.206	1.152	1.063	1.002	0.965
TrustMF	0.998	0.952	0.916	0.891	0.863	1.158	1.125	1.011	0.962	0.931
SRANS	0.995	0.942	0.898	0.863	0.844	**1.056**	**1.004**	**0.965**	**0.919**	**0.862**

Table 7
Precision, Recall, and F1 evaluation on the *Epinions* dataset for all users, different values of *top_N*, and 5-fold cross validation.

Algorithms	Precision			Recall			F1		
	P@5	P@10	P@15	R@5	R@10	R@15	F1@5	F1@10	F1@15
KCF	0.935	0.925	0.915	0.551	0.612	0.691	0.693	0.737	0.787
TKCF	0.939	0.923	0.912	0.563	0.632	0.665	0.704	0.751	0.769
TRACCF	0.954	0.943	0.935	0.582	0.646	0.712	0.723	0.767	0.808
IFCCF	0.971	0.957	0.952	0.675	0.718	0.786	0.796	0.821	0.861
MV	0.968	0.948	0.945	0.658	0.685	0.748	0.783	0.795	0.835
DGCTARS	0.951	0.949	0.942	0.751	0.788	0.832	0.839	0.861	0.884
TrustSVD	0.975	0.961	0.958	0.776	0.831	0.872	0.864	0.891	0.913
SocialMF	0.958	0.946	0.939	0.758	0.809	0.847	0.846	0.872	0.891
TrustMF	0.969	0.954	0.949	0.769	0.827	0.859	0.857	0.886	0.902
SRANS	**0.981**	**0.975**	**0.966**	**0.788**	**0.852**	**0.887**	**0.874**	**0.909**	**0.925**

In addition, the results of the experiments based on the Precision, Recall, and F1 measures are reported in the Tables 7–10 for the Epinions, Flixster, and FilmTrust datasets. These experiments are performed based on different values of parameter *top_N* (i.e. $top\_N = 5, 10, 15$) for the *All users* view. On the other hand, the experiments are performed based on $top\_N = 5$ for the *Cold users* view, because these users have not rated more than 5 items. Table 7 reports the results of the experiments based on the Precision, Recall, and F1 measures on the Epinions dataset for the *All users* view. As you can see from these results, the proposed method outperforms the other recommendation methods based on all of the Precision, Recall, and F1 measures and all values of the parameter *top_N*. These experiments are repeated for the Flixster and FilmTrust datasets and the results are respectively reported in the Tables 8 and 9. As it is clear from these results, the proposed method can significantly obtain better performance than the other methods based on the Precision, Recall, and F1 measures and also different values of the parameter *top_N* for the *All users* view. Moreover, the experiments are performed for the *Cold users* view and the results are shown in Table 10 for all of the Epinions, Flixster, and FilmTrust datasets. The results of the Table 10 show that the proposed method can be effective for the *Cold users* view based on the Precision, Recall, and F1 measures and also can obtain significant results in comparison to the other methods.

Table 8
Precision, Recall, and F1 evaluation on the *Flixster* dataset for all users, different values of *top_N*, and 5-fold cross validation.

Algorithms	Precision			Recall			F1		
	P@5	P@10	P@15	R@5	R@10	R@15	F1@5	F1@10	F1@15
KCF	0.825	0.812	0.775	0.684	0.697	0.715	0.748	0.751	0.744
TKCF	0.848	0.819	0.786	0.716	0.728	0.743	0.776	0.771	0.764
TRACCF	0.876	0.851	0.827	0.738	0.756	0.772	0.801	0.801	0.799
IFCCF	0.913	0.894	0.863	0.765	0.783	0.798	0.832	0.835	0.829
MV	0.927	0.915	0.881	0.771	0.789	0.815	0.842	0.847	0.847
DGCTARS	0.935	0.926	0.892	0.798	0.829	0.854	0.861	0.875	0.873
TrustSVD	0.952	0.938	0.921	0.813	0.842	0.863	0.877	0.887	0.891
SocialMF	0.946	0.932	0.913	0.809	0.831	0.846	0.872	0.879	0.878
TrustMF	0.941	0.929	0.904	0.805	0.825	0.837	0.868	0.874	0.869
SRANS	**0.964**	**0.942**	**0.934**	**0.821**	**0.852**	**0.878**	**0.887**	**0.895**	**0.905**

Table 9
Precision, Recall, and F1 evaluation on the *FilmTrust* dataset for all users, different values of *top_N*, and 5-fold cross validation.

