Multimedia Tools and Applications (2023) 82:38275–38296
https://doi.org/10.1007/s11042-023-15141-2

An effective profile expansion technique based on movie genres and user demographic information to improve movie recommendation systems

Samad Mohamadi [1] · Vahe Aghazarian [1] · Alireza Hedayati [1]

Received: 6 October 2021 / Revised: 10 January 2023 / Accepted: 13 March 2023 /
Published online: 25 March 2023

Abstract
Movie recommendation systems are efficient tools to help users find their relevant movies by investigating the previous interests of users. These systems are established on considering the ratings of users provided for movies in the past and using them to predict their interests in the future. However, users mainly provide insufficient ratings leading to make a problem called data sparsity. This problem makes reducing the effectiveness of movie recommendation systems. On the other hand, other available data such as genres of movies and demographic information of users play a vital role in assisting recommenders in order to better produce recommendations. This paper proposes a movie recommendation method utilizing the movies' genres and users' demographic information. In particular, we propose an effective model to evaluate the user's rating profile and determine the minimum number of ratings required to produce an accurate prediction. Then, appropriate virtual ratings are incorporated into the profiles with insufficient ratings to expand them. These virtual ratings are calculated using similarity values between users obtained by genres of movies and demographic information of users. Furthermore, an effective measure is introduced to determine how much an item is reliable. This measure guarantees the virtual ratings' reliability. Finally, unknown ratings for target user are predicted based on the expanded rating profiles. Experiments performed on two well-known movie recommendation datasets demonstrate that the proposed approach is more efficient than other compared recommenders.

Keywords Recommendation systems · Movie · Data sparsity · Demographic information · Genre

✉ Vahe Aghazarian
v_aghazarian@iauctb.ac.ir

Samad Mohamadi
s.mohamadi@iauctb.ac.ir

Alireza Hedayati
hedayati@iauctb.ac.ir

[1] Department of Computer Engineering, Islamic Azad University, Central Tehran Branch, Tehran, Iran

1 Introduction

In recent years, various commercial and non-commercial websites have released a large amount of information and products on the Internet. This issue makes users to be confused by a vast amount of information in order to find their relevant information. Recommendation systems play an important role in helping users find their relevant information by recommending information of potential interest to them [28, 33]. The main purpose of these systems is to investigate the behavior of users in the past and extract appropriate knowledge to predict their interests in the future. Generally, recommendation systems attempt to reach two main objectives. The first one is to satisfy users in the system by providing the most relevant recommendations to them. It is due to the fact that users like to automatically receive the most relevant recommendations from the system to prevent wasting their time. The second objective is to enhance the satisfaction level of e-commerce owners by increasing the amount of profit through cross-selling [43, 45].

Basically, recommendation methods can be categorized into three main approaches based on the way they use to produce recommendations for users. These approaches include collaborative filtering (CF) [42], content-based [32], and hybrid models [24]. CF is a popular approach employed in designing recommenders which its main idea is to utilize ratings data to predict users' preferences. This approach can be performed in two different scenarios including memory-based CF and model-based CF. In the memory-based CF, first of all, similarity measurements including Pearson correlation coefficient, Cosine, and Jaccard are utilized to determine the similarity value between users/items and form a set of nearest neighbors for target user. Then, the unknown ratings are predicted according to the preferences of the neighbors of target user. The memory-based CF models can also benefit from using a heterogeneous parameterized similarity measure, which can capture hidden relationships between users [10]. Model-based CF utilizes machine learning techniques to construct a model to predict the unknown ratings of users. Various learning models have been used in recommendation systems including clustering approaches, matrix factorization, singular value decomposition, and so on [27, 34, 47]. Content-based models utilize content information of items in the recommendation process. The content information is related to the products used in the recommendation system. For instance, in a movie recommendation system, the content information can be the genres of movies provided in the system. Hybrid models combine the characteristics of two or more other models to produce more accurate recommendations.

One of the main resources used in recommendation systems to produce recommendations is user-item ratings matrix. The efficiency of recommendation systems directly depends on the quality of user-item ratings matrix. In other words, if users provide sufficient ratings in the system, it leads to produce more accurate recommendations [2, 4]. However, users mainly tend to assign ratings to a few items while there are usually enormous items in the system. This issue is known as one of the most common problems in recommendation systems which is called data sparsity. It is worth noting that data sparsity problem has a great impact on diminishing the effectiveness of recommendation systems. Many researches have been conducted in the literature to develop appropriate approaches in order to alleviate data sparsity problem [5, 6, 38]. One of the main ideas of these approaches is to use additional data such as trust relationships between users [1, 3], tag information [11], and review information [22] in the recommendation process. In a movie recommendation system, some additional data related to movies and users exist in the system that can be utilized to deal with the data sparsity problem. These data include the genres of movies and demographic information of users. Nevertheless, it is a challenging issue how to employ such additional data in movie recommendation systems [9, 40]. One of the practical solutions to address the data sparsity problem is to expand the rating profiles of users by adding implicit ratings. In the literature, a number of researches have been conducted to develop effective methods to deal with the data sparsity problem by expanding the rating profiles of users. However, it should be noted that some rating profiles contain sufficient explicit ratings and incorporating additional implicit ratings into them may lead to decline their performance in making recommendations. Therefore, it is a critical issue in the recommender systems to identify whether a rating profile requires to be expanded. On the other hand, the reliability of implicit ratings used in the expansion mechanisms should be taken into account to avoid adding ineffective ratings to the rating profiles of users. By investigating the previous works [9, 12, 35, 41], it can be inferred that most of them have neglected to incorporate the reliability measurement into the movie recommendation procedure.

We propose a movie recommender system in this paper by employing the genres of movies and demographic information of users to address the above-mentioned challenges. To this end, first of all, a model is developed in order to determine whether the target user's rating profile is appropriate to produce accurate recommendations or not. In other words, the developed model determines how many ratings are required for each user to generate an accurate prediction with a high probability. This criterion is used to demonstrate that a rating profile contains sufficient ratings for producing reliable recommendations or not. Then, the quality of rating profiles containing insufficient ratings is boosted using an effective profile expansion technique which incorporates some virtual ratings to these profiles. These virtual ratings are calculated using the similarity values between users which are computed according to the genres of movies and demographic information of users. Moreover, the reliability values of users and items are calculated using appropriate reliability measurements to guarantee that the incorporated virtual ratings are reliable. Experimental results on two movie recommendation datasets indicate the superiority of the proposed approach in respect to other models. In the following, we provide a list of the main contributions of this paper:

- We develop a model in order to evaluate the users' rating profiles and determine how many ratings are required for generating an accurate prediction.
- We propose a powerful profile expansion technique which incorporates some virtual ratings to user-item ratings matrix for improving its quality.
- Movies' genres and users' demographic information are used as additional data in the proposed movie recommender system.
- The reliability measures of users and items are used in the proposed method to guarantee the reliability of calculated virtual ratings.
- The proposed method generates a denser user-item ratings matrix than the original matrix which results in alleviating data sparsity problem significantly.

