Multimedia Tools and Applications
https://doi.org/10.1007/s11042-018-7079-x

A novel approach based on multi-view reliability measures to alleviate data sparsity in recommender systems

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

Received: 7 June 2018 / Revised: 11 November 2018 / Accepted: 12 December 2018
Published online: 14 January 2019

Abstract

Recommender systems are intelligent programs to suggest relevant contents to users according to their interests which are widely expressed as numerical ratings. Collaborative filtering is an important type of recommender systems which has established itself as the principal means of recommending items. However, collaborative filtering suffers from two important problems including cold start and data sparsity. These problems make it difficult to accurately compute user similarity and hence to find reliable similar users. To deal with these problems, a novel recommender method is proposed in this paper which is based on three different views of reliability measures. For the first view, a user-based reliability measure is proposed to evaluate the performance of users' rating profiles in predicting unseen items. Then, a novel mechanism is proposed to enhance the rating profiles with low quality by adding a number of reliable ratings. To this end, an item-based reliability measure is proposed as the second view of the reliability measures and then a number of items with highest reliability values are selected to add into the target rating profile. Then, similarity values between users and also initial ratings of unseen items are calculated using the enhanced users' rating profiles. Finally, a rating-based reliability measure is used as the third view of the reliability measures to evaluate the initial predicted ratings and a novel mechanism is proposed to recalculate unreliable predicted ratings. Experimental results using four well-known datasets indicate that the proposed method significantly outperforms other recommender methods.

Keywords Recommender systems · Collaborative filtering · Reliability measure · Data sparsity · Rating profile

✉ 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

1 Introduction

With the extensive development of web technologies and e-commerce sites in the past years, an increasing number of online services are becoming popular such as Google News and Yahoo! News for reading news articles, Netflix and Youtube for watching movies and videos, etc. These online services lead to increase the amount of information on the web which is called information overload problem. Therefore, users have to spend much more time to find their interesting items among a large number of choices. Recommender systems have proven to be useful techniques to deal with this problem and can help the users in finding their relevant items in a reasonable time. The main idea behind recommender systems is to use users' past preferences to predict future interests of users. The degree of users' satisfaction depends on the quality of results provided by recommender systems. Therefore, developing a powerful technique is an important issue to improve the performance of a recommender system.

Recommender techniques can be categorized into three main groups including collaborative filtering, content-based, and hybrid methods. Collaborative filtering analyzes accessing behaviors of users similar to target user and then recommends items based on collaborative activities [7, 27]. To this end, this method uses opinions and actions of other users with similar tastes to provide suggestions to the target user. The main idea of collaborative filtering approach is that if users have similar accessing behaviors in the past, they will also prefer similar items in the future. Collaborative filtering approaches can be categorized into two groups including memory-based and model-based methods. Memory-based methods use a similarity measure to calculate predictions for users based on their past ratings. These predictions can be computed as a weighted average of ratings given by the neighbors of the target user which the weight is based on similarity measure between the target user and his/her neighbors [16, 40]. In the model-based methods, a model is constructed based on user-item ratings matrix in training phase. Then, the trained model is used to predict unseen items for users in test set. To construct the model, several data mining and machine learning techniques are used as probabilistic models [14], latent semantic models [15, 21], clustering models [1, 19, 32], dimensionality reduction techniques [28, 35], etc. Content-based recommender methods attempt to suggest relevant items to the target user by using the correlation between contents of items and users' preferences [5, 48]. Hybrid recommender methods are based on combining some other techniques to improve the overall performance of recommender systems. For example, a hybrid method can be constructed by combining collaborative filtering and content-based recommender methods to produce more accurate recommendations for users [50].

In reality, users usually provide a few ratings about items of recommender systems. Therefore, the users may not have sufficient and enough common ratings. It leads to reduce the performance of the systems in calculating similarity values between the users and also finding similar users. These problems called as cold start and data sparsity in recommender systems [41]. The cold start problem is related to the situation when a user (item) in the system has expressed (received) a few number of ratings [20, 24]. On the other hand, the data sparsity problem is related to sparsity of ratings that recommender systems face, since the number of items is usually millions and users can provide ratings for small portions of the items [33, 45]. These problems occur when available data in the system is insufficient for identifying similar users or items as neighbors set. In other words, these problems occur when there is no intersection at all between two users (items) based on available ratings and hence similarity measure is not computable at all. Even when the computation of similarity measure is possible, it may not be a reliable value because of insufficient information processed [31, 49].

In this paper, a novel method is proposed to alleviate data sparsity problem with considering three different views of reliability measures in recommender systems. The basic idea is to evaluate the performance of users' rating profiles in predicting unseen items as well as enhancing their quality by using a novel mechanism. A user's rating profile refers to the ratings which are assigned to items by the user. To this end, a novel user-based reliability measure is proposed using four different factors to evaluate the quality of users' rating profiles in predicting unseen items. This reliability measure can be used to identify unreliable users' profiles which may produce inaccurate predictions. Moreover, a novel mechanism is proposed to enhance the unreliable users' profiles. In this mechanism, the unreliable profiles are enhanced by adding a number of reliable ratings. The main advantage of the proposed mechanism is to improve the performance of recommender systems especially about sparse user-item ratings matrixes. Moreover, an item-based reliability measure is proposed based on three different factors to calculate the reliability of items. Then, a subset of items with highest reliability measure is selected for adding into the user's rating profile. These reliable ratings are calculated as a weighted average of ratings given by users which the weight is proportional to the user-based reliability measures of the users. Then, similarity values between the target user and other users are calculated based on the enhanced profiles and initial ratings of unseen items are predicted using these calculated similarity values. Finally, the quality of initial predicted ratings is evaluated using a rating-based reliability measure and a rating with low reliability is recalculated using a novel mechanism. The main contributions of the paper are summarized as follows:

- A user-based reliability measure is proposed with considering four different factors to evaluate the quality of users' rating profiles in predicting unseen items. This reliability measure is used to identify unreliable users' profiles which have a negative effect on the performance of recommender systems.
- An item-based reliability measure is proposed which is based on three different factors to evaluate the reliability of items. This measure is used to select a subset of items with highest reliability measure for adding to the user's profile. This leads to improve the reliability and also accuracy of the added ratings.
- An effective mechanism is proposed to enhance the quality of the unreliable users' profiles by adding a number of reliable ratings. To this end, a subset of items is selected based on the item-based reliability measure. Therefore, the proposed mechanism leads to alleviate data sparsity problem by adding reliable ratings into the users' profiles with low user-based reliability measure.
- A rating-based reliability measure is used to evaluate the quality of initial predicted ratings. Moreover, the predicted ratings with low quality are recalculated by removing useless users from the neighbors set. To this end, a confidence measure is proposed to identify useful users in the initial neighbors set of users.
- A series of experiments are performed on four well-known datasets to verify the performance of the proposed method in comparison with other recommender systems. The results demonstrate that the proposed method can outperform the other methods.

The rest of the 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 other recommender methods by performing several experiments on four well-known datasets. Finally, some concluding comments are discussed in Section 5.

2 Related works

In recent years, the amount of information available online has grown very fast and it has exceeded the capacity of individual users to process such information. This caused a strong interest in research fields and technologies to manage information overload problem. Moreover, intelligent algorithms are expected to provide better user experience in using online services [25]. Recommender systems have thus been created to assist users in retrieving and accessing interesting items by automatically acquiring user preferences from historical data and matching items with the preferences [34, 36].

Cold start problem is about the users who only have rated a few items in recommender system. Recommender systems cannot provide suitable and reliable recommendations for this kind of users, because they have not enough ratings to use in recommendation process. Many approaches have been proposed to alleviate the cold start problem by using additional information such as social information [6, 9, 26]. Although these methods can improve the performance of recommender systems on the cold start problem, but they need additional information which may not exist in all of the recommender systems. In [24], an approach is proposed to resolve the cold start problem which incorporates classification methods in a pure collaborative filtering system while the use of demographic data helps for the identification of other users with similar behavior. In [20], a method is proposed to address the cold start problem which is based on building models derived from explicit ratings and error information of predicted ratings. This method uses these constructed models to calculate new predictions of actual ratings. In [8], the authors proposed a method based on the query expansion techniques used in information retrieval to alleviate the cold start problem in recommender systems. To this end, three kinds of techniques including item-global, item-local and user-local are proposed and the performed experiments show that both item-global and user-local methods improve the performance of the system based on precision metric.

Many approaches have been proposed to deal with data sparsity problem in recommender systems. In [37], an efficient imputation method is proposed to address the data sparsity problem in the context of neighborhood-based collaborative filtering. To this end, a set of key ratings are identified and an auto-adaptive imputation method is proposed to fill the missing values in the set of key ratings. Then, similarity values between users or items can be calculated using the filled missing values of the key ratings. In [44], two strategies are considered as missing value imputation in nearest neighbor based collaborative filtering to resolve the data sparsity problem. Therefore, two novel effective collaborative filtering methods based on missing data imputation are proposed which utilize user's demographic information before conducting collaborative filtering process. To this end, user's age range and occupation information are used in the imputation step. In [42], a framework of imputation-boosted collaborative filtering is proposed to address the data sparsity problem in recommender systems. This framework firstly uses an imputation method using machine learned classifier to fill-in the sparse user-item rating matrix. Then, a traditional Pearson correlation-based collaborative filtering method is performed on this matrix to predict a novel rating. In [10], an approach is proposed to resolve the data sparsity problem which is based on multiple types of auxiliary implicit feedback. Therefore, the method generates target data from a linear regression of auxiliary feedback and forms the nearest neighbors with a set of purchased items in multiple dimensions. Then, a novel ranking model is considered to accommodate both of the original and generated data. Yang et al. [46] proposed a method called blocks-coupled non-negative matrix factorization to resolve the data sparsity problem. The main advantage of the method is to improve the reconstruction performance of extreme sparseness matrix by blocking the matrix and modeling based on full use of the coupling mechanism. It leads to impose constraints on consistency as the matrix is decomposed.

Although the accuracy of predictions is an important issue in recommender systems, reliability is another important issue which should be considered into recommendation process. Because, it is shown that considering the reliability of predictions can improve the performance of recommender systems and also make better recommendations for users [31]. Several studies show that including reliability measures or confidence models in recommender systems leads to increase the quality of recommendations [2–4, 13, 29]. In [13], a reliability measure is proposed to evaluate the effectiveness of predicted ratings which is based on two positive and negative factors. The positive factor is based on the summation of similarity values between the target user and her/his neighbors. Moreover, the negative factor is based on the variance of ratings that the neighbors of the target user have made for a specific item. In [29], several approaches are proposed to estimate confidence of individual rating predictions in collaborative filtering-based recommender systems. Moreover, a method is proposed for evaluating the performance of the confidence estimation algorithms. In [49], the authors proposed a novel ranking-based approach which uses posterior rating distribution and confidence level of prediction as two key factors for predicting uncertainty. The goal of this approach is to improve the accuracy of personalized product ranking through incorporating the uncertainty information. In [9], the authors proposed a novel merging method to incorporate social trust information in providing recommendations. This proposed merging method can be used to alleviate the cold start problem by merging the ratings of a user's trusted neighbors to complement and represent the preferences of the user. Moreover, a confidence measure is used to evaluate the quality of merged ratings which is based on the number of ratings and the ratio of conflicts between positive and negative opinions.

3 Proposed method

In this section, the proposed method is introduced as User Profile Enhancing based on Multi-view Reliability measures (in short UPEMR). The proposed method consists of six steps including: (1) User-based reliability calculation, (2) Item-based reliability calculation, (3) User profile enhancing, (4) Initial rating prediction, (5) Rating-based reliability calculation, and (6) Final rating prediction. In the following subsections, additional details about the steps of the proposed method are discussed.