Algorithms	Precision			Recall			F1		
	P@5	P@10	P@15	R@5	R@10	R@15	F1@5	F1@10	F1@15
KCF	0.828	0.807	0.765	0.738	0.755	0.782	0.780	0.781	0.773
TKCF	0.811	0.809	0.791	0.753	0.778	0.819	0.781	0.793	0.805
TRACCF	0.825	0.815	0.786	0.816	0.839	0.864	0.821	0.827	0.823
IFCCF	0.849	0.834	0.815	0.834	0.861	0.886	0.841	0.847	0.849
MV	0.834	0.819	0.798	0.827	0.857	0.873	0.831	0.838	0.834
DGCTARS	0.839	0.827	0.801	0.866	0.896	0.915	0.852	0.861	0.854
TrustSVD	0.851	0.829	0.823	0.857	0.885	0.921	0.854	0.856	0.869
SocialMF	0.842	0.812	0.806	0.846	0.852	0.881	0.844	0.832	0.842
TrustMF	0.845	0.824	0.812	0.851	0.873	0.892	0.848	0.848	0.851
SRANS	**0.859**	**0.838**	**0.832**	**0.874**	**0.901**	**0.938**	**0.866**	**0.869**	**0.882**

Table 10
Precision, Recall, and F1 evaluation on the *Epinions, Flixster*, and *FilmTrust* datasets for cold start users, $top\_N = 5$, and 5-fold cross validation.

Algorithms	Epinions			Flixster			FilmTrust		
	P@5	R@5	F1@5	P@5	R@5	F1@5	P@5	R@5	F1@5
KCF	0.743	0.565	0.642	0.625	0.553	0.587	0.631	0.706	0.666
TKCF	0.756	0.572	0.651	0.641	0.586	0.612	0.645	0.752	0.694
TRACCF	0.769	0.593	0.669	0.662	0.628	0.645	0.662	0.761	0.708
IFCCF	0.812	0.684	0.743	0.713	0.665	0.688	0.698	0.805	0.748
MV	0.786	0.647	0.709	0.726	0.683	0.704	0.671	0.781	0.722
DGCTARS	0.795	0.746	0.769	0.742	0.697	0.719	0.684	0.842	0.755
TrustSVD	0.828	0.769	0.797	0.759	0.739	0.749	0.712	0.858	0.778
SocialMF	0.804	0.721	0.761	0.754	0.721	0.737	0.678	0.826	0.745
TrustMF	0.819	0.752	0.784	0.749	0.718	0.733	0.689	0.847	0.759
SRANS	**0.841**	**0.782**	**0.811**	**0.763**	**0.756**	**0.759**	**0.731**	**0.864**	**0.792**

4.5. Sensitivity analysis of the parameters

In this section, several experiments are performed to show the effect of different values of the input parameters on the performance of the proposed method. The parameter r is an important parameter which is used in the neighborhood adaptation step of the proposed method (see Section 3.2.2) as the threshold value for the calculated reliability measure. Fig. 4 shows the results of different values of the parameter r on the performance of the proposed method over MAE and RMSE measures for both of the *All users* and *Cold users* views and also for the Epinions, Flixster, and FilmTrust datasets. As it is clear from this figure, the MAE and RMSE measures for the proposed method are decreased when the value of the parameter r is increased from 0 to 0.9 with step 0.1. Therefore, it can be concluded that the higher value of the parameter r leads to improve the performance of the proposed method on the accuracy of the rating prediction process for both of the *All users* and *Cold users* views.

Moreover, the effect of different values of the parameter r on the performance of the proposed method is evaluated based on the Precision, Recall, and F1 measures and the results are reported in Fig. 5. As you can see from these results, the values of the parameter r are increased from 0 to 0.9 with step 0.1 and it leads to increase the values of the Precision, Recall, and