The remaining parts of this paper are structured as follows: in section 2, related works are investigated, section 3 includes the details of the proposed method, section 4 refers to the discussion of experimental results, and section 5 provides some conclusions about the paper.

2 Related works

Many researches have been conducted in the literature to develop movie recommender systems [21, 39, 41]. A hybrid movie recommendation method is proposed in [41] by utilizing collaborative filtering, content-based filtering, and a fuzzy expert system to provide a list of relevant movies for users. In particular, the fuzzy expert system is used to determine the importance of movies recommended to user as a recommendations list leading to provide more accurate recommendations list. In [39], deep autoencoder neural network is utilized to develop a social movie recommendation approach. This system is designed based on a hybrid model utilizing collaborative filtering, content-based filtering, and social information of users. In [21], the authors focused on developing a movie recommender system based on data clustering and computational intelligence concepts. To this end, the k-means clustering algorithm is improved by cuckoo search optimization approach and utilized to make recommendations for users. Indira et al. [19] employed some machine learning algorithms to propose an effective movie recommendation method. Accordingly, first of all, the input dataset is cleaned by removing the existing noises leading to make a more reliable dataset. Then, a feature selection approach based on principle component analysis model is used to select an appropriate subset of features for performing k-means clustering approach. Finally, a list of relevant movies is generated for each user based on the obtained clusters. Gan et al. [15] showed that the timeliness of users' preferences and their sequential characteristics play an important role in increasing the accuracy of movie recommender systems. Therefore, they proposed a user movie interest space model called UMIS by considering three indexes for describing different patterns of users' preferences. Moreover, a deep learning approach is applied to the UMIS model to generate movie recommendations. In [13], a movie recommendation method is proposed based on a user positive profile and a user negative profile which are created using two movie lists where one of them contains the movies that the user likes and another list contains the movies that the user does not like. The main purpose of this method is to make a recommendations list for each user in which the movies are most similar to the positive profile as well as most different from the negative profile. Roy et al. [35] addressed the cold start problem related to the new movies that there are no sufficient ratings assigned to them. To this end, they proposed a hybrid movie recommendation approach by utilizing the genres of movies for computing the similarity values between movies and making recommendations for users.

Ortega et al. [30] introduced a matrix factorization method for recommender systems which is able to calculate not only predictions values, but also it can calculate reliability values. The authors have shown that their proposed method contains significant advantages in boosting the effectiveness of recommenders. [26] incorporates reliability concept into a trust-aware recommender system in order to evaluate the trust networks of users and identify ineffective users who have negative effect on calculating unknown ratings. In [18], two efficient components including users' reliability and influence propagation are utilized to develop a social affect model for recommender systems. In particular, the users' reliability and influence propagation components are used to make a user-shared feature space to produce recommendations for users. [49] proposes a reliability measurement in order to calculate the reliability of users' profiles by investigating the possible conflicts in the profiles of neighbors of the target user. The interests of users are grouped using k-means clustering algorithm to find unreliable profiles in the neighborhood of users. Jiang et al. [20] utilized a hypergraph random walk approach by considering weighting models on hyperedges and vertices to make recommendations through a ranking process. Moreover, trust relationships between users and reliability measure are employed into constructing the hypergraph leading to an improvement in the prediction accuracy. In [7], the authors addressed the shilling attacks in recommender systems by proposing a matrix factorization approach based on a reliability measure which demonstrates that how much a prediction is reliable. Accordingly, the shilling attacks can be detected by investigating the predicted ratings that their reliability values are unusual. The authors experimentally showed that their proposed model is able to detect most of the existing attacks in recommender systems. Deep learning-based models are utilized in [12] to design a movie recommendation approach by achieving users' representations based on different aspects. Also, different fusion functions are deployed to integrate the obtained representations and make movie recommendations. In [37], the authors employed multimodal heterogeneous information networks to propose a movie recommendation model by learning video embedding and achieving appropriate context among videos. Huang et al. [16] showed that incorporating explanations into the recommendation process can improve the effectiveness of sequential recommender systems. They utilized knowledge graphs to find semantic paths between users and items in the system. Moreover, a path-wise explanation for each path is provided for the recommendation according to the semantic paths.

The main differences between the proposed method and other related works discussed above are as follows. First, most of the related movie recommendation models did not consider the genres of movies and demographic information of users in their recommendation procedure. Although these data resources are very helpful for movie recommendation methods, the previous works mainly focused on other data resources such as the user-item rating matrix. Second, unlike previous works that make movie recommendations based on the original user-item rating matrix, the proposed method uses an improved rating matrix expanded by adding virtual ratings. This leads to an enhancement in the quality of movie recommendations specially for those users whose rating profiles do not contain sufficient explicit ratings. Therefore, the proposed method can be very effective in addressing the data sparsity problem in movie recommender systems. Third, the previous works mainly made movie recommendations based on the predicted ratings without considering the reliability measurement in the recommendation process. Whilst, the proposed method evaluates the reliability of users and items using two effective measures and deploys them in the rating profile expansion mechanism. The main purpose of the proposed reliability measures is to guarantee the reliability of the virtual ratings leading to a better movie recommendation procedure.

3 Proposed method

This section provides the details of the proposed movie recommendation method which is based on a Genre and Demographic information based Profile Expansion technique. We name the proposed method as GDPE which includes five steps: (1) rating profile evaluation (2) calculating reliability of users (3) calculating reliability of items (4) rating profile expansion, and (5) making recommendations. A general schema of GDPE method is provided in Fig. 1. The details of each step of the proposed GDPE model are discussed in the following subsections.