3.1 User-based reliability calculation

In this step, a user-based reliability measure is proposed based on four different factors. The measure can be used to evaluate the performance of users' rating profiles in predicting unseen items. To calculate the proposed measure, first of all, the four factors are introduced and then a mechanism is considered to combine these factors as the proposed user-based reliability measure. The first factor is the number of ratings in the rating profile of target user a which are provided for items. It can be concluded that the more ratings provided by a user can produce more reliable predictions. Therefore, the first factor has a positive effect on the proposed user-based reliability measure. The factor can be calculated as follows:

$$f_i(I_a) = 1 - \frac{\overline{a}}{\overline{a} + |I_a|} \tag{1}$$

where, $|I_a|$ is the number of ratings that the target user a has assigned to items and $\overline{a}$ is the median of the values for $|I_a|$.

The number of users in neighbors set of the target user is the second factor used to calculate the proposed user-based reliability measure. The predicted ratings can be more reliable, when more users are considered as the neighbors set of the target user into recommendation process. Therefore, the second factor can be considered in calculating the proposed measure as a positive factor. It should be noted that, the neighbors set of the target user can be obtained by considering a threshold value for similarity values between the target user and other users in the system. In other words, the users whose similarity values with the target user are higher than a specific threshold value can be determined as the neighbors set of the target user. Therefore, the second factor is calculated as follows:

$$f_k(K_a) = 1 - \frac{\overline{k}}{\overline{k} + |K_a|} \tag{2}$$

where, $|K_a|$ is the number of users in the neighborhood of the target user a, $\overline{k}$ is the median of the values of $|K_a|$, and K_a is the set of neighbors for the target user a which can be calculated as follows:

$$K_a = \{u \in U | sim(a, u) \geq \theta\} \tag{3}$$

where, U is the set of all users in the system, $sim(a, u)$ is the similarity value between the target user a and user u, and θ is a threshold value for the similarity value. The similarity value between the target user a and user u is calculated using the Pearson correlation coefficient function as follows:

$$sim(a, u) = \frac{\sum_{i \in A_{a,u}} \left(r_{a,i} - \overline{r_a}\right)\left(r_{u,i} - \overline{r_u}\right)}{\sqrt{\sum_{i \in A_{a,u}} \left(r_{a,i} - \overline{r_a}\right)^2} \sqrt{\sum_{i \in A_{a,u}} \left(r_{u,i} - \overline{r_u}\right)^2}} \tag{4}$$

where, $r_{a,\,i}$ is the rate of item i given by user a, $\overline{r_a}$ is the average of the rates given by user a, and $A_{a,\,u}$ is the set of items which are rated by both users a and u.

After calculating the neighbors set of the target user, it is an important issue to determine the number of ratings that are provided by the neighbor users. If the neighbor users provide more ratings for the items then the system can produce more reliable recommendations for the target user. Therefore, the number of ratings provided by the neighbor users of the target user can be used as the third factor of the user-based reliability measure. It should be noted that, the third factor has a positive effect on the measure because more ratings provided by the neighbor users leads to produce more reliable predictions for the target user. Therefore, the following equation can be used to calculate the third factor of the user-based reliability measure:

$$f_{i_k}(I_{K_a}) = 1 - \frac{\overline{i_k}}{\overline{i_k} + |I_{K_a}|} \tag{5}$$

where, $|I_{K_a}|$ is the number of ratings that the neighbors of the target user a have assigned to items, K_a is the set of neighbors for the target user a (see Eq. (3)), and $\overline{i_k}$ is the median of the values of $|I_{K_a}|$.

One of the important factors that the reliability measure depends on is the similarity values between the target user and other users in their neighbors set. In other words, if the target user has high similarity values with other users in nearest neighbors set then it can be concluded that, the predicted ratings based on these neighbors have high reliability values. Therefore, the fourth factor is based on the summation of the similarity values between the target user and other users in their neighbors set which is used as a positive factor for the user-based reliability measure. The fourth factor is calculated as follows:

$$f_s(S_a) = 1 - \frac{\overline{s}}{\overline{s} + S_a} \tag{6}$$

where,

$$S_a = \sum_{u \in K_a} sim(a, u) \tag{7}$$

K_a is the set of neighbors for the target user a (see Eq. (3)), $sim(a, u)$ is the similarity value between the target user a and user u(see Eq. (4)), and $\overline{s}$ is the median of the values of S_a.

Finally, the calculated factors can be used to measure the reliability of the target user's profile. To this end, a weight-based mechanism is used to combine the factors as the proposed user-based reliability measure. Some assumptions should be considered to determine the weights of the factors. First, the factors $|I_a|$ and $|K_a|$ in Eqs. (1) and (2) are not dependent on any other factors. In other words, the values of the factors $|I_a|$ and $|K_a|$ are independently calculated based on the number of ratings provided by the target user a and the number of users in neighbors set of the target user a, respectively. Therefore, the weights of these factors are considered as with a constant value of 1 in the user-based reliability measure. Second, the values of $|I_{K_a}|$ and S_a in Eqs. (5) and (7) are dependent on the value of $|K_a|$. In other words, the lower value of $|K_a|$ leads to reduce the values of $|I_{K_a}|$ and S_a, and vice versa. Therefore, the weights of $|I_{K_a}|$ and S_a are considered as the value of $f_k(K_a)$. After determining the weights of the factors, the proposed user-based reliability measure can be calculated as the geometric average of the identified factors as follows:

$$UR_a = \left[f_i(I_a).f_k(K_a).f_{i_k}(I_{K_a})^{f_k(K_a)}.f_s(S_a)^{f_k(K_a)} \right]^{\frac{1}{2+2f_k(K_a)}} \tag{8}$$

where, UR_a is the user-based reliability measure for the target user a. Based on Eq. (8), the proposed user-based reliability measure can be calculated using four different factors. There are two main advantages in the proposed reliability measure. Firstly, the four factors are determined based on their impacts on the quality of user's rating profile in predicting unseen items. Therefore, the proposed reliability measure can be used to evaluate the effectiveness of the target user's profile. Secondly, the only information that used in the proposed measure is the ratings provided by users in user-item rating matrix and the measure needs no more information to evaluate the quality of users' profiles. Therefore, the proposed measure can be used in various types of recommender systems which provide user-item rating matrix.

3.2 Item-based reliability calculation

In this step, an item-based reliability measure is proposed based on three different factors which can be used to evaluate the effectiveness of items for users' rating profiles. Specifically, the item-based reliability measure is used to select a subset of items to add into target user's profile in the profile enhancing mechanism. To this end, a subset of items with highest reliability measure can be used for adding into the target user's profile. To calculate the measure, three factors are defined and then the proposed measure is calculated as a combination of the factors. The number of ratings provided for the target item i by users is the first factor of the proposed item-based reliability measure. It should be noted that, the more ratings for the item i lead to increase its reliability value. Therefore, the factor has a positive effect on the reliability measure and can be calculated as follows:

$$f_i(I_i) = 1 - \frac{\bar{i}}{\bar{i} + |I_i|} \tag{9}$$

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

The second factor of the item-based reliability measure is the standard deviation of ratings provided for the target item i. The factor determines the difference of users' opinions about the target item. It can be concluded that, the higher values of the factor make a lower reliability value for the target item. Therefore, the factor has a negative effect on the reliability measure and a decreasing function can be used to calculate it as follows:

$$f_{sd}(stdev(I_i)) = \frac{max - stdev(I_i)}{max - min} \tag{10}$$

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

The third factor of the item-based reliability measure is defined based on the reliability values of the users who have provided a rating for the target item i. To this end, the user-based reliability values are used and the summation of the values is considered as the third factor of the measure. Obviously, providing ratings by more reliable users leads to increase the reliability of the target item. Therefore, the third factor is a positive factor for the measure and can be calculated as follows:

$$f_c(C_i) = 1 - \frac{\bar{c}}{\bar{c} + C_i} \tag{11}$$

where,

$$C_i = \sum_{u \in U_i} UR_u \tag{12}$$

U_i is the set of users which have assigned a rating to item i, UR_u is the user-based reliability measure for user u which is calculated using Eq. (8), and $\bar{c}$ is the median of the values of C_i.

Finally, the proposed item-based reliability measure can be calculated as the weights of the identified factors. The first factor (i.e. $|I_i|$) is not dependent on any other factors. Therefore, the weight of this factor is considered as with a constant value of 1. On the other hand, the values of $stdev(I_i)$ and C_i are dependent on the value of $|I_i|$. In other words, the lower value of $|I_i|$ makes the values of $stdev(I_i)$ and C_i also be low and vice versa. Hence, the weights of $stdev(I_i)$ and C_i are considered as the value of $f_i(I_i)$. Therefore, the proposed item-based reliability measure can be calculated as the geometric average of the identified factors as follows:

$$IR_i = \left[f_i(I_i).f_{sd}(stdev(I_i))^{f_i(I_i)}.f_c(C_i)^{f_i(I_i)}\right]^{\frac{1}{1+2f_i(I_i)}} \tag{13}$$

where, IR_i is the proposed item-based reliability measure for item i. It should be noted that, the measure is based on three factors which can be calculated by using user-item rating matrix. Therefore, the proposed measure needs no more information to calculate the three factors. On the other hand, a large number of recommender systems have been developed which provide user-item rating matrix as the main information for the systems. Therefore, the proposed measure can be used in all of the recommender systems which provide user-item rating matrix to improve their performance in providing more reliable recommendations.

3.3 User profile enhancing

After calculating the proposed user-based and item-based reliability measures, a novel mechanism is introduced to enhance the rating profiles of the users with low user-based reliability measure. To this end, the rating profiles of the users are modified by adding a number of reliable ratings based on the proposed reliability measures. The main advantage of this step is to alleviate data sparsity problem in recommender systems. In other words, the proposed method makes a denser user-item rating matrix than the original matrix by adding a number of reliable ratings into the users' rating profiles. Therefore, the calculation of similarity values between the users performs simply and also the system can be able to find better neighbors set for the users.

The main idea behind the proposed mechanism is to use the proposed user-based reliability measure to determine the effectiveness of user's rating profile. In other words, the higher value of the user-based reliability measure shows that the rating profile of the user has a higher performance to predict unrated items. Therefore, in the proposed mechanism, a threshold value (i.e. α) is used to enhance the rating profiles of the users. To this end, the user's rating profile will be modified if its user-based reliability measure is lower than the threshold value α.

In the proposed mechanism, a number of reliable ratings will be added into the rating profiles of the users whose user-based reliability measures are lower than the threshold value. It should be noted that, the number of reliable ratings which added into the user's rating profile depends on its user-based reliability measure. In other words, a user with a higher value of the user-based reliability measure needs a lower number of reliable ratings for adding into her/his rating profile, and vice versa. Therefore, the following equation is used to calculate the sufficient number of reliable ratings for adding into the rating profile of the target user a:

$$m' = (1-UR_a) \times m \tag{14}$$

where, UR_a is the user-based reliability measure for the target user a which can be calculated by Eq. (8), and m is the maximum number of reliable ratings that can be added into the user's rating profile.

After calculating the number of needed reliable ratings for adding into the target user's rating profile (i.e. Eq. (14)), a number of items (i.e. m') with highest value of the item-based reliability measure are selected for using in the proposed user rating profile enhancing mechanism. Then, the reliable ratings of the selected items are calculated based on the proposed user-based reliability measure as follows:

$$R_i = \frac{\sum_{u \in U_i} UR_u . r_{u,i}}{\sum_{u \in U_i} UR_u} \tag{15}$$

where, R_i is the reliable rating of item i, U_i is the set of users who have assigned a rating to item i, UR_u is the user-based reliability measure for user u which is calculated using Eq. (8), and $r_{u,i}$ is the rate of item i given by user u.

Finally, the calculated reliable ratings can be added into the rating profile of the target user. It should be noted that, the real ratings of the target user have been considered without any changes and the reliable ratings which added into the target user's profile are completely different from these real ratings. The enhanced profile of the target user has a higher performance than her/his initial profile, because there are some new reliable ratings in the enhanced profile which lead to alleviate data sparsity problem in recommender systems.