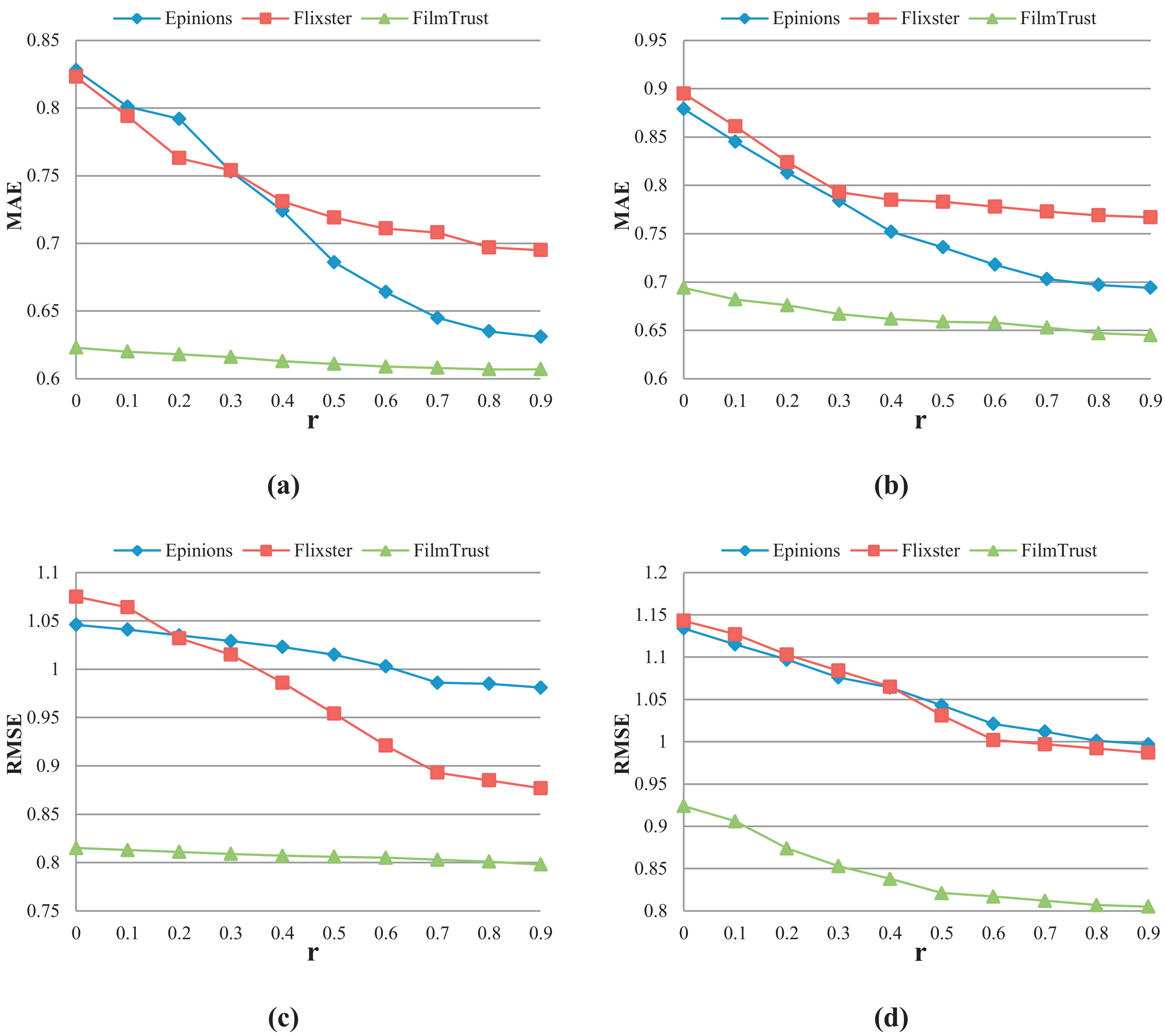

Fig. 4. The effect of parameter r on the system performance: (a) MAE for All users, (b) MAE for Cold users, (c) RMSE for All users, (d) RMSE for Cold users.

F1 measures for the proposed method on the Epinions, Flixster, and FilmTrust datasets. Therefore, it can be conducted that, the higher values of the parameter r make to improve the performance of the proposed method for both of the *All users* and *Cold users* views based on the Precision, Recall, and F1 measures. It should be noted that, the value of the parameter top_N is set to $top\_N = 10$ and $top\_N = 5$ in these experiments for the *All users* and *Cold users* views, respectively.

Another parameter of the proposed method is the parameter θ which can be effective on the accuracy of the rating prediction process. This parameter is used in the neighborhood adaptation step of the proposed method as the threshold value of the proposed confidence model (see Section 3.2.2). In other words, the parameter θ is used to remove those of the useless users from the initial neighbors set of the active user and calculate new adapted neighbors set to predict a new rating for the target item. The effect of different values of the parameter θ is evaluated based on the used evaluation measures for the Epinions, Flixster, and FilmTrust datasets. Fig. 6 reports the results of the performed experiments for different values of the parameter θ based on the MAE and RMSE measures and both of the *All users* and *Cold users* views. As it is shown in this figure, the values of the MAE and RMSE measures are decreased when the value of θ is increased from 0 to 0.9 with step 0.1 for the *All users* and *Cold users* views. It can be concluded from these results that, the higher values of the parameter θ have positive effects on the accuracy measures of the predicted ratings in the proposed method. On the other hand, Fig. 7 shows the effect of different values of the parameter θ on the Precision, Recall, and F1 measures for the *All users* and *Cold users* views and also for the Epinions, Flixster, and FilmTrust datasets. The value of the parameter top_N is set to $top\_N = 10$ and $top\_N = 5$ in these experiments for the *All users* and *Cold users* views, respectively. Based on these results, the values of the Precision, Recall, and F1 measures are increased when the value of the parameter θ is increased.