3.1 Rating profile evaluation

The number of available ratings in the user's rating profile plays a critical role in producing accurate predictions for unseen items. The performance of recommender systems depends on

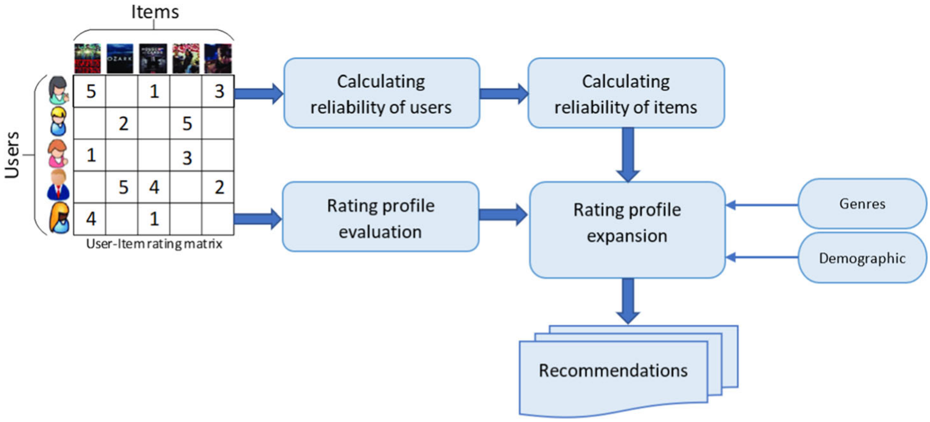


Fig. 1 The general schema of the proposed GDPE model

existing sufficient ratings in the rating profiles of users. Accordingly, one should find that whether the number of ratings in a rating profile is sufficient to produce accurate predictions or not. To address this issue, in this section, an efficient probabilistic methodology is developed for investigating the effectiveness of user's rating profile based on the assigned ratings to items. In particular, this methodology determines that how many ratings are required for predicting unknown ratings accurately. Thus, the number of ratings in a rating profile is compared with the computed number of required ratings, and if the number of existing ratings is lower than the calculated minimum value for a rating profile, it will be considered as an unreliable rating profile which should be expanded by the proposed rating profile expansion technique.

Suppose that U and I indicate users set and items set in the system, respectively. Also, the ratings assigned by users to items are in the range of $[minR,\ maxR]$. The probability of assigning a rating $r \in [minR,\ maxR]$ by user u is defined as:

$$\alpha_{u,r} = \Pr\left[r_{u,i} = r\right] = \frac{n_{u,r}}{|I_u|} \tag{1}$$

where $r_{u,\,i}$ is the item i's rating which is made by user u, the number of ratings assigned by user u in the level r is indicated by $n_{u,\,r}$, and I_u refers to those items rated by user u. To compute the true rating provided by user u, the majority rule is defined as follows:

$$l_u = \max_{r \in [minR, maxR]} \left\{\alpha_{u,r}\right\} \tag{2}$$

where the true rating for user u is denoted by l_u. Then, according to Eq. (3), we can calculate the minimum number of ratings which are required for producing an accurate prediction for user u:

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

where δ is a predefined threshold value which determines the level of reliability in producing accurate predictions, and the second largest value of $\alpha_{u,\,r}$ is denoted by $\widetilde{\alpha}_u$, which is obtained as follows:

$$\widetilde{\alpha}_u = \max_{r\in[minR,maxR]} \left\{\alpha_{u,r} | r \neq l_u\right\} \tag{4}$$

Considering Eq. (3), if $|I_u| \geq n_u^{'}$, then $\Pr\left[\widehat{l}_u = l_u\right] \geq 1-\delta$ [46], where $\widehat{l}_u$ is the predicted rating for user u. Eq. (3) is theoretically proofed in the following.

Lemma 1 *$Pr\left[r_{u,i} = r\right] = \alpha_{u,r}$ denotes the probability mass function (pmf) of $r_{u,i}$, where $r \in [minR, maxR]$, $u \in U$, and $i \in I_u$.*

Proof of Lemma 1 Suppose that the probability of ratings provided by user u in different levels is indicated by $\rho = (\rho_{u,\, minR}, \ldots, \rho_{u,\, maxR})$ which follows Dirichlet distribution. Also, ρ denotes an example which is randomly drawn through $Dirichlet(\alpha_{u,\, r})$. Therefore, $Pr[r_{u,\, i} = r|\ \rho] = \rho_{u,\, r}$ can be considered as the conditional pmf of $r_{u,\, i}$, which denotes a fundamental feature of Dirichlet distributions as $Pr[r_{u,\, i} = r] = \int Pr\ [\rho]\ Pr\ [r_{u,\, i} = r|\ \rho]d\rho = \int Pr\ [\rho]\rho_{u,\, r}d\rho = \alpha_{u,\, r}$ [8].

Theorem 1 Let $X = X_1 + X_2 + \ldots + X_n$ and $\sigma^2 = Var[X] = \sum_{i=1}^{n} Var[X_i]$, where $X_1, X_2, \ldots, X_n$ are n random variables that their values are in the range [0,1]. Therefore, two states can be gained for each $\tau \geq 0$, which are included $Pr[X \geq E[X] + \tau] < exp\left(-\frac{\tau^2}{2(\sigma^2+\tau/3)}\right)$ and $Pr[X \leq E[X] - \tau] < exp\left(-\frac{\tau^2}{2(\sigma^2+\tau/3)}\right)$ [25].

Proof of Eq. (3) To denote that $\Pr\left[\widehat{l}_u \neq l_u\right] \leq \delta$, we employ *Theorem 1*. Therefore, the following equation can be defined by considering the basic probability arguments:

$$Pr\left[\widehat{l}_u \neq l_u\right] = Pr\left[\bigcup_{r\neq l_u} \left\{\widehat{l}_u = r\right\}\right] \leq \sum_{r\neq l_u} Pr\left[\widehat{l}_u = r\right] \leq \sum_{r\neq l_u} Pr\left[n_{u,r} \geq n_{u,l_u}\right] \tag{5}$$

Furthermore, a set of random variables shown by $R_{u,i}^r$ are obtained as follows:

$$R_{u,i}^r = \begin{cases} 1, & with\ probability\ Pr\left[r_{u,i} = r\right] \\ 0, & with\ probability\ Pr\left[r_{u,i} = l_u\right] \\ 1/2, & otherwise \end{cases} \tag{6}$$

where the pmf of $r_{u,\, i}$ is computed through *Lemma 1*, $r \in [minR, maxR]$, $u \in U$, and $i \in I_u$. It can be proved that if $R_u^r = \sum_{i\in I_u} R_{u,i}^r$, then $R_u^r = n_{u,r} + \left(I_u - n_{u,r} - n_{u,l_u}\right)/2$. Thus, the following equation can be defined:

$$R_u^r \geq \frac{I_u}{2} \Leftrightarrow n_{u,r} + \frac{I_u - n_{u,r} - n_{u,l_u}}{2} \geq \frac{I_u}{2} \Leftrightarrow n_{u,r} \geq n_{u,l_u} \tag{7}$$