3.4 Initial rating prediction

In this step, similarity values between the target user and the others can be calculated based on the enhanced rating profiles using Eq. (4). Then, a set of neighbors for the target user a is formed as initial neighbors set using Eq. (3). Finally, the initial rating of unrated item i for the target user a can be predicted using the following equation:

$$P_{a,i} = \overline{r_a} + \frac{\sum_{u \in K_{a,i}} sim(a,u).\left(r_{u,i} - \overline{r_u}\right)}{\sum_{u \in K_{a,i}} sim(a,u)} \tag{16}$$

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

3.5 Rating-based reliability calculation

In this step, a reliability measure [13] is used to calculate the reliability values of initial predicted ratings. This reliability measure is useful to determine the quality of initial predicted ratings. To this end, two positive and negative factors are considered to calculate the rating-based reliability measure. The positive factor is based on the summation of similarity values between the target user and other users in his/her initial neighbors set. Therefore, this factor can be calculated using Eq. (17) as follows:

$$f_s\left(S_{a,i}\right) = 1 - \frac{\overline{S}}{\overline{S} + S_{a,i}} \tag{17}$$

where, $\overline{S}$ denotes the median values of $S_{a,\,i}$ in the system, and $S_{a,\,i}$ is the summation of similarity values between the target user a and his/her initial neighbors set which can be calculated as follows:

$$S_{a,i} = \sum_{u \in K_{a,i}} sim(a,u) \tag{18}$$

where, $K_{a,\,i}$ is a subset of users in the initial neighbors set of target user a who have rated item i, and $sim(a,u)$ denotes the similarity value between users a and u which is calculated using Eq. (4).

On the other hand, the negative factor of the rating-based reliability measure is defined based on the differences between the ratings which are assigned to the target item by the neighbors of the target user. Therefore, the negative factor can be calculated using Eq. (19) as follows:

$$f_v(V_{a,i}) = \left(\frac{max{-}min{-}V_{a,i}}{max{-}min}\right)^{\gamma} \tag{19}$$

where,

$$\gamma = \frac{\ln 0.5}{\ln \dfrac{max{-}min{-}\overline{V}}{max{-}min}} \tag{20}$$

and $\overline{V}$ represents the median of the values of $V_{a,\,i}$ in the system, max and min indicate the maximum and minimum values of ratings in the system, respectively. The value of $V_{a,\,i}$ can be calculated using Eq. (21) as follows:

$$V_{a,i} = \frac{\sum_{u \in K_{a,i}} sim(a,u)\left(r_{u,i}{-}\overline{r_u}{-}P_{a,i} + \overline{r_a}\right)^2}{\sum_{u \in K_{a,i}} sim(a,u)} \tag{21}$$

where, $\overline{r_a}$ is the average of ratings for the target user a, $K_{a,\,i}$ is a subset of users in the initial neighbors set of target user a who have rated item i, $P_{a,\,i}$ is the initial rating of item i for user a which is calculated using Eq. (16), and $sim(a, u)$ indicates the similarity value between users a and u which is calculated using Eq. (4).

Finally, the rating-based reliability measure of the initial rating of item i for user a (i.e. $P_{a,\,i}$) can be calculated based on the calculated positive and negative factors using Eq. (22):

$$RR_{a,i} = \left(f_s(S_{a,i}).f_v(V_{a,i})^{f_s(S_{a,i})}\right)^{\frac{1}{1+f_s(S_{a,i})}} \tag{22}$$

where, $f_s(S_{a,\,i})$ and $f_v(V_{a,\,i})$ refer to the positive and negative factors of the rating-based reliability measure which are calculated using Eqs. (17) and (19), respectively.

3.6 Final rating prediction

In this step, the initial predicted ratings are evaluated using the rating-based reliability measure. In other words, the calculated rating-based reliability measures (i.e. Eq. (22)) for the initial predicted ratings are used to determine the quality of these ratings. Moreover, an effective mechanism is used to improve the quality of the initial predicted ratings with low reliability through removing useless users from initial neighbors set of the target user. It should be noted that, all of the users in initial neighbors set of the target user are used to predict the initial ratings using Eq. (16). However, some of these users may have a negative effect on the quality of the initial predicted ratings. Therefore, these useless users can be removed from the initial neighbors set of the target user. To this end, the rating-based reliability measure of the initial predicted ratings is calculated using Eq. (22) and then the ratings which their reliability values are less than a threshold value β can be determined as unreliable ratings. Then, a new rating can be calculated for the unreliable ratings using a new set of the neighbors for the target user. The new neighbors set of the target user is formed by removing useless users from the initial neighbors set. Therefore, a confidence model between the users is proposed which is based on the rating-based reliability measure. This confidence model is calculated using Eq. (23) as follows:

$$C_{a,v} = \frac{\sum_{i \in A_{a,v}} RR_{a,i}\left(r_{a,i} - \overline{r_a}\right) RR_{v,i}\left(r_{v,i} - \overline{r_v}\right)}{\sqrt{\sum_{i \in A_{a,v}} RR_{a,i}^2 \left(r_{a,i} - \overline{r_a}\right)^2} \sqrt{\sum_{i \in A_{a,v}} RR_{v,i}^2 \left(r_{v,i} - \overline{r_v}\right)^2}} \tag{23}$$

where, $C_{a,\,v}$ indicates the confidence value between users a and v, $A_{a,\,v}$ is a set of items which are rated by both of the users a and v, $RR_{a,\,i}$ refers to the rating-based reliability measure which is calculated using Eq. (22), $r_{a,\,i}$ is the rating of item i given by user a, and $\overline{r_a}$ is the average of ratings provided by user a.

The users in the initial neighbors set of the target user whose confidence values are less than a threshold value γ are considered as useless neighbors. Therefore, the new neighbors set of the target user can be formed by removing the useless users from the initial neighbors set. In other words, user v will be removed from the initial neighbors set of the target user a, if the confidence value $C_{a,\,v}$ is less than the threshold value γ. After calculating the new neighbors set of the target user for an unreliable predicted rating, a new rating can be calculated based on this new neighbors set. Therefore, the final ratings are calculated using Eq. (24) as follows:

$$P_{a,i}^{new} = \overline{r_a} + \frac{\sum_{u \in K_{a,i}^{new}} sim(a,u)\left(r_{u,i} - \overline{r_u}\right)}{\sum_{u \in K_{a,i}^{new}} sim(a,u)} \tag{24}$$

where, $P_{a,i}^{new}$ indicates the final rating of item i for the target user a, $sim(a, u)$ is the similarity value between the users a and u which is calculated using Eq. (4), $r_{u,\,i}$ is the rating of item i provided by user u, $\overline{r_u}$ is the average of ratings provided by user u, and $K_{a,i}^{new}$ refers to a subset of users in the new neighbors set of target user a who have rated item i. The new neighbors set $K_{a,i}^{new}$ can be calculated using the following Equation:

$$K_{a,i}^{new} = \left\{ v \in K_{a,i} \middle| C_{a,v} \ge \gamma \right\} \tag{25}$$

where, $K_{a,\,i}$ refers to the initial neighbors set of target user a which is calculated using Eq. (3), $C_{a,\,v}$ is the confidence value between the users a and v, and γ is the threshold value for confidence values. After calculating the final ratings of unseen items for the target user, a subset of top_N items with highest ratings are selected as recommendation list to suggest to the target user. The pseudo code of the proposed method is shown in Fig. 1.

Algorithm 1. User Profile Enhancing based on Multi-view Reliability measures (UPEMR)

Input: Parameters $\boldsymbol{\theta}$, $\boldsymbol{\alpha}$, $\boldsymbol{m}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$.
Output: Predicted ratings for users.
Begin algorithm:

Split dataset into a training set $\boldsymbol{Tr}$ and a test set $\boldsymbol{Te}$;
Let $\boldsymbol{U}$ be the set of all users;
for all $\boldsymbol{a} \in \boldsymbol{U}$ **do**
    Calculate the four factors of the user-based reliability measure using Eqs. (1)-(7) based on the training set $\boldsymbol{Tr}$;
    Calculate the user-based reliability measure $\boldsymbol{UR_a}$ for the target user $\boldsymbol{a}$ using Eq. (8);
end for
Let $\boldsymbol{I}$ be the set of all items;
for all $\boldsymbol{i} \in \boldsymbol{I}$ **do**
    Calculate the three factors of the item-based reliability measure using Eqs. (9)-(12) based on the training set $\boldsymbol{Tr}$;
    Calculate the item-based reliability measure $\boldsymbol{IR_i}$ for the item $\boldsymbol{i}$ using Eq. (13);
end for
for all $\boldsymbol{a} \in \boldsymbol{U}$ **do**
    if $(\boldsymbol{UR_a} < \boldsymbol{\alpha})$ **then**
        Let $\boldsymbol{I_a}$ be the set of items that have been rated by the target user $\boldsymbol{a}$;
        Set $\boldsymbol{I'_a = I - I_a}$;
        Sort $\boldsymbol{I'_a}$ descending based on their item-based reliability measures;
        Calculate $\boldsymbol{m'}$ using Eq. (14);
        Select $\boldsymbol{top\_m'}$ items from $\boldsymbol{I'_a}$ as $\boldsymbol{\tilde{I}_a}$;
        for all $\boldsymbol{i} \in \boldsymbol{\tilde{I}_a}$ **do**
            Calculate reliable rating of the item $\boldsymbol{i}$ using Eq. (15);
            Add the calculated reliable rating to the rating profile of the target user $\boldsymbol{a}$;
        end for
    end if
end for
for all $\boldsymbol{a} \in \boldsymbol{U}$ **do**
    Calculate the similarity values between the target user $\boldsymbol{a}$ and other users using Eq. (4) based on the enhanced rating profiles;
    Calculate the initial nearest neighbors set of the target user $\boldsymbol{a}$ using Eq. (3);
end for
for all $\boldsymbol{r_i(a)} \in \boldsymbol{Te}$ **do**
    Predict the initial rating $\boldsymbol{P_{a,i}}$ of the item $\boldsymbol{i}$ for the target user $\boldsymbol{a}$ using Eq. (16);
    Calculate the rating-based reliability measure $\boldsymbol{RR_{a,i}}$ of the item $\boldsymbol{i}$ for the target user $\boldsymbol{a}$ using Eq. (22);
    if $(\boldsymbol{RR_{a,i} < \beta})$ **then**
        Calculate the new nearest neighbors set $\boldsymbol{K_{a,i}^{new}}$ for the target user $\boldsymbol{a}$ using Eq. (25);
        Predict the new rating $\boldsymbol{P_{a,i}^{new}}$ based on $\boldsymbol{K_{a,i}^{new}}$ using Eq. (24);
    end if
end for

End algorithm.

Fig. 1 Pseudo code of the proposed method

3.7 An illustrated example

In this section, an example is represented to describe the steps of the proposed method for predicting an unknown rating for a given user. To this end, a small dataset with five users and five items is used which each user rated a few number of the items. Moreover, the rating values in this dataset are in the range of 1 (min) and 5 (max). This sample dataset is shown in Table 1. It should be noted that, the purpose of this example is to predict the rating of item i_4 for target user u_1 by applying the proposed method.

In the first step of the proposed method, user-based reliability measures for all of the users are calculated using Eqs. (1)-(8). For simplicity, in this example we set $\theta = 0.1$ for Eq. (3) as a threshold value for the similarity values (see Eq. (4)). Therefore, the user-based reliability measures for the users are calculated which UR_{u_1}, UR_{u_2}, UR_{u_3}, UR_{u_4} and UR_{u_5} are equal to

Table 1 The example dataset consisting of five users and five items

	i_1	i_2	i_3	i_4	i_5
u_1	1	–	5	**?**	–
u_2	2	–	4	2	4
u_3	2	5	–	–	3
u_4	–	2	–	3	4
u_5	4	3	2	–	–

0.48, 0.58, 0.5, 0.39 and 0.0, respectively. Then, in the second step, the item-based reliability measures for all of the items are calculated using Eqs. (9)-(13). Based on these equations, the item-based reliability measures for the items are calculated which IR_{i_1}, IR_{i_2}, IR_{i_3}, IR_{i_4} and IR_{i_5} are equal to 0.38, 0.0, 0.0, 0.5 and 0.61, respectively.

n the third step, the proposed mechanism is applied to enhance the rating profiles of users with low user-based reliability measures. Suppose that the threshold value for the user-based reliability measure is equal to 0.5 (i.e. $\alpha = 0.5$). Moreover, in this example we set $m = 4$ for Eq. (14) as the maximum number of reliable ratings that can be added to the user's rating profile. It can be seen that the calculated user-based reliability measures for the users u_1, u_4, and u_5 are less than the predefined threshold value (i.e. $UR_{u_1}, UR_{u_4}, UR_{u_5} < \alpha$). Therefore, the rating profiles of these users are enhanced using the proposed user profile enhancing mechanism. The result of this step is represented in Table 2 in which the reliable ratings are shown in bold face and underlined.