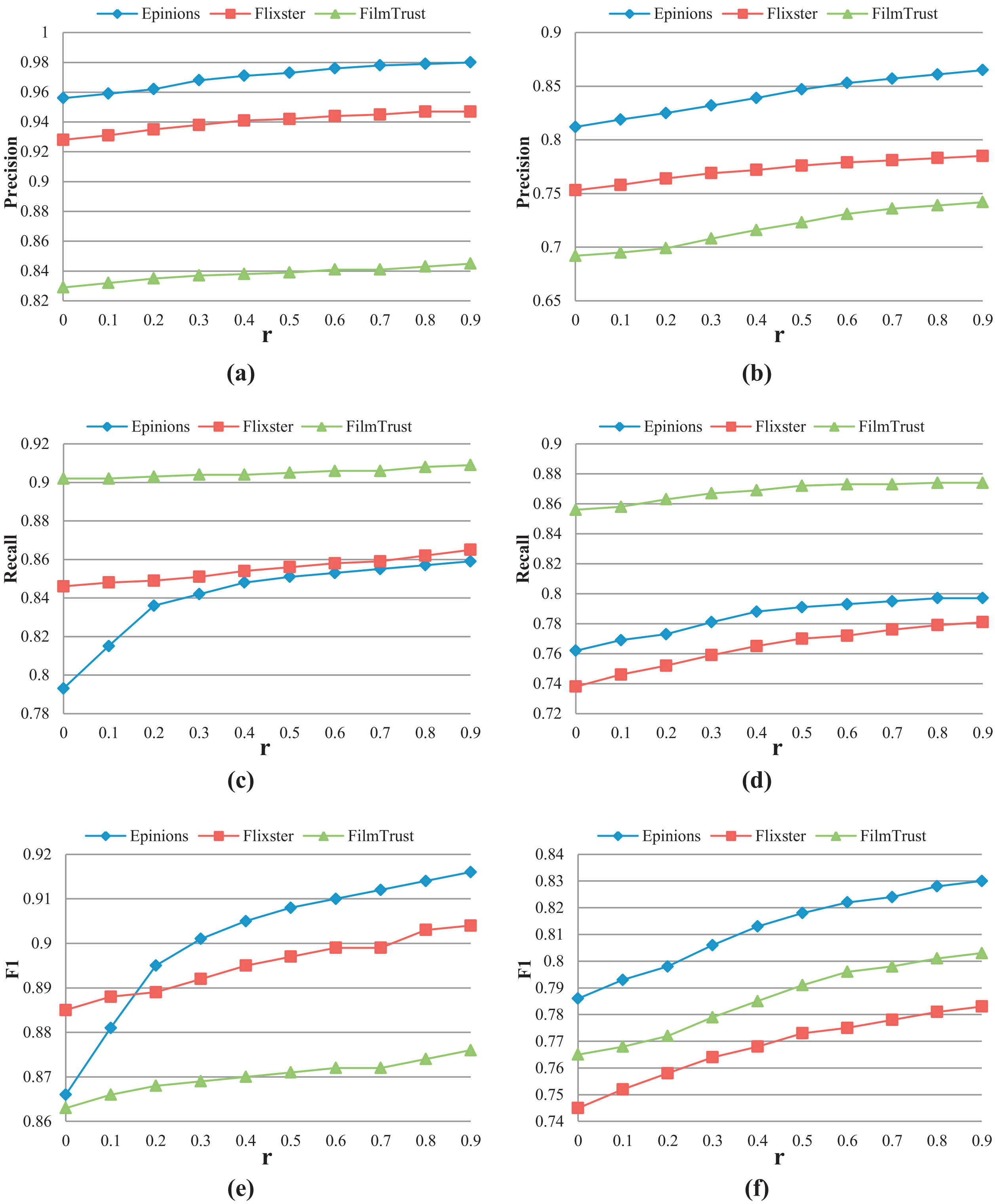

Fig. 5. The effect of parameter *r* on the system performance: (a) Precision for All users, (b) Precision for Cold users, (c) Recall for All users, (d) Recall for Cold users, (e) F1 for All users, (f) F1 for Cold users.