Moreover, it can be inferred that $Pr\left[n_{u,r} \geq n_{u,l_u}\right] = Pr\left[R_u^r \geq \frac{I_u}{2}\right]$. Accordingly, we should indicate that $\Pr\left[R_u^r \geq \frac{I_u}{2}\right] \leq \frac{\delta}{maxR - minR}, \forall r \neq l_u$ to prove Eq. (3), meaning that $\Pr\left[\widehat{l}_u \neq l_u\right] \leq \sum_{r\neq l_u} \Pr\left[n_{u,r} \geq n_{u,l_u}\right] = \Pr\left[R_u^r \geq \frac{I_u}{2}\right] \leq \delta$. Suppose $Var\left[R_u^r\right] = \left(\alpha_{u,l_u} + \alpha_{u,r} - \left(\alpha_{u,l_u} - \alpha_{u,r}\right)^2\right)\frac{I_u}{4}$ is variance of R_u^r and $E\left[R_u^r\right] = \frac{I_u}{2} - \frac{\left(\alpha_{u,l_u} - \alpha_{u,r}\right)I_u}{2}$ refers to the expectation. Therefore, *Theorem 1* is applied to $\tau = \frac{\left(\alpha_{u,l_u} - \alpha_{u,r}\right)I_u}{2}$ to obtain the following equation:

$$Pr\left[R_u^r \geq \frac{I_u}{2}\right] = Pr\left[R_u^r \geq E\left[R_u^r\right] + \tau\right] \leq exp\left(-\frac{\tau^2}{2\left(Var\left[R_u^r\right] + \frac{\tau}{3}\right)}\right)$$
$$= exp\left(-I_u \times \left(\frac{2\left(\alpha_{u,l_u} + \alpha_{u,r}\right)}{\left(\alpha_{u,l_u} - \alpha_{u,r}\right)^2} - 2 + \frac{4}{3\left(\alpha_{u,l_u} - \alpha_{u,r}\right)}\right)^{-1}\right) \tag{8}$$
$$\leq exp\left(-I_u \times \left(\frac{2\left(\alpha_{u,l_u} + \widetilde{\alpha}_u\right)}{\left(\alpha_{u,l_u} - \widetilde{\alpha}_u\right)^2} - 2 + \frac{4}{3\left(\alpha_{u,l_u} - \widetilde{\alpha}_u\right)}\right)^{-1}\right) \leq \frac{\delta}{(maxR - minR)}$$

Eq. (8) results in proving Eq. (3). It should be noted that, the proposed probabilistic methodology is used to determine that how much a rating profile is effective. For each user u, if $|I_u| < n_u^{'}$, then this profile is considered as an unreliable profile which must be expanded. Therefore, the calculated $n_u^{'}$ can be used in the proposed profile expansion technique to determine that whether a rating profile should be expanded or not.

3.2 Calculating reliability of users

This section introduces a reliability measure according to three different factors which is used in the next step to calculate the reliability of items. For this purpose, at first, the used three factors are determined, and then they are combined using a mechanism to calculate the reliability value. To calculate the reliability of user u, the number of ratings in the user u's profile is considered as the first factor. It should be noted that providing further ratings by the user results in increasing the reliability of predictions in recommender systems. Thus, there is a direct relationship between the value of reliability for a user and the number of ratings in their profile. Eq. (9) calculates the first factor of the reliability of user u as follows:

$$f_i(I_u) = 1 - \frac{\overline{u}}{\overline{u} + |I_u|} \tag{9}$$

where I_u refers to the items set rated by user u, and $\overline{u}$ indicates the median of the values for $|I_u|$.

In addition to the ratings assigned by user u, the ratings that have been expressed by the users in their nearest neighbors set have a significant impact on the user u's reliability. Thus, we consider this important criterion as the second factor of the reliability of user u which has a positive effect on the reliability value as considering further ratings in the recommendation process results in increasing the predictions' reliability. The nearest neighbors set is constructed using the similarity values between the target user and other users. A subset of most similar users is selected as the nearest neighbors set. To this end, the similarity value between users u and v is measured using the Pearson correlation coefficient function as follows:

$$sim(u,v) = \frac{\sum_{i \in I_{u,v}} \left(r_{u,i} - \overline{r}_u\right)\left(r_{v,i} - \overline{r}_v\right)}{\sqrt{\sum_{i \in I_{u,v}} \left(r_{u,i} - \overline{r}_u\right)^2} \sqrt{\sum_{i \in I_{u,v}} \left(r_{v,i} - \overline{r}_v\right)^2}} \tag{10}$$

where the rating of item i assigned by user u is denoted by $r_{u,i}$, the average of ratings provided by user u is denoted by $\overline{r}_u$, and $I_{u,v}$ implies the set of common items for users u and v. After computing the similarity values, the nearest neighbors are determined by selecting N users with the highest similarity values. Then, the second factor is defined using the number of ratings provided by the nearest neighbors as follows:

$$f_{i_{NN}}(I_{NN_u}) = 1 - \frac{\overline{i}_{NN}}{\overline{i}_{NN} + |I_{NN_u}|} \tag{11}$$

where NN_u represents the nearest neighbors of user u, I_{NN_u} refers to the items set rated by the nearest neighbors, $|I_{NN_u}|$ denotes the number of ratings generated by the nearest neighbors, and $\overline{i}_{NN}$ implies the median of the values of $|I_{NN_u}|$.

The third factor used in the reliability of users is defined regarding the similarity values between the target user and the nearest neighbors. The higher similarity values result in increasing the reliability of predictions in recommender systems. Thus, this factor is defined according to the summation of similarity values using the following equation:

$$f_s(S_u) = 1 - \frac{\overline{s}}{\overline{s} + S_u} \tag{12}$$

where,

$$S_u = \sum_{v \in NN_u} sim(u, v) \tag{13}$$

and $\overline{s}$ refers to the median of the values of S_u.

To compute the user u's reliability, the defined factors are combined using a weight-based mechanism. By investigating Eqs. (9), (11), and (12), we can conclude that the first and second factors are independently computed according to the number of items rated by user u and the number of ratings generated by the nearest neighbors, respectively. Thus, to compute the user u' reliability, the weights of the first and second factors are set to 1. The third factor depends on the value of the second factor. Specifically, the higher value of the second factor increases the value of the third factor, and vice versa. Accordingly, to calculate the reliability of the target user u, the weight of the third factor is determined as the value of the second factor. Therefore, the reliability of the target user u is calculated using the geometric average of the defined factors as follows:

$$UR_u = \left[f_i(I_u) . f_{i_{NN}}(I_{NN_u}) . f_s(S_u)^{f_{i_{NN}}(I_{NN_u})} \right]^{\frac{1}{2 + f_{i_{NN}}(I_{NN_u})}} \tag{14}$$

where UR_u refers to the reliability of the target user u. Figure 2 illustrates the main procedure of the calculation of the reliability of users.