In the fourth step of the proposed method, similarity values between the target user u_1 and the others can be calculated based on the new rating profiles using Eq. (4). Then, the initial rating of item i_4 for the target user u_1 is calculated using Eq. (16) which is equal to 2.8 (i.e. $P_{u_1,i_4} = 2.8$). In the fifth step of the proposed method, the rating-based reliability measure of the initial predicted rating is calculated using Eqs. (17)-(22). Suppose that the rating-based reliability measure is equal to 0.6 (i.e. $RR_{u_1,i_4} = 0.6$). Moreover, in this example we set $\beta = 0.7$ as a threshold value for the rating-based reliability measure. Therefore, the final rating of item i_4 for the target user u_1 can be calculated in the sixth step of the proposed method by removing useless users from the initial neighbors set of the target user. To this end, the confidence values between the target user and others are calculated using Eq. (23). Suppose that the confidence values are $C_{u_1,u_2} = 0.7$, $C_{u_1,u_4} = 0.55$ and $C_{u_1,u_5} = 0.65$. Moreover, suppose that $\gamma = 0.6$ as the threshold value of the confidence measure. Based on the proposed mechanism, the user u_4 is identified as a useless neighbor and can be removed from the initial neighbors set. Therefore, the final rating of item i_4 for the target user u_1 is calculated using Eq. (24) which is equal to 1.71 (i.e. $P^{new}_{u_1,i_4} = 1.71$).

Table 2 The result of the proposed user profile enhancing mechanism

	i_1	i_2	i_3	i_4	i_5
u_1	1	<u>**3.68**</u>	5	**?**	<u>**3.65**</u>
u_2	2	–	4	2	4
u_3	2	5	–	–	3
u_4	<u>**1.69**</u>	2	<u>**4.45**</u>	3	4
u_5	4	3	2	<u>**2.4**</u>	<u>**3.65**</u>

3.8 Computational complexity analysis

In this section, all steps of the proposed method are considered to compute the computational complexity of the method. The proposed method consists of six steps. In the first step, a user-based reliability measure is calculated for the users based on four different factors. The complexity of the user-based reliability calculation is $O(|U||I|)$ where $|U|$ is the number of users and $|I|$ denotes the number of items. In the second step, an item-based reliability measure is calculated for the items which its complexity is $O(|U||I|)$. In the third step, the user profile enhancing mechanism is performed to improve the performance of the users' profiles. The complexity of the mechanism is $O(|U|^2m')$ where m' denotes the number of reliable ratings which are added to the users' profiles. In the fourth step, the initial ratings of unseen items are predicted based on the enhanced profiles. To calculate the initial rating of an unseen item, the similarity value between the target user and other users is needed to be computed using Eq. (4). It should be noted that, the computation of the similarity value between two users needs $O(|I|)$. Therefore, the complexity of calculating the similarity values between all pairs of the users is $O(|U|^2|I|)$. In the fifth step, a rating-based reliability measure is calculated for the initial predicted ratings which its complexity is $O(|U||I|)$. In the sixth step, a confidence value is calculated between the users which its complexity is $O(|U|^2)$. Moreover, the final ratings of the unseen items are calculated which its complexity is $O(|U|^2|I|)$. Therefore, it can be concluded that the complexity of the sixth step is $O(|U|^2+|U|^2|I|)$ which can be reduced to $O(|U|^2|I|)$. Finally, the complexity of the proposed method (i.e. Steps 1-6) is $O(|U||I|+|U||I|+|U|^2m'+|U|^2|I|+|U||I|+|U|^2|I|)$ which can be reduced to $O(|U|^2|I|)$.

4 Experimental results

In this section, several experiments are conducted to compare the performance of the proposed method (i.e. UPEMR) with other recommender methods including User-based collaborative filtering (UCF), Item-based collaborative filtering (ICF) [39], Multi-level collaborative filtering (MLCF) [36], Item-global profile expansion (IGPE) [8], User opinion spreading (UOS) [11], User-local profile expansion (ULPE) [8], Resource allocation based collaborative filtering (RACF) [18], Slope-One [23], Regularized singular value decomposition (RSVD) [43], Non-negative matrix factorization (NMF) [22], Probabilistic matrix factorization (PMF) [30], Two-layer neighbor selection collaborative filtering (TLCF) [51], Popularity-based probabilistic latent semantic analysis (PPLSA) [17], and Multi-channel feature vector based collaborative filtering (MCFV) [52]. Additional details about the experiments are presented in the following subsections.

4.1 Datasets

Four well-known datasets are used in the experiments including MovieLens,[1] Netflix,[2] BookCrossing,[3] and Jester[4] to verify the effectiveness of the proposed method. The

[1] http://grouplens.org/datasets/movielens/
[2] http://www.prea.gatech.edu/
[3] http://www2.informatik.uni-freiburg.de/~cziegler/BX/
[4] http://eigentaste.berkeley.edu/dataset/

MovieLens dataset was collected by the GroupLens research group [38] which contains 1682 movies, 943 users, and 100,000 ratings. All of the rating values in the MovieLens dataset are integer numbers in the range of 1 (bad) to 5 (excellent). In this dataset, each user has rated at least 20 movies. The used Netflix dataset is a small version of the original Netflix dataset which is available in the personalized recommendation algorithms toolkit (PREA) [47]. This dataset consists of 4427 users, 1000 items, and 56,136 ratings. The ratings are integer numbers in the rage of 1 (bad) to 5 (excellent) scales. The BookCrossing is a book recommendation dataset which users have provided ratings for items (i.e. books). This dataset contains 1.1 million ratings of 270,000 books which are provided by 90,000 users. In addition, the ratings in this dataset are on a scale from 1 (bad) to 10 (excellent). The Jester is a jock recommendation dataset which contains the ratings about 100 jocks provided by 24,938 users. The ratings in this dataset are real numbers from −10 to 10 and the users have provided between 15 and 35 ratings for the jocks. Table 3 shows the descriptions of the evaluation datasets.

4.2 Evaluation metrics

To evaluate the performance of recommender methods, four evaluation metrics are used in the experiments which include normalized mean absolute error (NMAE), normalized root mean squared error (NRMSE), Kendall Tau correlation (τ), and catalog coverage (CC). The NMAE and NRMSE metrics are used to measure the accuracy of predicted ratings. To this end, the predicted ratings are compared with the real ratings and their differences are considered as prediction error. These metrics are calculated as follows:

$$NMAE = \frac{\sum_{i=1}^{n} |r_i - p_i|}{n \times (max - min)} \tag{26}$$

$$NRMSE = \frac{\sqrt{\sum_{i=1}^{n} (r_i - p_i)^2}}{n \times (max - min)} \tag{27}$$

where, r_i and p_i are respectively real and predicted ratings of item i, max and min are respectively maximum and minimum values of ratings in the system, and n is the total number of ratings which are predicted using a recommender method. The lower values of NMAE and NRMSE metrics show the higher prediction accuracy. It should be noted that, the NRMSE

Table 3 The descriptions of the evaluation datasets

Dataset	#Users	#Items	#Ratings	Sparsity (%)
MovieLens	943	1682	100 K	93.69
Netflix	4427	1000	56,136	98.73
BookCrossing	90 K	270 K	1.1 M	99.99
Jester	24,938	100	617 K	75.25

metric squares the error before summing it and tends to castigate the large errors more heavily [32]. On the other hand, the NMAE metric considers each error as equal value.

Moreover, the Kendall Tau correlation metric is used to measure the similarity values for the orderings of two ranked lists which can be calculated using the following equation [12]:

$$\tau = \frac{(C - D)}{\sqrt{(C + D + TR) \times (C + D + TP)}} \tag{28}$$

where, C is the number of items' pairs which the recommender method predicted in the same order as real ratings list of the user (i.e. Concordant pairs), D is the number of items' pairs which the recommender method predicted in the wrong order (i.e. Discordant pairs), TR denotes the number of items' pairs with the same real ratings, and TP refers to the number of items with the same predicted ratings. It is clear that, when there is no discordant pairs of items (i.e. $D = 0$) the Kendall Tau value will be 1. In other words, when the ranking of items in the predicted and the real rating lists are exactly the same, we will have the maximum value for Kendall Tau correlation metric (i.e. $\tau = 1$). On the other hand, for two completely dissimilar item lists (i.e. $C = 0$) the value of Kendall Tau metric will be −1. The higher values of Kendall Tau metric show the higher prediction accuracy for a recommender system.

The CC metric is an evaluation measure to evaluate the performance of recommender methods which refers to the percentage of distinct items in recommendation lists of users [12]. This metric can be calculated as follows:

$$CC = \frac{I_r}{I} \tag{29}$$

where, I_r denotes the total number of distinct items in the $top - N$ recommendation lists of users, and I is the total number of items in a recommender system. The higher values of CC metric indicate the higher performance of a recommender system.

4.3 Parameter setting

There are some parameters in the proposed method which need to be initialized before performing the experiments. The parameter θ is used in the proposed method as a threshold value for the similarity values between the users (i.e. Eq. (3)) which its value is set to $\theta = 0.6$ for all of the used datasets. The parameters α, β, and γ are used in the proposed method as the threshold values for the user-based reliability measure (i.e. Section 3.3), the rating-based reliability measure (i.e. Section 3.6), and the confidence values between the users (i.e. Eq. (25)), respectively. The values of the parameters α and γ are set to $\alpha, \gamma = 0.5$ for the Netflix and BookCrossing datasets, and $\alpha, \gamma = 0.6$ for the MovieLens and Jester datasets. The value of parameter β is set to $\beta = 0.6$ for the Netflix and BookCrossing datasets, and $\beta = 0.7$ for the MovieLens and Jester datasets. The parameter m is used in the proposed method as the maximum number of reliable ratings that can be added to the user's rating profile (i.e. Eq. (14)) which is set to $m = 20$ for the MovieLens and Jester datasets and $m = 30$ for the Netflix and BookCrossing datasets. In all of the experiments, the length of recommendation lists for the users (i.e. $top - N$) is set to 10 for all of the compared methods. For the rest of the compared 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. The experimental results refer to the average of the evaluation metrics over five different runs which in each run, the ratings of training set are randomly selected for all of the used datasets.

4.4 Performance comparison

In this section, the results of experiments are reported to compare the proposed method with several state-of-the-art recommender methods. Moreover, several sparsity levels (i.e. L) for training set are used to indicate the performance of the compared methods for alleviating data sparsity problem in recommender systems. For example, sparsity level $L = 10\%$ means that 90% of all data is randomly selected as the training set and the remaining data is used as the test set. Therefore, the higher values for the sparsity levels lead to consider a less number of ratings as the training set.