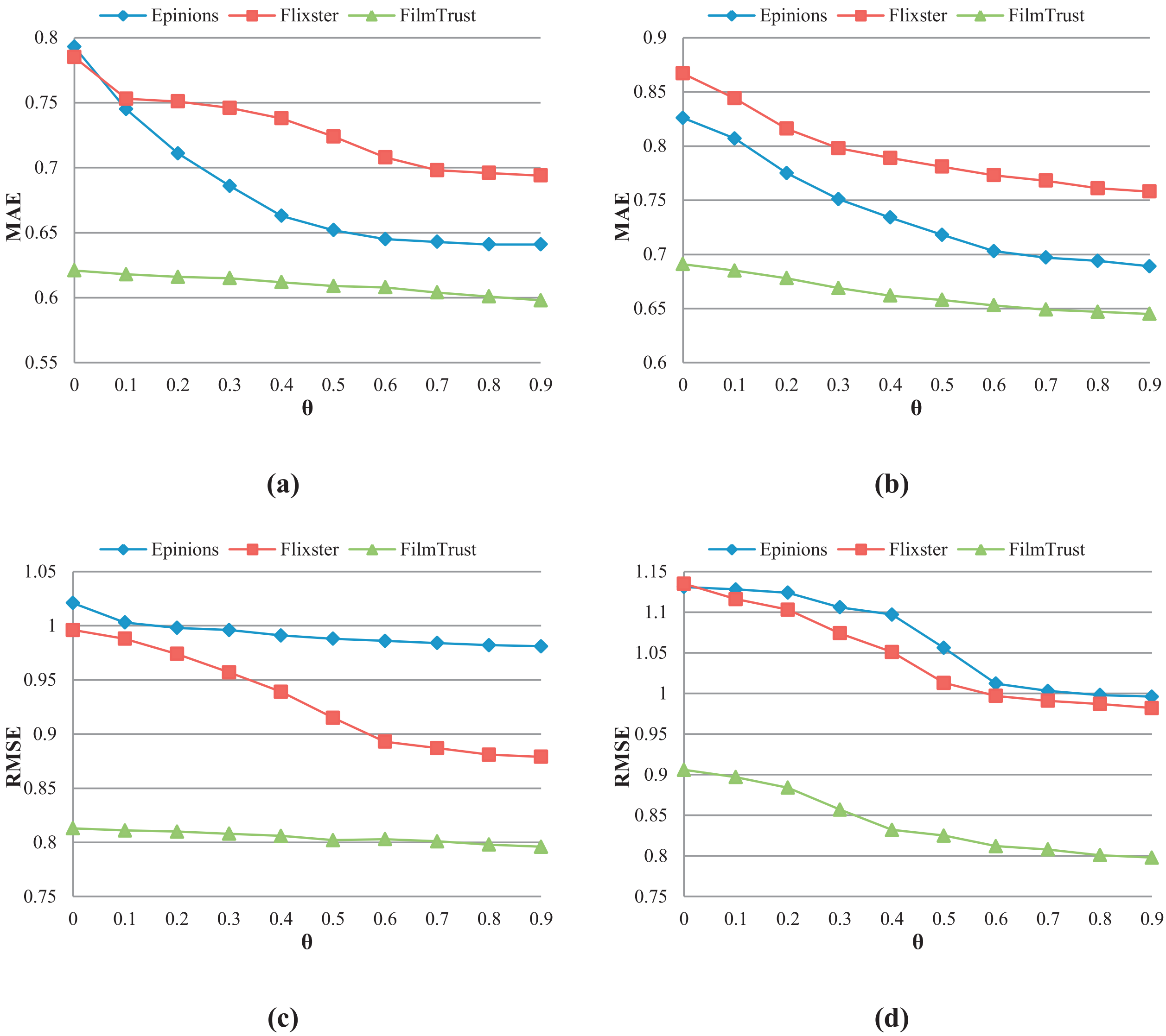

Fig. 6. The effect of parameter θ on the system performance: (a) MAE for All users, (b) MAE for Cold users, (c) RMSE for All users, (d) RMSE for Cold users.

Therefore, the higher values of the parameter θ have positive effects on the performance of the proposed method based on the mentioned evaluation measures for both of the *All users* and *Cold users* views.