3.3 Calculating reliability of items

In this section, the reliability of items is evaluated using a measure according to three different factors. To this end, these factors are defined and then a combination of them is considered as the reliability of items. The first factor is determined according to the number of ratings generated for the item by the users. Assigning further ratings to an item results in increasing its

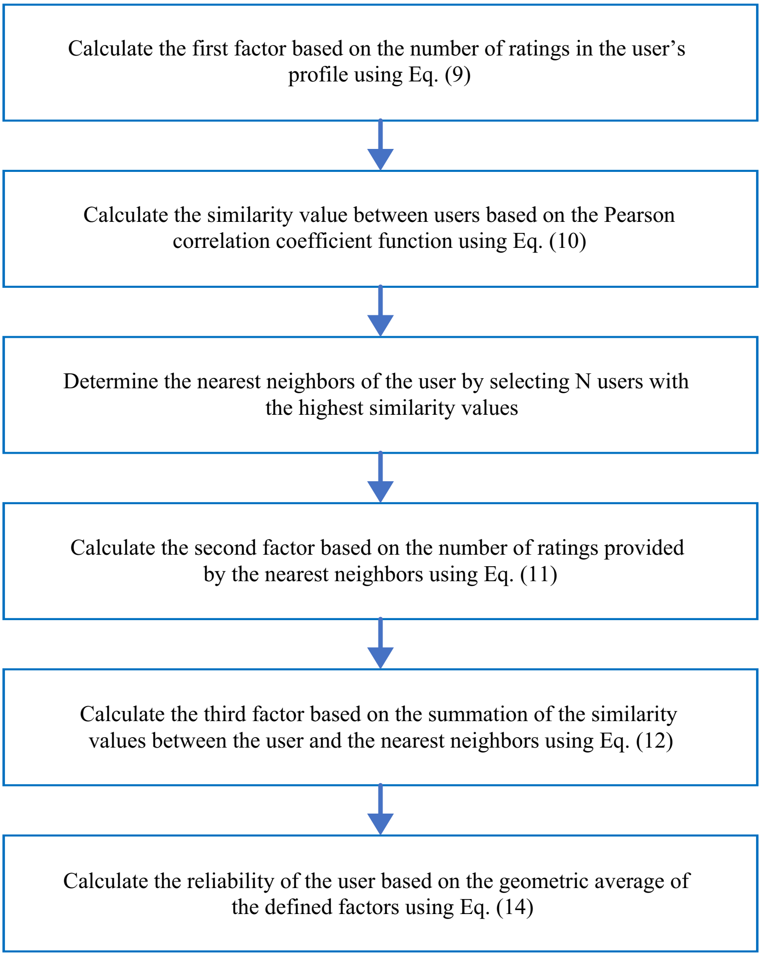


Fig. 2 The flowchart of the calculation of the reliability of users

reliability. Thus, the first factor has a positive effect on the item's reliability. The following equation is used to define the first factor:

$$f_i(I_i) = 1 - \frac{\bar{i}}{\bar{i} + |I_i|} \tag{15}$$

where $|I_i|$ implies the number of ratings generated for item i, and $\bar{i}$ refers to the median of the values of $|I_i|$.

The second factor used in calculating the item's reliability is the standard deviation of ratings which are assigned to the target item i by the users. The main purpose of this factor is to measure the differences in the opinions of users on the target item in which the higher value of this factor results in decreasing the reliability value. Therefore, this factor impacts negatively the value of reliability of the target item and is computed using Eq. (16) as follows:

$$f_{sd}(stdev(I_i)) = \frac{max - stdev(I_i)}{max - min} \tag{16}$$

where $stdev(I_i)$ denotes the value of standard deviation for the ratings generated for item i, and the maximum and minimum of all values of $stdev(I_i)$ are denoted by max and min, respectively.

The reliability values of users calculated using Eq. (14) are employed to define the third factor. To do this, the summation of reliability values related to the users who have provided a rating for the target item i is used to calculate this factor. The main motivation behind the third factor is the fact that assigning the ratings to the target item by reliable users results in increasing its reliability value. Thus, the following equation is utilized to define the third factor of the target item i's reliability:

$$f_c(C_i) = 1 - \frac{\overline{c}}{\overline{c} + C_i} \tag{17}$$

where

$$C_i = \sum_{u \in U_i} UR_u \tag{18}$$

and $\overline{c}$ refers to the median value of all calculated values of C_i, U_i implies a subset of users who have rated the target item i, and UR_u is the user u's reliability computed by Eq. (14).

Finally, the reliability of items is achieved according to the integration of the three factors. To this end, we need to set the weights of these factors in the integration mechanism. The number of ratings assigned to the target item does not depend on the other factors, thus the weight of the first factor is set to 1. The second and third factors depend directly on the value of the first factor. Therefore, the weights of the second and third factors are set to the value of the first factor. Accordingly, the geometric average of the three factors is considered as the reliability of the target item i:

$$IR_i = \left[f_i(I_i).f_{sd}(stdev(I_i))^{f_i(I_i)}.f_c(C_i)^{f_i(I_i)} \right]^{\frac{1}{1+2f_i(I_i)}} \tag{19}$$

where IR_i implies the reliability of the target item i. This reliability measurement is used in the proposed rating profile expansion technique to determine most reliable items in order to add to the unreliable rating profiles. This leads to an enhancement in the reliability of the proposed rating profile expansion technique by providing more efficient user-item rating matrix for the recommendation process.

3.4 Rating profile expansion

In this section, an efficient rating profile expansion technique is developed in order to improve the quality of user-item rating matrix. In particular, each unreliable rating profile which contains insufficient ratings is expanded by incorporating additional virtual ratings. In order to guarantee the reliability of these virtual ratings, for each user u who has an unreliable rating profile I_u, the items in the set $\overline{I}_u = I - I_u$ are sorted regarding their reliability values calculated using Eq. (19) and a subset of most reliable items is considered to incorporate into the user u's profile. It is worth noting that the minimum number of required ratings (i.e., n'_u) obtained using Eq. (3) is used to determine that how many virtual ratings should be added to the rating profile. Accordingly, the number of virtual ratings which should be considered to expand the rating profile of user u is equal to $n'_u - |I_u|$. To calculate the virtual ratings, first of all, the demographic information of users and genres of movies are employed to obtain the similarity values between users. The similarity value between users u and v based on the demographic information is calculated by performing a weighted average on the demographic attributes such as age, gender, and occupation using the following equation:

$$sim(u,v)_{demo} = \frac{\sum_{j=1}^{|D|} x_j d_j}{\sum_{j=1}^{|D|} d_j} \tag{20}$$

where $sim(u,\ v)_{demo}$ denotes the similarity value between users u and v based on demographic information, D is the set of demographic attributes, d_j is the weight of jth demographic attribute, and $x_j = 1$ if the jth demographic attribute for both users u and v is equal; otherwise $x_j = 0$.