Tables 4, 5, 6, and 7 show the results of the experiments based on NMAE and NRMSE measures. In these experiments, different values of sparsity levels (i.e. $L = 10\ \%\ , 20\ \%\ , 50\ \%\ ,$ $80\ \%\ , 90\%$) are used to simulate the performance of the compared recommender methods for data sparsity problem. Table 4 reports the results of experiments based on the MovieLens dataset. As you can see from this table, the proposed method obtains the best results for all of the Lvalues compared to the other methods. Therefore, it can be concluded that the proposed method can alleviate data sparsity problem in comparison with the other recommender systems. The NMAE and NRMSE values for the proposed method are respectively 0.161 and 0.203 at the sparsity level of $L = 10\%$. On the other hand, the second best method (i.e. PPLSA) obtains the values of 0.170 and 0.224 at the sparsity level of $L = 10\%$ for the NMAE and NRMSE measures, respectively. Moreover, the results of experiments for the Netflix dataset based on the NMAE and NRMSE metrics are shown in Table 5. It is shown that the proposed method outperforms the other recommender methods based on these metrics for all of the L values. For example, the NMAE and NRMSE values for the proposed method are 0.193 and 0.246 at the sparsity level of $L = 10\%$, respectively. The PPLSA method obtains the values of 0.199 and 0.251 respectively for the NMAE and NRMSE metrics at the sparsity level of $L = 10\%$ as the second best results. The results of Tables 4 and 5 indicate that the proposed method can significantly outperform other recommender systems especially when the data sparsity problem is considered.

Table 6 shows the results of the experiments for the BookCrossing dataset based on the NMAE and NRMSE metrics over different values of sparsity levels. It can be seen from these results that the proposed method have the lower values of the NMAE and NRMSE metrics for all of the L values compared to the other recommender methods. For the sparsity level of $L =$ 10%, the proposed method obtains the values 0.137 and 0.168 for the NMAE and NRMSE measures, respectively. On the other hand, the second best method (i.e. RSVD) obtains the values 0.141 and 0.186 for the NMAE and NRMSE measures, respectively. The results of the experiments in Table 7 show that the proposed method obtains better performance than the other recommender methods based on the NMAE and NRMSE measures for the Jester dataset. For example, the values of the NMAE and NRMSE measures for the proposed method are respectively 0.165 and 0.181 for the sparsity level of $L = 10\%$. The second best results for the sparsity level of $L = 10\%$ are obtained by the PPLSA method which the values are 0.169 and 0.195 for the NMAE and NRMSE measures, respectively. Moreover, the proposed method obtains the values of 0.374 and 0.432 respectively for the NMAE and NRMSE measures at the sparsity level of $L = 90\%$. The PPLSA method obtains the second best results at the sparsity

Table 4 NMAE and NRMSE values over the *MovieLens* dataset for different sparsity levels (L)

Algorithm	NMAE					NRMSE				
	L = 10%	L = 20%	L = 50%	L = 80%	L = 90%	L = 10%	L = 20%	L = 50%	L = 80%	L = 90%
UCF	0.181	0.182	0.249	0.257	0.268	0.259	0.269	0.425	0.431	0.471
ICF	0.174	0.178	0.245	0.249	0.264	0.246	0.252	0.392	0.418	0.467
MLCF	0.199	0.209	0.227	0.244	0.253	0.336	0.341	0.372	0.412	0.445
IGPE	0.261	0.271	0.313	0.334	0.342	0.353	0.397	0.489	0.538	0.565
UOS	0.194	0.197	0.204	0.233	0.253	0.312	0.336	0.361	0.399	0.432
ULPE	0.294	0.301	0.319	0.337	0.347	0.412	0.436	0.492	0.559	0.569
RACF	0.184	0.187	0.239	0.256	0.272	0.279	0.282	0.386	0.429	0.475
Slope-1	0.182	0.184	0.264	0.289	0.310	0.264	0.273	0.452	0.487	0.526
RSVD	0.173	0.182	0.258	0.267	0.294	0.241	0.263	0.443	0.465	0.498
NMF	0.189	0.193	0.271	0.294	0.323	0.298	0.314	0.474	0.516	0.553
PMF	0.180	0.192	0.246	0.258	0.262	0.251	0.295	0.411	0.452	0.461
TLCF	0.174	0.176	0.218	0.239	0.258	0.245	0.249	0.368	0.406	0.454
MCFV	0.179	0.181	0.207	0.226	0.251	0.251	0.256	0.364	0.394	0.428
PPLSA	0.170	0.175	0.195	0.213	0.239	0.224	0.237	0.283	0.312	0.364
UPEMR	**0.161**	**0.163**	**0.181**	**0.207**	**0.221**	**0.203**	**0.211**	**0.259**	**0.263**	**0.281**

The best results are shown in boldface

Table 5 NMAE and NRMSE values over the *Netflix* dataset for different sparsity levels (L)

Algorithm	NMAE					NRMSE				
	L = 10%	L = 20%	L = 50%	L = 80%	L = 90%	L = 10%	L = 20%	L = 50%	L = 80%	L = 90%
UCF	0.229	0.242	0.363	0.413	0.441	0.300	0.317	0.597	0.634	0.681
ICF	0.221	0.223	0.345	0.381	0.418	0.299	0.301	0.512	0.616	0.653
MLCF	0.222	0.228	0.352	0.375	0.435	0.283	0.291	0.561	0.589	0.669
IGPE	0.231	0.244	0.382	0.397	0.402	0.306	0.315	0.617	0.628	0.642
UOS	0.212	0.213	0.317	0.392	0.423	0.271	0.272	0.489	0.620	0.663
ULPE	0.247	0.259	0.371	0.376	0.387	0.312	0.326	0.605	0.611	0.635
RACF	0.217	0.219	0.359	0.383	0.419	0.278	0.281	0.573	0.618	0.658
Slope-1	0.255	0.263	0.347	0.371	0.398	0.325	0.337	0.547	0.574	0.639
RSVD	0.248	0.261	0.301	0.324	0.356	0.316	0.321	0.445	0.462	0.478
NMF	0.264	0.273	0.292	0.316	0.335	0.357	0.389	0.437	0.446	0.451
PMF	0.259	0.265	0.273	0.291	0.321	0.345	0.371	0.411	0.435	0.439
TLCF	0.216	0.225	0.261	0.288	0.315	0.275	0.308	0.372	0.409	0.431
MCFV	0.208	0.217	0.243	0.272	0.307	0.254	0.279	0.354	0.381	0.416
PPLSA	0.199	0.208	0.235	0.254	0.283	0.251	0.267	0.331	0.362	0.385
UPEMR	**0.193**	**0.197**	**0.224**	**0.241**	**0.257**	**0.246**	**0.253**	**0.305**	**0.348**	**0.361**

The best results are shown in boldface

Table 6 NMAE and NRMSE values over the *BookCrossing* dataset for different sparsity levels (L)

Algorithm	NMAE					NRMSE				
	L = 10%	L = 20%	L = 50%	L = 80%	L = 90%	L = 10%	L = 20%	L = 50%	L = 80%	L = 90%
UCF	0.181	0.183	0.293	0.389	0.407	0.228	0.246	0.416	0.483	0.529
ICF	0.191	0.196	0.321	0.394	0.413	0.264	0.285	0.437	0.487	0.547
MLCF	0.158	0.160	0.275	0.375	0.398	0.207	0.212	0.382	0.465	0.524
IGPE	0.168	0.169	0.281	0.368	0.392	0.215	0.224	0.395	0.453	0.508
UOS	0.235	0.238	0.352	0.427	0.453	0.309	0.324	0.458	0.502	0.563
ULPE	0.153	0.154	0.259	0.332	0.354	0.194	0.202	0.337	0.397	0.436
RACF	0.185	0.187	0.308	0.381	0.397	0.231	0.248	0.419	0.472	0.515
Slope-1	0.221	0.225	0.346	0.418	0.436	0.301	0.318	0.451	0.496	0.556
RSVD	0.141	0.143	0.272	0.361	0.386	0.186	0.195	0.364	0.441	0.491
NMF	0.196	0.201	0.263	0.336	0.361	0.275	0.291	0.349	0.408	0.442
PMF	0.209	0.213	0.268	0.345	0.373	0.286	0.304	0.352	0.426	0.469
TLCF	0.173	0.184	0.254	0.318	0.349	0.225	0.247	0.325	0.382	0.421
MCFV	0.162	0.173	0.241	0.312	0.338	0.213	0.238	0.314	0.377	0.415
PPLSA	0.146	0.159	0.225	0.291	0.325	0.192	0.206	0.303	0.362	0.397
UPEMR	**0.137**	**0.140**	**0.196**	**0.275**	**0.312**	**0.168**	**0.181**	**0.279**	**0.354**	**0.385**

The best results are shown in boldface

Table 7 NMAE and NRMSE values over the *Jester* dataset for different sparsity levels (L)

Algorithm	NMAE					NRMSE				
	L = 10%	L = 20%	L = 50%	L = 80%	L = 90%	L = 10%	L = 20%	L = 50%	L = 80%	L = 90%
UCF	0.219	0.223	0.356	0.518	0.562	0.276	0.287	0.481	0.597	0.656
ICF	0.171	0.176	0.283	0.449	0.483	0.204	0.228	0.413	0.518	0.549
MLCF	0.197	0.201	0.319	0.478	0.526	0.246	0.263	0.451	0.562	0.605
IGPE	0.254	0.260	0.314	0.426	0.458	0.297	0.328	0.435	0.487	0.517
UOS	0.195	0.200	0.317	0.467	0.514	0.229	0.256	0.439	0.554	0.583
ULPE	0.273	0.281	0.329	0.438	0.462	0.343	0.374	0.458	0.493	0.528
RACF	0.212	0.219	0.334	0.492	0.547	0.258	0.279	0.472	0.586	0.624
Slope-1	0.261	0.267	0.391	0.537	0.584	0.325	0.341	0.504	0.614	0.671
RSVD	0.208	0.218	0.271	0.365	0.397	0.251	0.267	0.398	0.456	0.482
NMF	0.215	0.219	0.278	0.381	0.403	0.264	0.283	0.406	0.461	0.489
PMF	0.224	0.228	0.289	0.395	0.417	0.282	0.293	0.419	0.468	0.495
TLCF	0.183	0.207	0.268	0.361	0.392	0.215	0.266	0.375	0.437	0.463
MCFV	0.178	0.185	0.264	0.345	0.387	0.209	0.243	0.369	0.421	0.456
PPLSA	0.169	0.174	0.259	0.332	0.381	0.195	0.217	0.348	0.406	0.443
UPEMR	**0.165**	**0.168**	**0.253**	**0.327**	**0.374**	**0.181**	**0.199**	**0.334**	**0.397**	**0.432**

The best results are shown in boldface

level of $L = 90\%$ which the values are 0.381 and 0.443 for the NMAE and NRMSE measures, respectively. According to the results of Tables 4, 5, 6, and 7, it is shown that the proposed method can be an effective method for recommender systems with sparse user-item rating matrixes.

The experiments are repeated based on the Kendal Tau correlation and catalog coverage measures and the results are shown in Tables 8, 9, and 10. The results of the Kendal Tau correlation and catalog coverage measures are shown in Table 8 for the MovieLens dataset. These results indicate that the proposed method outperforms the other recommender methods under different values of data sparsity levels except for $L = 10\%$ and $L = 20\%$ which obtains the second best results for the Kendal Tau correlation measure. In these cases, the values of the Kendal Tau correlation measure are 0.639 and 0.609 for the sparsity levels of $L = 10\%$ and $L = 20\%$, respectively. On the other hand, the IGPE method obtains the best values of the Kendal Tau correlation measure for the sparsity levels of $L = 10\%$ and $L = 20\%$ which the results are 0.839 and 0.832, respectively. Moreover, the proposed method outperforms other recommender methods based on the coverage measure for all of the considered sparsity levels. The results of experiments based on the Kendal Tau correlation and catalog coverage measures are reported in Table 9 for the Netflix dataset. As it is clear from this table, the proposed method obtains the best results for all of the used data sparsity levels L based on both of the Kendal Tau correlation and catalog coverage measures. For example, the proposed method obtains the value of 0.976 for the Kendal Tau correlation measure at the sparsity level $L = 10\%$. The second best result for the Kendal Tau correlation measure at the sparsity level $L = 10\%$ is obtained by the IGPE method which its value is 0.971.