4.6. Statistical analysis

In this section, a statistical significance test is used to determine that the experimental results are statistically significant. To this end, the Friedman test (Friedman, 1937) has been performed on the results which is a non-parametric test used to measure the statistical differences of methods over multiple datasets. Therefore, the recommendation methods are ranked separately based on different evaluation measures for each dataset which the method with the best result gets rank 1, the second best result gets rank 2, and so on. The Friedman test is calculated using Eq. (21):

$$F_F = \frac{(q-1)\chi_F^2}{q(t-1)-\chi_F^2} \tag{21}$$

where,

$$\chi_F^2 = \frac{12q}{t(t+1)}\left[\sum_{j=1}^{t} R_j^2 - \frac{t(t+1)^2}{4}\right] \tag{22}$$

and q is the number of datasets, t is the number of methods, and R_j is the average rank of the jth method over all datasets.

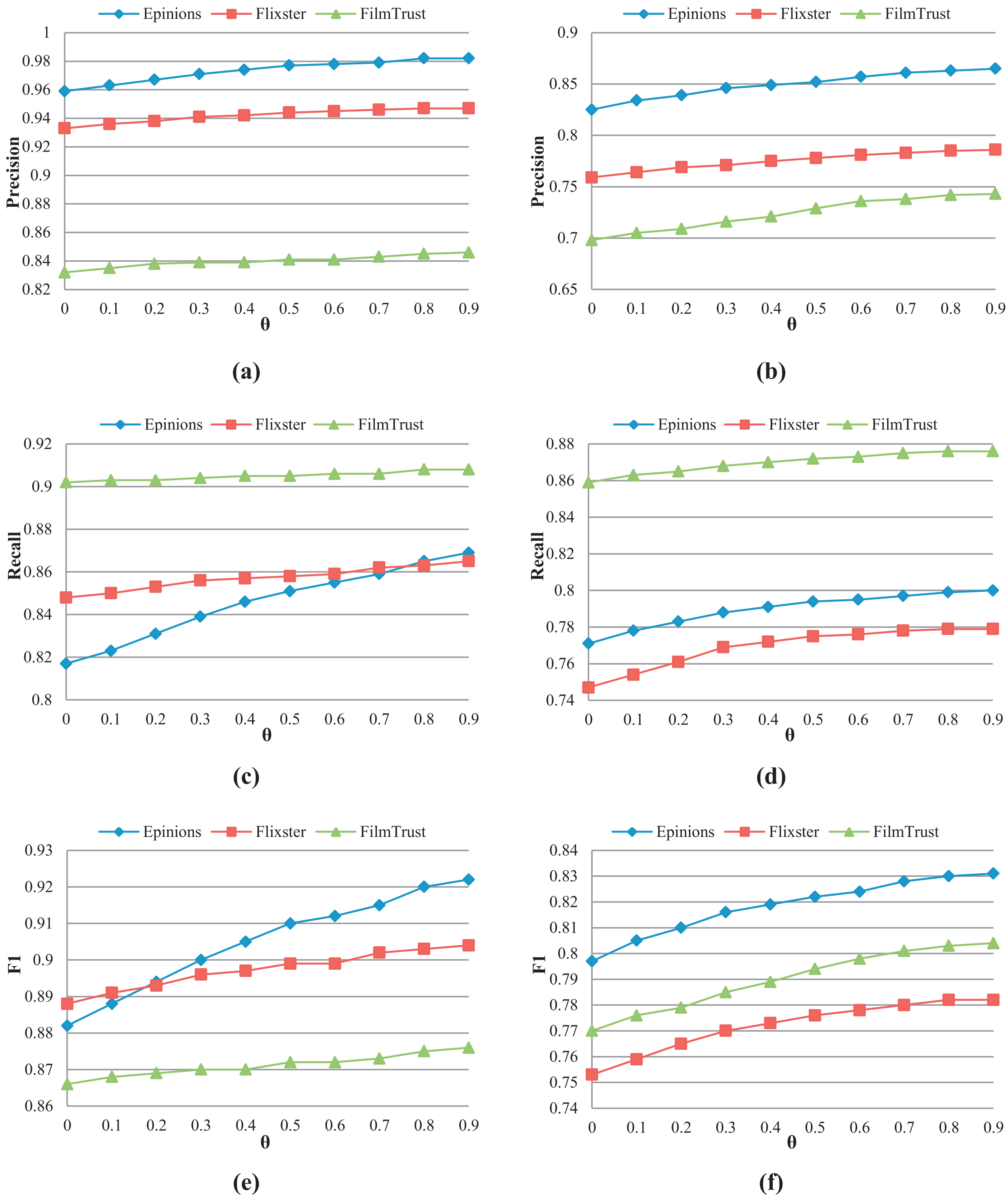

Fig. 7. The effect of parameter θ on the system performance: (a) Precision for All users, (b) Precision for Cold users, (c) Recall for All users, (d) Recall for Cold users, (e) F1 for All users, (f) F1 for Cold users.