To calculate similarity values based on genres of movies, we consider the genres of movies that have been rated by the users. Therefore, the similarity value between users u and v based on the genres of movies is computed using Eq. (21):

$$sim(u,v)_{genre} = \frac{\sum_{j=1}^{g} G_{u,j} G_{v,j}}{\sqrt{\sum_{j=1}^{g} G_{u,j}^2}\sqrt{\sum_{j=1}^{g} G_{v,j}^2}} \tag{21}$$

where $sim(u,\ v)_{genre}$ indicates the similarity value between users u and v based on genres of movies, $G_{|U| \times g}$ is the user-genre matrix, g is the number of genres for the movies, $G_{u,\ j}$ indicates that whether user u has rated the movies in the genre j. It should be noted that $G_{u,j} =$ 1 if user u has rated at least one movie in the genre j, otherwise; $G_{u,\ j} = 0$.

Eq. (22) computes the Harmonic mean of the demographic-based and genre-based similarity values as follows:

$$sim(u,v)_{demo\_genre} = \begin{cases} \dfrac{2 \times sim(u,v)_{demo} \times sim(u,v)_{genre}}{sim(u,v)_{demo} + sim(u,v)_{genre}} & if\ sim(u,v)_{demo} \neq 0\ and\ sim(u,v)_{genre} \neq 0 \\ sim(u,v)_{demo} & if\ sim(u,v)_{demo} \neq 0\ and\ sim(u,v)_{genre} = 0 \\ sim(u,v)_{genre} & if\ sim(u,v)_{demo} = 0\ and\ sim(u,v)_{genre} \neq 0 \\ 0 & if\ sim(u,v)_{demo} = 0\ and\ sim(u,v)_{genre} = 0 \end{cases} \tag{22}$$

where $sim(u,\ v)_{demo\_genre}$ indicates the combined similarity value between users u and v based on both demographic and genres information.

Finally, the virtual rating $VR_{u,\ i}$ for user u and each item $i \in \overline{I}_u$ is computed according to the combined similarity values in the following:

$$VR_{u,i} = \frac{\sum_{v \in NN_u} sim(u,v)_{demo\_genre} \times r_{v,i}}{\sum_{v \in NN_u} sim(u,v)_{demo\_genre}} \tag{23}$$

where $r_{v,\ i}$ refers to the rating corresponding to user u and item i, and NN_u represents the nearest neighbors set of user u which is obtained by selecting N users with the highest similarity values.

After calculating the virtual ratings, the expansion of rating profiles can be performed by incorporating these virtual ratings. For each user u, if $|I_u| < n_u^{'}$ then the virtual ratings in the set $\overline{I}_u$ will be calculated using Eq. (23). Then, the items' reliability calculated using Eq. (19) is utilized to sort the items in the set $\overline{I}_u$. Finally, the user u's profile is expanded through adding a number of $n_u^{'} - |I_u|$ virtual ratings with the highest reliability. This process is repeated for all users with unreliable profiles which results in making a new user-item rating matrix using the proposed rating profile expansion method.

Algorithm 1 Genre and Demographic information based Profile Expansion technique (GDPE)

Inputs: Parameters $\boldsymbol{\delta}$, $\boldsymbol{N}$, $\boldsymbol{top\_N}$.
Output: The list of recommendations.
Begin algorithm:
Generate the training set ($\boldsymbol{Tr}$) and the test set ($\boldsymbol{Te}$);
for all $\boldsymbol{u} \in \boldsymbol{U}$ **do**
    Compute $\boldsymbol{n}_{\boldsymbol{u}}'$ using Eq. (3) based on $\boldsymbol{Tr}$;
end for
for all $\boldsymbol{u} \in \boldsymbol{U}$ **do**
    Utilize Eqs. (9), (11), and (12) to compute the factors of user $\boldsymbol{u}$'s reliability;
    Calculate the reliability of user $\boldsymbol{u}$ (i.e., $\boldsymbol{UR}_{\boldsymbol{u}}$) using Eq. (14);
end for
for all $\boldsymbol{i} \in \boldsymbol{I}$ **do**
    Utilize Eqs. (15)-(17) to compute the factors of item $\boldsymbol{i}$'s reliability;
    Calculate the reliability of item $\boldsymbol{i}$ (i.e., $\boldsymbol{IR}_{\boldsymbol{i}}$) using Eq. (19);
end for
for all $\boldsymbol{u} \in \boldsymbol{U}$ **do**
    Let $\boldsymbol{I}_{\boldsymbol{u}}$ be a subset of items rated by user $\boldsymbol{u}$;
    if ($|\boldsymbol{I}_{\boldsymbol{u}}| < \boldsymbol{n}_{\boldsymbol{u}}'$) **then**
        Set $\overline{\boldsymbol{I}}_{\boldsymbol{u}} = \boldsymbol{I} - \boldsymbol{I}_{\boldsymbol{u}}$;
        Sort $\overline{\boldsymbol{I}}_{\boldsymbol{u}}$ descending based on their reliability values;
        Select $\boldsymbol{n}_{\boldsymbol{u}}' - |\boldsymbol{I}_{\boldsymbol{u}}|$ items from $\overline{\boldsymbol{I}}_{\boldsymbol{u}}$ as $\tilde{\boldsymbol{I}}_{\boldsymbol{u}}$;
        for all $\boldsymbol{i} \in \tilde{\boldsymbol{I}}_{\boldsymbol{u}}$ **do**
            Compute virtual rating $\boldsymbol{VR}_{\boldsymbol{u},\boldsymbol{i}}$ using Eq. (23);
            Add $\boldsymbol{VR}_{\boldsymbol{u},\boldsymbol{i}}$ to user $\boldsymbol{u}$**'s** rating profile;
        end for
    end if
end for
for all $\boldsymbol{u} \in \boldsymbol{U}$ **do**
    Utilize the expanded rating profiles to compute the similarity values between user $\boldsymbol{u}$ and other users using Eq. (10);
end for
for all $\boldsymbol{r}_{\boldsymbol{u},\boldsymbol{i}} \in \boldsymbol{Te}$ **do**
    Predict $\boldsymbol{P}_{\boldsymbol{u},\boldsymbol{i}}$ using Eq. (24);
end for
Generate $\boldsymbol{top\_N}$ recommendations list;
End algorithm.