Finally, the proposed method is compared with the other recommender methods based on the Kendal Tau correlation and catalog coverage measures for the BookCrossing and Jester datasets and their results are reported in Tables 10 and 11, respectively. It can be concluded from Table 10 that the proposed method significantly outperforms the other recommender methods at all of the used data sparsity levels based on the BookCrossing dataset. For example, the values of the Kendal Tau correlation and catalog coverage measures for the proposed method at the sparsity level $L = 10\%$ are 0.892 and 0.564, respectively. On the other hand, the PPLSA method obtains the second best results for the Kendal Tau correlation and catalog coverage measures at the sparsity level $L = 10\%$ which the values are 0.881 and 0.521, respectively. Table 11 shows the results of the experiments for the Jester dataset based on the Kendal Tau correlation and catalog coverage measures and also different levels of the data sparsity. As you can see from these results, the proposed method obtains the best results for both of the used evaluation measures and also all of the data sparsity levels in comparison with the other recommender methods. Therefore, it can be concluded that the proposed method can be effective for the recommender systems especially in sparse user-item rating matrixes.

4.5 Sensitivity analysis of the parameters

In this section, the effect of different values of input parameters is evaluated on the performance of the proposed method based on the used evaluation measures. To this end, the data sparsity level is set to $L = 20\%$ to perform the experiments of sensitivity analysis for the parameters. There are five parameters including θ, α, m, β and γ which are used in the proposed method. The parameter θ is an important parameter that is used as a threshold value for the user similarity measure in Eq. (3). Figure 2 shows the results of different values for the parameter θ on the performance of the proposed method over the NMAE, NRMSE, Kendall

Table 8 Kendall Tau correlation (τ) and catalog coverage (CC) values over the *MovieLens* dataset for different sparsity levels (L)

Algorithm	Kendall Tau correlation					Catalog coverage				
	L = 10%	L = 20%	L = 50%	L = 80%	L = 90%	L = 10%	L = 20%	L = 50%	L = 80%	L = 90%
UCF	0.472	0.376	0.118	0.062	0.055	0.194	0.169	0.128	0.091	0.074
ICF	0.395	0.373	0.103	0.067	0.067	0.213	0.191	0.172	0.158	0.149
MLCF	0.320	0.178	0.083	0.070	0.063	0.429	0.428	0.409	0.362	0.337
IGPE	**0.839**	**0.832**	0.215	0.169	0.104	0.212	0.208	0.201	0.198	0.214
UOS	0.252	0.134	0.055	0.041	0.039	0.534	0.506	0.467	0.377	0.300
ULPE	0.448	0.442	0.146	0.117	0.092	0.208	0.193	0.175	0.162	0.141
RACF	0.391	0.386	0.184	0.135	0.098	0.254	0.243	0.216	0.153	0.118
Slope-1	0.387	0.385	0.162	0.092	0.064	0.103	0.082	0.076	0.064	0.061
RSVD	0.388	0.376	0.136	0.087	0.072	0.218	0.203	0.153	0.109	0.094
NMF	0.425	0.405	0.168	0.107	0.089	0.065	0.043	0.041	0.038	0.034
PMF	0.379	0.374	0.147	0.119	0.091	0.159	0.140	0.138	0.132	0.127
TLCF	0.521	0.493	0.256	0.181	0.124	0.512	0.465	0.432	0.356	0.281
MCFV	0.562	0.531	0.275	0.198	0.147	0.529	0.483	0.456	0.359	0.294
PPLSA	0.598	0.564	0.327	0.231	0.169	0.551	0.508	0.482	0.425	0.356
UPEMR	0.639	0.609	**0.363**	**0.254**	**0.184**	**0.565**	**0.510**	**0.504**	**0.455**	**0.388**

The best results are shown in boldface

Table 9 Kendall Tau correlation (τ) and catalog coverage (CC) values over the *Netflix* dataset for different sparsity levels (L)

Algorithm	Kendall Tau correlation					catalog coverage				
	L = 10%	L = 20%	L = 50%	L = 80%	L = 90%	L = 10%	L = 20%	L = 50%	L = 80%	L = 90%
UCF	0.539	0.468	0.075	0.042	0.041	0.119	0.116	0.123	0.077	0.056
ICF	0.772	0.723	0.533	0.503	0.391	0.135	0.128	0.116	0.102	0.071
MLCF	0.897	0.865	0.485	0.356	0.264	0.144	0.131	0.125	0.071	0.049
IGPE	0.971	0.966	0.648	0.416	0.347	0.112	0.109	0.106	0.093	0.092
UOS	0.732	0.698	0.502	0.297	0.213	0.159	0.154	0.134	0.053	0.022
ULPE	0.639	0.584	0.315	0.251	0.238	0.143	0.138	0.132	0.124	0.106
RACF	0.633	0.573	0.289	0.243	0.216	0.173	0.166	0.153	0.093	0.064
Slope-1	0.658	0.619	0.396	0.268	0.185	0.288	0.279	0.259	0.156	0.106
RSVD	0.796	0.753	0.512	0.364	0.257	0.345	0.331	0.214	0.163	0.098
NMF	0.889	0.835	0.638	0.487	0.345	0.186	0.165	0.123	0.115	0.105
PMF	0.831	0.784	0.582	0.413	0.286	0.314	0.298	0.214	0.186	0.182
TLCF	0.932	0.893	0.612	0.509	0.416	0.326	0.308	0.267	0.205	0.189
MCFV	0.921	0.884	0.582	0.463	0.384	0.376	0.351	0.302	0.243	0.218
PPLSA	0.963	0.952	0.667	0.558	0.435	0.438	0.409	0.367	0.271	0.245
UPEMR	**0.976**	**0.971**	**0.723**	**0.584**	**0.458**	**0.475**	**0.452**	**0.397**	**0.283**	**0.269**

The best results are shown in boldface

Table 10 Kendall Tau correlation (τ) and catalog coverage (CC) values over the *BookCrossing* dataset for different sparsity levels (L)

Algorithm	Kendall Tau correlation					catalog coverage				
	L = 10%	L = 20%	L = 50%	L = 80%	L = 90%	L = 10%	L = 20%	L = 50%	L = 80%	L = 90%
UCF	0.457	0.413	0.192	0.145	0.118	0.282	0.234	0.152	0.103	0.081
ICF	0.421	0.392	0.156	0.106	0.087	0.304	0.251	0.167	0.112	0.087
MLCF	0.486	0.457	0.237	0.162	0.134	0.292	0.268	0.162	0.119	0.093
IGPE	0.874	0.851	0.368	0.201	0.158	0.373	0.327	0.261	0.196	0.172
UOS	0.293	0.254	0.112	0.086	0.062	0.325	0.295	0.213	0.173	0.124
ULPE	0.798	0.732	0.289	0.187	0.141	0.371	0.343	0.276	0.214	0.185
RACF	0.189	0.125	0.092	0.051	0.048	0.248	0.211	0.143	0.101	0.074
Slope-1	0.382	0.312	0.187	0.153	0.101	0.224	0.194	0.126	0.092	0.071
RSVD	0.354	0.285	0.213	0.172	0.125	0.485	0.459	0.382	0.295	0.226
NMF	0.371	0.349	0.254	0.207	0.173	0.437	0.393	0.314	0.263	0.182
PMF	0.349	0.316	0.236	0.192	0.164	0.456	0.417	0.325	0.207	0.176
TLCF	0.752	0.693	0.272	0.245	0.201	0.453	0.412	0.365	0.309	0.243
MCFV	0.861	0.846	0.485	0.328	0.237	0.492	0.465	0.412	0.334	0.261
PPLSA	0.881	0.858	0.516	0.364	0.259	0.521	0.486	0.437	0.362	0.297
UPEMR	**0.892**	**0.865**	**0.548**	**0.397**	**0.272**	**0.564**	**0.518**	**0.456**	**0.382**	**0.317**

The best results are shown in boldface

Table 11 Kendall Tau correlation (τ) and catalog coverage (CC) values over the *Jester* dataset for different sparsity levels (L)

Algorithm	Kendall Tau correlation					catalog coverage				
	L = 10%	L = 20%	L = 50%	L = 80%	L = 90%	L = 10%	L = 20%	L = 50%	L = 80%	L = 90%
UCF	0.346	0.315	0.253	0.164	0.112	0.567	0.547	0.413	0.295	0.189
ICF	0.327	0.298	0.241	0.175	0.126	0.521	0.473	0.384	0.246	0.152
MLCF	0.357	0.305	0.264	0.179	0.143	0.596	0.541	0.462	0.347	0.221
IGPE	0.541	0.497	0.374	0.282	0.195	0.723	0.665	0.592	0.463	0.356
UOS	0.391	0.342	0.282	0.214	0.151	0.498	0.454	0.391	0.282	0.163
ULPE	0.456	0.432	0.313	0.264	0.171	0.745	0.679	0.563	0.426	0.327
RACF	0.384	0.359	0.272	0.216	0.153	0.542	0.512	0.406	0.324	0.249
Slope-1	0.264	0.251	0.175	0.123	0.095	0.613	0.561	0.465	0.318	0.274
RSVD	0.382	0.346	0.298	0.256	0.199	0.672	0.635	0.574	0.482	0.376
NMF	0.421	0.361	0.315	0.259	0.206	0.485	0.416	0.351	0.265	0.208
PMF	0.356	0.320	0.274	0.216	0.168	0.553	0.482	0.417	0.354	0.299
TLCF	0.553	0.504	0.415	0.308	0.217	0.713	0.642	0.584	0.419	0.347
MCFV	0.576	0.521	0.437	0.312	0.245	0.748	0.684	0.612	0.487	0.391
PPLSA	0.612	0.534	0.452	0.361	0.272	0.753	0.705	0.637	0.521	0.416
UPEMR	**0.635**	**0.558**	**0.483**	**0.392**	**0.284**	**0.772**	**0.735**	**0.664**	**0.542**	**0.438**

The best results are shown in boldface

Tau correlation and catalog coverage measures for the MovieLens, Netflix, BookCrossing and Jester datasets. As you can see from this figure, the NMAE and NRMSE values for all of the used datasets decrease when the value of the parameter θ is increased (see Fig. 2a and b). Moreover, the catalog coverage values for all of the used datasets decrease with increasing the value of the parameter θ (see Fig. 2d). On other hand, the Kendall Tau correlation values increase when the value of the parameter θ is increased (see Fig. 2c). Therefore, it can be concluded that the higher value of the parameter θ has a positive effect on the NMAE, NRMSE and Kendall Tau correlation measures. However, the higher value of the parameter θ has a negative effect on the catalog coverage measure. Therefore, the value of the parameter θ is set to $\theta = 0.6$ as a trade-off between the accuracy and coverage measures to compare the proposed method with the other recommender methods.