Table 11
The results of Friedman test for the comparisons between recommendation methods based on different evaluation measures.

Measures	χ_F^2	F_F	$F(9, 18)$	Significance
MAE	24.81	22.65	2.46	+
RMSE	24.01	16.06	2.46	+
Precision	24.27	17.78	2.46	+
Recall	25.72	40.19	2.46	+
F1	25.63	37.42	2.46	+

The Fisher distribution with $t-1$ and $(t-1)(q-1)$ degrees of freedom is used for the Friedman test. On the other hand, the null hypothesis in the Friedman test means that all methods perform equivalently at the significance level α. Therefore, the null hypothesis is accepted when the calculated value of F_F is less than the critical value. Otherwise, the null hypothesis is rejected in the Friedman test. It should be noted that, three different datasets and ten recommendation methods are used in the experimental results (i.e. Section 4.4). Therefore, the parameters q and t are set to $q = 3$ and $t = 10$ in the experiments. Moreover, the value of significance level is set to $\alpha = 0.05$, and the critical value of Fisher distribution with $t-1=9$ and $(t-1)(q-1) = 18$ is equal to $F(9, 18) = 2.46$. Table 11 shows the results of the Friedman test for comparison between the recommendation methods based on different evaluation measures. It should be noted that, these results are computed according to the results in Tables 1–10. It can be concluded from Table 11 that, the calculated values of F_F are greater than the critical value 2.46 for all of the evaluation measures. Therefore, the null hypothesis is rejected and it can be concluded that these results are statistically significant.

5. Conclusion and future work

Recommender system is an important tool which is used in various research areas such as information retrieval and information filtering. Although a massive number of algorithms have been developed for the recommender systems, there are still several shortcomings to be resolved in these systems. In this paper, a novel social recommendation method based on an adaptive neighbor selection mechanism is proposed to improve the performance of the recommender systems on the prediction accuracy and neighbor selection challenges. The obtained results from the performed experiments on the three real-world datasets show that in most cases the proposed method outperforms the other state-of-the-art methods. Future work will focus on incorporating additional information such as distrust statements between the users into the proposed neighbor selection mechanism to improve the performance of the recommender systems. In addition, the performance of the proposed method can be improved by considering fuzzy concepts and community detection techniques for identifying the initial neighbors set of the users. Finally, the proposed method can be used in other types of the recommender systems such as context-aware recommender systems to improve the performance of these systems.

References

Alahmadi, D. H., & Zeng, X. J. (2015). ISTS: Implicit social trust and sentiment based approach to recommender systems. *Expert Systems with Applications, 42*, 8840–8849.
Bahmani, B., Kumar, R., & Vassilvitskii, S. (2012). Densest subgraph in streaming and mapreduce. In *The 38th international conference on very large data bases: Vol. 5* (pp. 454–465).
Bedi, P., & Vashisth, P. (2014). Empowering recommender systems using trust and argumentation. *Information Sciences, 279*, 569–586.
Bilge, A., & Polat, H. (2013). A scalable privacy-preserving recommendation scheme via bisecting k-means clustering. *Information Processing & Management, 49*, 912–927.
Birtolo, C., & Ronca, D. (2013). Advances in clustering collaborative filtering by means of fuzzy C-means and trust. *Expert Systems with Applications, 40*, 6997–7009.
Cechinel, C., Sicilia, M. A., Alonso, S. S., & Barriocanal, E. G. (2013). Evaluating collaborative filtering recommendations inside large learning object repositories. *Information Processing & Management, 49*, 34–50.
Deng, S., Huang, L., & Xu, G. (2014). Social network-based service recommendation with trust enhancement. *Expert Systems with Applications, 41*, 8075–8084.
Domingues, M. A., Jorge, A. M., & Soares, C. (2013). Dimensions as virtual items: improving the predictive ability of top-N recommender systems. *Information Processing & Management, 49*, 698–720.
Fang, H., Guo, G., & Zhang, J. (2015). Multi-faceted trust and distrust prediction for recommender systems. *Decision Support Systems, 71*, 37–47.
Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association, 32*, 675–701.
Guo, G., Zhang, J., & Smith, N. Y. (2015a). Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems. *Knowledge-Based Systems, 74*, 14–27.
Guo, G., Zhang, J., & Smith, N. Y. (2015b). 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).
Guo, G., Zhang, J., & Thalmann, D. (2014). Merging trust in collaborative filtering to alleviate data sparsity and cold start. *Knowledge-Based Systems, 57*, 57–68.
Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems, 22*, 5–53.
Hernando, A., Bobadilla, J., Ortega, F., & Tejedor, J. (2013). Incorporating reliability measurements into the predictions of a recommender system. *Information Sciences, 218*, 1–16.
Hernando, A., Bobadilla, J., & Ortega, F. (2016). A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model. *Knowledge-Based Systems, 97*, 188–202.