3.5 Recommendation

To generate a recommendations list for the target user, firstly, the final similarity values between users are computed by employing the expanded rating profiles, and then the unknown ratings are predicted. For this purpose, Eq. (10) is utilized to compute these similarity values as the Pearson correlation coefficient. In addition, the unknown rating $P_{u,i}$ for user u and item i is predicted as:

$$P_{u,i} = \overline{r}_u + \frac{\sum_{v \in NN_u} sim(u,v) \times \left(r_{v,i} - \overline{r}_v\right)}{\sum_{v \in NN_u} sim(u,v)} \tag{24}$$

where $sim(u,\ v)$ refers to the final similarity value between users u and v (Eq. (10)) based on the expanded rating profiles, and NN_u indicates a set containing N users that are considered as the user u's nearest neighbors. After calculating the unknown ratings using Eq. (24), the $top\ \_\ N$ recommendations list is generated for the target user. Algorithm 1 indicates the pseudo-code of GDPE model.

4 Experiments

It should be investigated that how much the proposed method is effective by assessing its performance in comparison to other recommendation approaches through conducting extensive experiments. Accordingly, probabilistic matrix factorization (PMF) [36], nonlinear probabilistic matrix factorization (NLPMF) [23], multi-level collaborative filtering (MLCF) [31], item-global profile expansion (IGPE) [14], popularity-based probabilistic latent semantic analysis (PPLSA) [17], matrix factorization model based on linked open data (MF-LOD) [29], user profile correlation-based similarity (UPCSim) for movie recommender systems [44], positive profile and negative profile (PP/NP) for movie recommender systems [13], genre-similarity hybrid filtering (GSHF) [35], and graph-based hybrid recommender system for movie recommendation (GHRS) [48] are utilized as competitors in the experiments.

4.1 Datasets

MovieLens 100 K and MovieLens 1 M are two popular movie recommendation datasets collected by the GroupLens Research Project[1] that are used in this paper to carry out the experiments. There are 100,000 ratings in MovieLens 100 K which are assigned to 1682 movies by 943 users while MovieLens 1 M consists of 1,000,209 ratings provided by 6040 users to 3900 movies. The ratings in these datasets are in the range of $[1,\ 5]$ with step 1, where 1 and 5 indicate the lowest and highest interests, respectively. These datasets contain the demographic information of users including age, gender, occupation, and zip code, and also 18 different genres of movies.

4.2 Evaluation metrics

For illustrating the recommendation methods' efficiency, four evaluation metrics: mean absolute error (MAE), precision, recall, and F1 are utilized in the experiments. MAE indicates a metric in which the amount of error between the predicted ratings (p_i) and real ratings (r_i) is formulized as follows:

$$MAE = \frac{1}{|Te|} \sum_{i=1}^{|Te|} |r_i - p_i| \tag{25}$$

where $|Te|$ indicates the number of ratings that are predicted by recommenders.

[1] https://grouplens.org/datasets/movielens/

Precision and recall are two metrics which indicate the quality of recommendations list produced by recommendation methods. In particular, the ratio of relevant items produced by recommendation methods in the recommendation list and the ratio of relevant items in the test set recommended to users are referred to precision and recall, respectively. These metrics can be computed as follows:

$$precision = \frac{|\{recommended\ items\ that\ are\ relevant\}|}{top\_N} \tag{26}$$

$$recall = \frac{|\{recommended\ items\ that\ are\ relevant\}|}{|\{all\ relevant\ items\}|} \tag{27}$$

F1 metric is defined as the Harmonic mean of precision and recall values which can be calculated as follows:

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \tag{28}$$

4.3 Experimental setup

The proposed GDPE method has some input parameters that their values should be determined before carrying out the experiments. Parameters δ and N in Eqs. (3) and (24) are set to $\delta = 0.4$ and $N = 90$ for both Movielens 100 K and Movielens 1 M datasets. These values are obtained through performing several experiments that are discussed in section 4.5. The parameter d_j is the weight of jth demographic attribute which is used in Eq. (20). We use three demographic information for users including age, gender, and occupation that their weights are set to 0.3, 0.3, and 0.4, respectively. It is worth noting that the weights of demographic attributes are selected by performing a greedy search strategy to find their optimal values. In the experiments, we use different sparsity levels to show the effectiveness of the compared recommendation methods in alleviating data sparsity problem. Accordingly, we define a sparsity level θ referring to the amount of data considered in the training process of the recommendation methods. The sparsity levels are set to $\theta = 10\ \%\ , 20\ \%\ , 50\ \%\ , 80\ \%\ , 90\%$ in which $\theta = 20\%$ means that the training set is formed by selecting 80% of whole data and the test set comprises the remaining data. According to this assumption, the higher value of sparsity level shows that a lower amount of data is utilized in the training set.

4.4 Comparison with other models

The results of experiments performed on Movielens 100 K and Movielens 1 M datasets are discussed in this section to investigate the superiority of our model compared to other recommenders. The experimental results based on MAE metric are reported in Table 1 in which different sparsity levels are considered to show the ability of compared recommendation models in addressing data sparsity problem. It can be seen that the MAE values of GDPE model are lower than other models for all the sparsity levels indicating that the proposed method achieves the best results compared to other competitors. In the case $\theta = 20\%$, GDPE achieves the MAE values of 0.802 and 0.827 for Movielens 100 K and Movielens 1 M

Table 1 The MAE values for compared algorithms based on MovieLens 100 K and MovieLens 1 M datasets, and different levels of sparsity (θ)

Algorithm	MovieLens 100 K					MovieLens 1 M				
	θ=10%	θ=20%	θ=50%	θ=80%	θ=90%	θ=10%	θ=20%	θ=50%	θ=80%	θ=90%
PMF	0.885	0.904	1.151	1.196	1.207	0.915	0.942	1.185	1.241	1.265
NLPMF	0.895	0.921	1.183	1.209	1.221	0.921	0.967	1.227	1.259	1.273
MLCF	0.903	0.936	1.217	1.234	1.249	0.953	0.991	1.246	1.287	1.296
IGPE	0.931	0.964	1.148	1.182	1.201	0.972	0.997	1.173	1.228	1.241
PPLSA	0.874	0.883	1.127	1.164	1.179	0.891	0.928	1.169	1.215	1.229
UPCSim	0.856	0.874	1.109	1.139	1.154	0.875	0.913	1.157	1.194	1.221
PP/NP	0.847	0.861	1.102	1.127	1.151	0.859	0.895	1.138	1.176	1.208
GSHF	0.834	0.852	0.994	1.118	1.139	0.851	0.874	1.095	1.149	1.182
MF-LOD	0.833	0.849	0.971	1.113	1.128	0.847	0.861	1.075	1.132	1.167
GHRS	0.814	0.821	0.945	1.102	1.114	0.829	0.853	1.037	1.123	1.154
GDPE	**0.781**	**0.802**	**0.914**	**0.998**	**1.056**	**0.806**	**0.827**	**0.984**	**1.062**	**1.108**

The best results are shown in bold

datasets, respectively. In this case, the GHRS method is the second-best performer by obtaining the MAE values of 0.821 and 0.853 for Movielens 100 K and Movielens 1 M datasets, respectively. Also, it is shown that the performance of the compared models is declined when the sparsity level increases from $\theta = 10\%$ to $\theta = 90\%$. This issue is expectable as increasing the sparsity level leads to consider lower data in the training process of recommendation methods. As a result, we can conclude that GDPE has a higher ability than other models in alleviating data sparsity problem.