The parameter α is used as a threshold value in the proposed user profile enhancing mechanism to enhance rating profiles of users (see Section 3.3). In other words, the user's rating profile will be enhanced if the user-based reliability measure for the user is lower than the value of parameter α. The effect of different values of parameter α over the NMAE, NRMSE, Kendall Tau correlation and catalog coverage measures is reported in Fig. 3. As you can see from this figure, the NMAE and NRMSE values decrease in most cases when the value of parameter α is increased. Moreover, the values of NMAE and NRMSE measures increase when the value of parameter α is higher than specific values for all of the used datasets (see Fig. 3a and b). For example, the values of NMAE and NRMSE measures are increased when the value of parameter α is higher than 0.5 for the Netflix dataset. On the other hand, the values of Kendall Tau correlation and catalog coverage measures increase in most

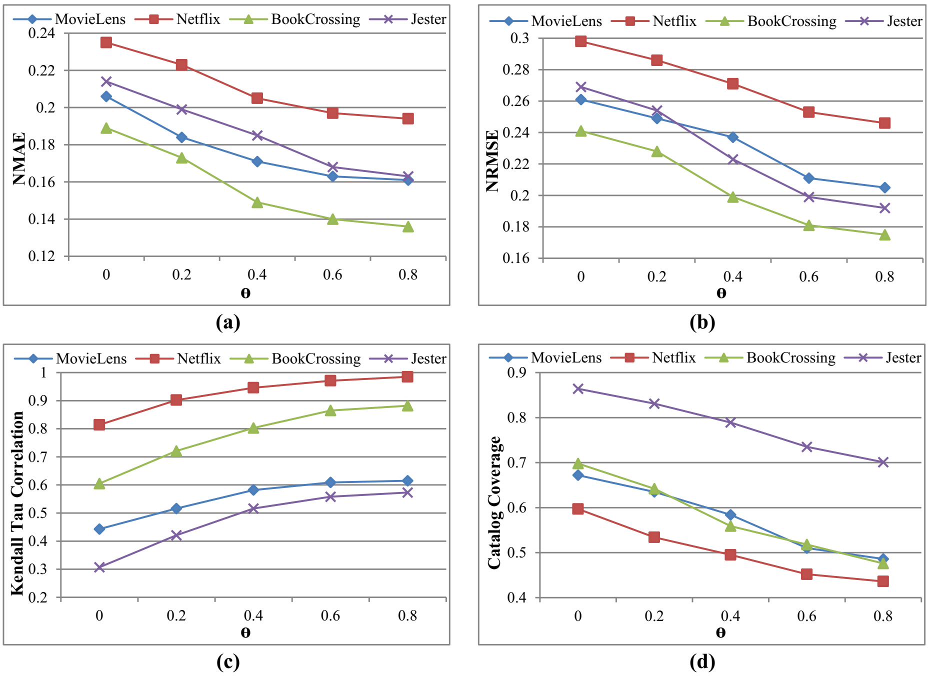

Fig. 2 The effect of parameter θ on the system performance for: **a** NMAE measure, **b** NRMSE measure, **c** Kendall Tau correlation measure, and **d** catalog coverage measure

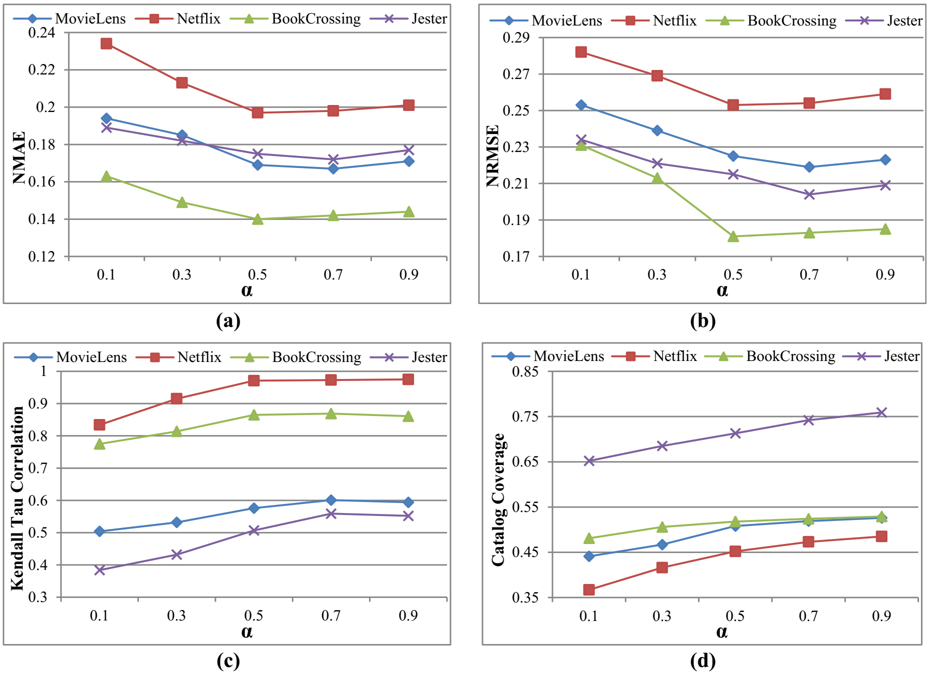

Fig. 3 The effect of parameter α on the system performance for: **a** NMAE measure, **b** NRMSE measure, **c** Kendall Tau correlation measure, and **d** catalog coverage measure

cases when the value of parameter α is increased. It should be noted that, the value of parameter α is set to $\alpha = 0.5$ for the Netflix and BookCrossing datasets and $\alpha = 0.6$ for the MovieLens and Jester datasets to compare the proposed method with the other recommender methods.

The parameter m is used as the maximum number of reliable ratings that can be added to the user's rating profile in the user profile enhancing step of the proposed method (i.e. Eq. (14)). Figure 4 shows the effect of different values of the parameter m over the used evaluation measures for the MovieLens, Netflix, BookCrossing and Jester datasets. As you can see from this figure, the NMAE, NRMSE and Kendall Tau correlation measures have different values for different values of the parameter m. For example, the best results for the NMAE and NRMSE measures are obtained when the value of parameter m is equal to $m = 20$ for the MovieLens and Jester datasets. Moreover, the best values of the NMAE and NRMSE measures for the Netflix and BookCrossing datasets can be obtained when the value of parameter m is equal to $m = 30$. The sensitivity analysis of the parameter m on the Kendall Tau correlation indicates that the best results of the proposed method are obtained when the value of parameter m is equal to $m = 10$ for all of the used datasets. The value of catalog coverage measure increases when the value of parameter m is increased for all of the used datasets. Therefore, it can be concluded that the higher value of parameter m has a positive effect on the proposed method based on the catalog coverage measure for all of the used datasets. In the experiments, the value of parameter m is set to $m = 20$ for the MovieLens and Jester datasets and $m = 30$ for the Netflix and BookCrossing datasets to compare the proposed method with the other recommender methods.

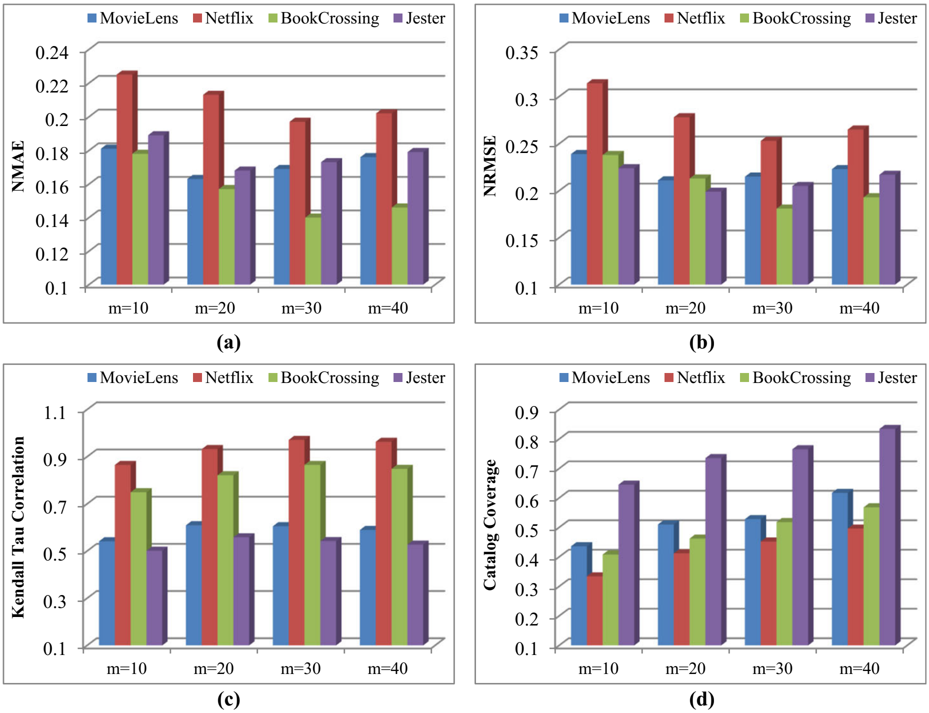

Fig. 4 The effect of parameter m on the system performance for: **a** NMAE measure, **b** NRMSE measure, **c** Kendall Tau correlation measure, and **d** catalog coverage measure

Furthermore, the parameter β is used as a threshold value for the rating-based reliability measure in the final rating prediction step of the proposed method (see Section 3.6). In other words, the initial predicted ratings which their rating-based reliability values are less than the parameter β will be recalculated using the proposed mechanism. The sensitivity analysis of the parameter β over the NMAE, NRMSE, Kendall Tau correlation, and catalog coverage measures are reported in Fig. 5. These results show that the NMAE and NRMSE values decrease in most cases when the value of parameter β is increased. In addition, the values of these measures increase when the value of parameter β is higher than specific values. For example, the values of NMAE measure increase for the BookCrossing dataset when the value of parameter β is higher than 0.5 (see Fig. 5a). On the other hand, the value of Kendall Tau correlation measure increases in most cases when the value of parameter β is increased (see Fig. 5c). Finally, it can be concluded that the higher values of the parameter β have negative effects on the catalog coverage measure (see Fig. 5d). Since, the value of catalog coverage measure decreases when the value of parameter β is increased. It should be noted that, the value of parameter β is set to $\beta = 0.6$ for the Netflix and BookCrossing datasets and also $\beta = 0.7$ for the MovieLens and Jester datasets to compare the proposed method with the other recommender methods.

The parameter γ is used as a threshold value for the proposed confidence measure between the users (i.e. Eq. (25)). In other words, the users in the initial neighbors set of target user who their confidence values are higher than the parameter γ will be considered as the final neighbors set. The effect of different values of the parameter γ over the NMAE, NRMSE, Kendall Tau correlation, and catalog coverage measures are reported in Fig. 6. As you can see

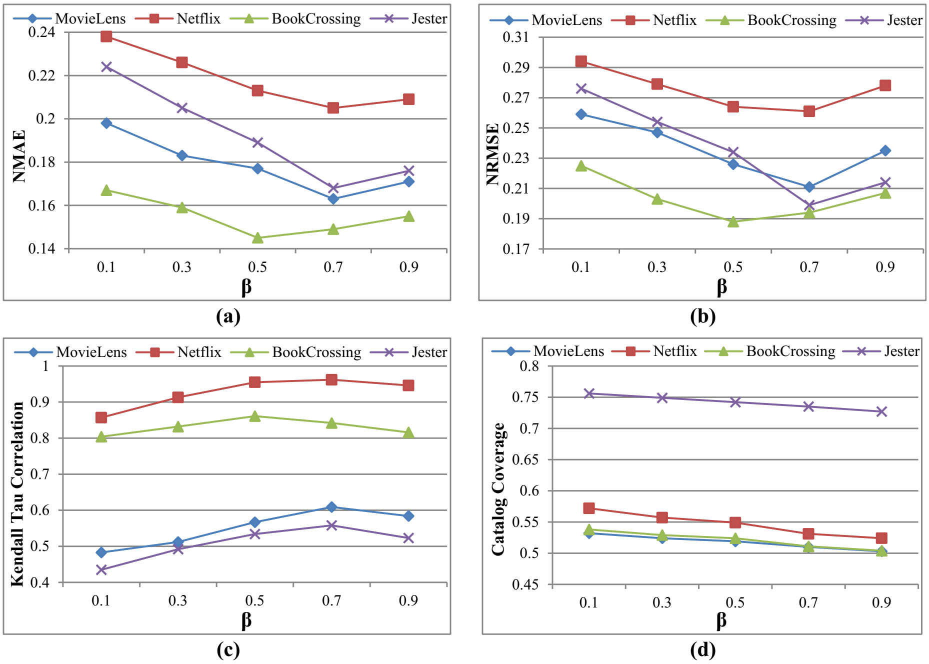

Fig. 5 The effect of parameter β on the system performance for: **a** NMAE measure, **b** NRMSE measure, **c** Kendall Tau correlation measure, and d catalog coverage measure