Hofmann, T. (2004). Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems, 22*, 89–115.
Hong, F. X., Zheng, X. L., & Chen, C. C. (2016). Latent space regularization for recommender systems. *Information Sciences, 360*, 202–216.
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 eecommender systems (RecSys)* (pp. 135–142).
Javari, A., & Jalili, M. (2015). A probabilistic model to resolve diversity–accuracy challenge of recommendation systems. *Knowledge and Information Systems, 44*, 609–627.
Kaššák, O., Kompan, M., & Bieliková, M. (2016). Personalized hybrid recommendation for group of users: Top-N multimedia recommender. *Information Processing & Management, 52*, 459–477.
Kim, Y. A., & Phalak, R. (2012). A trust prediction framework in rating-based experience sharing social networks without a web of trust. *Information Sciences, 191*, 128–145.
Lee, S., Kahng, M., & Lee, S. G. (2015). Constructing compact and effective graphs for recommender systems via node and edge aggregations. *Expert Systems with Applications, 42*, 3396–3409.
Liu, H., Hu, Z., Mian, A., Tian, H., & Zhu, X. (2014). A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-Based Systems, 56*, 156–166.
Ma, B., Zhang, N., Liu, G., Li, L., & Yuan, H. (2016). Semantic search for public opinions on urban affairs: A probabilistic topic modeling-based approach. *Information Processing & Management, 52*, 430–445.
Massa, P., & Avesani, P. (2007). Trust-aware recommender systems. *ACM conference on recommender systems*.
Moon, H. S., Kim, J. K., & Ryu, Y. U. (2013). A sequence-based filtering method for exhibition booth visit recommendations. *International Journal of Information Management, 33*, 620–626.
Moradi, P., Ahmadian, S., & Akhlaghian, F. (2015). An effective trust-based recommendation method using a novel graph clustering algorithm. *Physica A, 436*, 462–481.
O'Mahony, M. P., Hurley, N. J., & Silvestre, G. C. M. (2005). Recommender systems: Attack types and strategies. In *Proceedings of the 20th national conference on American association for artificial intelligence (AAAI-05)* (pp. 334–339).
Park, Y., Park, S., Jung, W., & Lee, S. (2015). Reversed CF: A fast collaborative filtering algorithm using a k-nearest neighbor graph. *Expert Systems with Applications, 42*, 4022–4028.
Pereira, A. L. V., & Hruschka, E. R. (2015). Simultaneous co-clustering and learning to address the cold start problem in recommender systems. *Knowledge-Based Systems, 82*, 11–19.
Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B. (2011). *Recommender systems handbook*. Springer Science, Business Media.
Shani, G., Brafman, R. I., & Heckerman, D. (2005). An MDP-based recommender system. *Journal of Machine Learning Research, 6*, 1265–1295.
Tsa, C. I,F., & Hung, C. (2012). Cluster ensembles in collaborative filtering recommendation. *Applied Soft Computing, 12*, 1417–1425.
Tsai, C. Y., & Lai, B. H. (2015). A Location-Item-Time sequential pattern mining algorithm for route recommendation. *Knowledge-Based Systems, 73*, 97–110.
Yan, S., Zheng, X., Chen, D., & Wang, Y. (2013). Exploiting two-faceted web of trust for enhanced-quality recommendations. *Expert Systems with Applications, 40*, 7080–7095.
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).
Yang, X., Guo, Y., & Liu, Y. (2013). Bayesian-inference-based recommendation in online social networks. *IEEE Transactions on Parallel and Distributed Systems, 24*, 642–651.
Yuan, W., Guan, D., Lee, Y. K., Lee, S., & Hur, S. J. (2010). Improved trust-aware recommender system using small-worldness of trust networks. *Knowledge-Based Systems, 23*, 232–238.