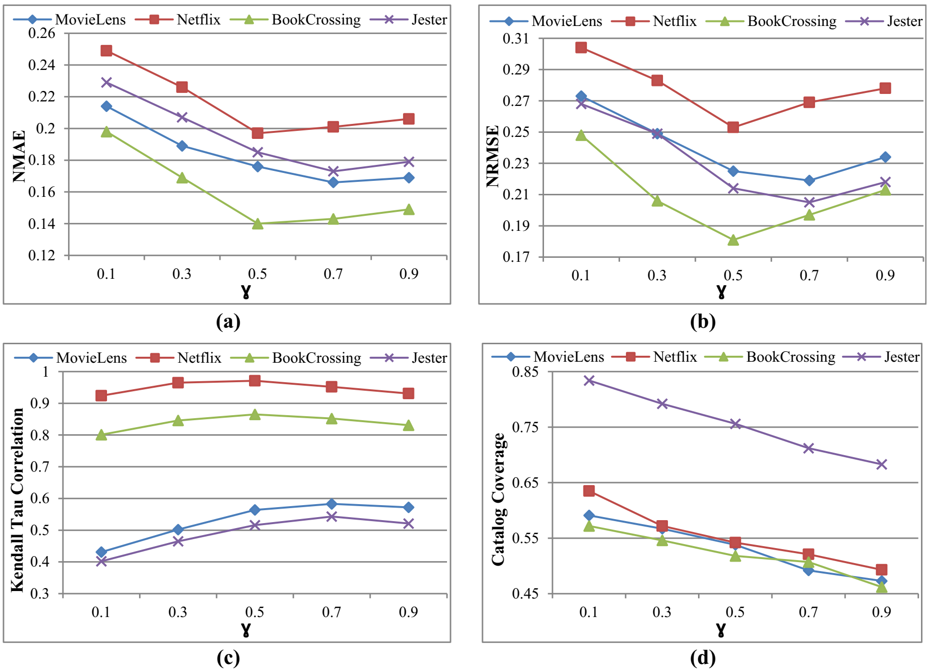

Fig. 6 The effect of parameter γ on the system performance for: **a** NMAE measure, **b** NRMSE measure, **c** Kendall Tau correlation measure, and **d** catalog coverage measure

from this figure, the NMAE and NRMSE values decrease in most cases when the value of parameter γ is increased. Moreover, the values of evaluation measures increase when the value of parameter γ is higher than specific values. On the other hand, the value of Kendall Tau correlation measure increases in most cases when the value of parameter γ is increased (see Fig. 6c). The results indicate that the higher values of parameter γ have negative effects on the catalog coverage measure (see Fig. 6d). In the experiments, the value of parameter γ is set to $\gamma = 0.5$ for the Netflix and BookCrossing datasets and also $\gamma = 0.6$ for the MovieLens and Jester datasets to compare the proposed method with the other recommender methods.

5 Conclusions

Data sparsity is an important problem which the recommender systems suffer from it. This problem has a negative impact on the quality of predictions which leads to reduce the performance of the systems. In this paper, a novel approach is proposed to address this problem. The proposed method is based on three reliability measures which are used to evaluate the reliability of users, items and predicted ratings in the system. The proposed user-based reliability measure is used to evaluate the effectiveness of rating profiles of users. The rating profiles of users with low user-based reliability measure are enhanced by using a novel mechanism which is based on the proposed item-based reliability measure. Moreover, the quality of initial predicted ratings is evaluated using the rating-based reliability measure and the ratings with low reliability values will be recalculated using a novel mechanism. The main advantage of the proposed method is to alleviate data sparsity problem in recommender systems. Experimental results using four real-world datasets indicate that the proposed method achieves higher performance compared to other recommender methods based on different evaluation measures. Future work will be focused on developing new reliability measures based on different factors such as social information in recommender systems. In other words, additional information such as trust and distrust relations between users can be useful to propose novel reliability measures. Also, using noise detection techniques in recommender systems can be effective for modeling reliability-based frameworks for recommending items. Moreover, the proposed method can be used in other types of recommender systems such as context-aware recommender systems to improve their performance.

References

1. Ahmadian S, Joorabloo N, Jalili M, Meghdadi M, Afsharchi M, Ren Y (2018) A temporal clustering approach for social recommender systems. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Barcelona, Spain, p 1139-1144
2. Ahmadian S, Meghdadi M, Afsharchi M (2018) A social recommendation method based on an adaptive neighbor selection mechanism. Inf Process Manag 54(4):707–725
3. Ahmadian S, Afsharchi M, Meghdadi M (2018) An effective social recommendation method based on user reputation model and rating profile enhancement. J Inf Sci. https://doi.org/10.1177/0165551518808191
4. Ahmadian S, Meghdadi M, Afsharchi M (2018) Incorporating reliable virtual ratings into social recommendation systems. Appl Intell 48(11):4448–4469

5. Boratto L, Carta S, Fenu G, Saia R (2017) Semantics-aware content-based recommender systems: design and architecture guidelines. Neurocomputing 254:79–85
6. 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
7. da Silva EQ, Camilo-Junior CG, Pascoal LML, Rosa TC (2016) An evolutionary approach for combining results of recommender systems techniques based on collaborative filtering. Expert Syst Appl 53:204–218
8. Formoso V, Fernández D, Cacheda F, Carneiro V (2013) Using profile expansion techniques to alleviate the new user problem. Inf Process Manag 49:659–672
9. 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
10. Guo G, Qiu H, Tan Z, Liu Y, Ma J, Wang X (2017) Resolving data sparsity by multi-type auxiliary implicit feedback for recommender systems. Knowl-Based Syst 138:202–207
11. He XS, Zhou MY, Zhuo Z, Fu ZQ, Liu JG (2015) Predicting online ratings based on the opinion spreading process. Physica A: Stat Mech Appl 436:658–664
12. Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst 22(1):5–53
13. Hernando A, Bobadilla J, Ortega F, Tejedor J (2013) Incorporating reliability measurements into the predictions of a recommender system. Inf Sci 218:1–16
14. Hernando A, Bobadilla J, Ortega F (2016) A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model. Knowl-Based Syst 97:188–202
15. Hong FX, Zheng XL, Chen CC (2016) Latent space regularization for recommender systems. Inf Sci 360: 202–216
16. Hu J, Zhang Y (2015) Research patterns and trends of recommendation system in China using co-word analysis. Inf Process Manag 51(4):329–339
17. Huang L, Tan W, Sun Y (2018) Collaborative recommendation algorithm based on probabilistic matrix factorization in probabilistic latent semantic analysis. Multimed Tools Appl. https://doi.org/10.1007/s11042-018-6232-x
18. Javari A, Gharibshah J, Jalili M (2014) Recommender systems based on collaborative filtering and resource allocation. Soc Netw Anal Min 4(1)
19. Katarya R, Verma OP (2016) A collaborative recommender system enhanced with particle swarm optimization technique. Multimed Tools Appl 75(15):9225–9239
20. Kim HN, Saddik AE, Jo GS (2011) Collaborative error-reflected models for cold-start recommender systems. Decis Support Syst 51(3):519–531
21. Langseth H, Nielsen TD (2015) Scalable learning of probabilistic latent models for collaborative filtering. Decis Support Syst 74:1–11
22. Lee DD, Seung HS (2001) Algorithms for non-negative matrix factorization. In: Neural Information Processing Systems
23. Lemire D, Maclachlan A (2005) Slope one predictors for online rating-based collaborative filtering. Soc Ind Math 5:471–480
24. Lika B, Kolomvatsos K, Hadjiefthymiades S (2014) Facing the cold start problem in recommender systems. Expert Syst Appl 41(4):2065–2073
25. Liu Y, Nie L, Liu L, Rosenblum DS (2016) From action to activity: sensor-based activity recognition. Neurocomputing 181:108–115
26. Liu Y, Zhang L, Nie L, Yan Y, Rosenblum DS (2016) Fortune teller: predicting your career path. In: AAAI'16 Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona, p 201–207
27. Liu Y, Li L, Liu J (2018) Bilateral neural embedding for collaborative filtering-based multimedia recommendation. Multimed Tools Appl 77(10):12533–12544
28. Luo X, Liu H, Gou G, Xia Y, Zhu Q (2012) A parallel matrix factorization based recommender by alternating stochastic gradient decent. Eng Appl Artif Intell 25(7):1403–1412
29. Mazurowski MA (2013) Estimating confidence of individual rating predictions in collaborative filtering recommender systems. Expert Syst Appl 40:3847–3857
30. Mnih A, Salakhutdinov R (2007) Probabilistic matrix factorization. Adv Neural Inf Process Syst:1257-1264
31. Moradi P, Ahmadian S (2015) A reliability-based recommendation method to improve trust-aware recommender systems. Expert Syst Appl 42:7386–7398
32. 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
33. Nikolakopoulos AN, Kouneli MA, Garofalakis JD (2015) Hierarchical Itemspace rank: exploiting hierarchy to alleviate sparsity in ranking-based recommendation. Neurocomputing 163:126–136

34. Nilashi M, Ibrahim O, Ithnin N (2014) Multi-criteria collaborative filtering with high accuracy using higher order singular value decomposition and neuro-fuzzy system. Knowl-Based Syst 60:82–101
35. 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
36. Polatidis N, Georgiadis CK (2016) A multi-level collaborative filtering method that improves recommendations. Expert Syst Appl 48:100–110
37. Ren Y, Li G, Zhang J, Zhou W (2012) The efficient imputation method for neighborhood-based collaborative filtering. In: Proceedings of the 21st ACM international conference on Information and knowledge management, Maui, Hawaii, USA. ACM, p 684-693
38. Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J (1994) GroupLens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, Chapel Hill, North Carolina, USA. ACM, p 175–186
39. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web ACM. p 285-295
40. Shi Y, Larson M, Hanjalic A (2013) Unifying rating-oriented and ranking-oriented collaborative filtering for improved recommendation. Inf Sci 229:29–39
41. Son LH (2016) Dealing with the new user cold-start problem in recommender systems: a comparative review. Inf Syst 58:87–104
42. Su X, Khoshgoftaar TM, Zhu X, Greiner R (2008) Imputation-boosted collaborative filtering using machine learning classifiers. In: Proceedings of the 2008 ACM symposium on Applied computing, Fortaleza, Ceara, Brazil. ACM, p 949-950
43. Webb B (2006) Netflix update: Try this at home. http://sifter.org/~simon/journal/20061211.html
44. Xia W, He L, Gu J, He K (2009) Effective collaborative filtering approaches based on missing data imputation. In: Fifth International Joint Conference on INC, IMS and IDC, Seoul, South Korea. IEEE
45. Xie F, Chen Z, Shang J, Fox GC (2014) Grey forecast model for accurate recommendation in presence of data sparsity and correlation. Knowl-Based Syst 69:179–190
46. Yang Z, Chen W, Huang J (2018) Enhancing recommendation on extremely sparse data with blocks-coupled non-negative matrix factorization. Neurocomputing 278:126–133
47. Yera R, Castro J, Martínez L (2016) A fuzzy model for managing natural noise in recommender systems. Appl Soft Comput 40:187–198
48. Yu L, Han F, Huang S, Luo Y (2018) A content-based goods image recommendation system. Multimed Tools Appl 77(4):4155–4169
49. Zhang M, Guo X, Chen G (2016) Prediction uncertainty in collaborative filtering: enhancing personalized online product ranking. Decis Support Syst 83:10–21
50. Zhang H, Ge D, Zhang S (2018) Hybrid recommendation system based on semantic interest community and trusted neighbors. Multimed Tools Appl 77(4):4187–4202
51. Zhang Z, Liu Y, Jin Z, Zhang R (2018) A dynamic trust based two-layer neighbor selection scheme towards online recommender systems. Neurocomputing 285:94–103
52. Zhang HR, Min F, Zhang ZH, Wang S (2018) Efficient collaborative filtering recommendations with multi-channel feature vectors. Int J Mach Learn Cybern. https://doi.org/10.1007/s13042-018-0795-8

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, artificial intelligence from University of Kurdistan, Sanandaj, Iran in 2014. Moreover, he has been received his Ph.D. degree in computer engineering, artificial intelligence from University of Zanjan, Zanjan, Iran, in 2018. He conducted a part of his Ph.D. research work in the laboratory of complex networks, RMIT University, Melbourne, Australia, form February 2018 to July 2018. His research interests include recommender systems, social networks analysis, data mining and machine learning.

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.

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 Associate Professor of the Department of Computer Engineering, University of Zanjan, Zanjan, Iran.