The results of experiments for both Movielens 100 K and Movielens 1 M datasets are reported in Tables 2, 3, and 4 for precision, recall, and F1, respectively. In these experiments, we consider different sparsity levels from $\theta = 10\%$ to $\theta = 90\%$ with step 10%. Moreover, the number of items recommended to target user is set to $top\_N = 10$ for all compared recommendation methods. These experimental results illustrate that GDPE achieves higher values of precision, recall, and F1 metrics in comparison to other competitors. It is worth noting that these metrics indicate how much the obtained recommendations list is appropriate for the target user. Therefore, we can conclude that the proposed method generates more

Table 2 The precision values for compared algorithms based on MovieLens 100 K and MovieLens 1 M datasets, and different levels of sparsity (θ). The length of recommendations list is set to $top\_N = 10$

Algorithm	MovieLens 100 K					MovieLens 1 M				
	θ=10%	θ=20%	θ=50%	θ=80%	θ=90%	θ=10%	θ=20%	θ=50%	θ=80%	θ=90%
PMF	0.629	0.604	0.486	0.448	0.441	0.584	0.562	0.457	0.397	0.391
NLPMF	0.618	0.597	0.483	0.439	0.432	0.576	0.551	0.451	0.391	0.386
MLCF	0.615	0.592	0.474	0.421	0.415	0.552	0.536	0.438	0.378	0.372
IGPE	0.586	0.564	0.491	0.454	0.446	0.548	0.523	0.462	0.408	0.399
PPLSA	0.637	0.611	0.496	0.461	0.458	0.591	0.581	0.469	0.421	0.413
UPCSim	0.641	0.618	0.508	0.475	0.471	0.604	0.588	0.486	0.438	0.429
PP/NP	0.646	0.632	0.517	0.486	0.473	0.609	0.597	0.501	0.452	0.445
GSHF	0.659	0.637	0.531	0.498	0.481	0.617	0.611	0.518	0.469	0.458
MF-LOD	0.665	0.641	0.544	0.503	0.488	0.632	0.619	0.532	0.481	0.473
GHRS	0.679	0.653	0.562	0.531	0.506	0.654	0.632	0.549	0.502	0.495
GDPE	**0.713**	**0.689**	**0.603**	**0.562**	**0.547**	**0.691**	**0.673**	**0.586**	**0.534**	**0.521**

The best results are shown in bold

Table 3 The recall values for compared algorithms based on MovieLens 100 K and MovieLens 1 M datasets, and different levels of sparsity (θ). The length of recommendations list is set to $top\ \_\ N = 10$

Algorithm	MovieLens 100 K					MovieLens 1 M				
	θ=10%	θ=20%	θ=50%	θ=80%	θ=90%	θ=10%	θ=20%	θ=50%	θ=80%	θ=90%
PMF	0.668	0.631	0.537	0.482	0.467	0.618	0.598	0.504	0.443	0.435
NLPMF	0.663	0.623	0.521	0.474	0.453	0.612	0.587	0.489	0.428	0.421
MLCF	0.654	0.615	0.519	0.466	0.448	0.607	0.581	0.465	0.414	0.402
IGPE	0.632	0.598	0.554	0.493	0.479	0.593	0.572	0.521	0.458	0.447
PPLSA	0.679	0.638	0.571	0.498	0.485	0.621	0.609	0.537	0.465	0.452
UPCSim	0.686	0.645	0.588	0.505	0.491	0.628	0.612	0.542	0.474	0.458
PP/NP	0.701	0.657	0.603	0.521	0.508	0.652	0.639	0.567	0.488	0.469
GSHF	0.708	0.671	0.615	0.536	0.524	0.661	0.648	0.585	0.506	0.491
MF-LOD	0.717	0.683	0.627	0.557	0.546	0.679	0.663	0.606	0.524	0.513
GHRS	0.725	0.701	0.636	0.572	0.559	0.693	0.678	0.615	0.541	0.528
GDPE	**0.756**	**0.734**	**0.662**	**0.605**	**0.593**	**0.738**	**0.712**	**0.638**	**0.572**	**0.564**

The best results are shown in bold

accurate and appropriate recommendations list than the other recommenders for the target user. The proposed method achieves the precision, recall, and F1 values for Movielens 100 K dataset in the case $\theta = 20\%$ in which their values are 0.689, 0.734, and 0.710, respectively. In this case, the second-best model is GHRS in which the values for precision, recall, and F1 metrics are 0.653, 0.701, and 0.676, respectively. The values of all evaluation metrics are decreased when the sparsity level increases from $\theta = 10\%$ to $\theta = 90\%$.

The results of Tables 1-4 prove that the proposed method is more effective than other methods in alleviating the data sparsity problem. In addition, we evaluate different recommendation models based on the sparsity level $\theta = 90\%$, which means that we only consider 10% of the whole provided ratings for each user. Considering such a small portion of ratings for each user in the recommendation process can appropriately model the behavior of cold start users as a common challenging problem in the recommender systems. This is due to the fact that the cold start users are those who have explicitly expressed only a small number of ratings about the available items. Hence, we can conclude that the proposed method not only alleviates the data sparsity problem but can also address the cold start challenge in the

Table 4 The F1 values for compared algorithms based on MovieLens 100 K and MovieLens 1 M datasets, and different levels of sparsity (θ). The length of recommendations list is set to $top\ \_\ N = 10$

Algorithm	MovieLens 100 K					MovieLens 1 M				
	θ=10%	θ=20%	θ=50%	θ=80%	θ=90%	θ=10%	θ=20%	θ=50%	θ=80%	θ=90%
PMF	0.647	0.617	0.510	0.464	0.453	0.601	0.579	0.479	0.418	0.411
NLPMF	0.639	0.609	0.501	0.455	0.442	0.593	0.568	0.469	0.408	0.402
MLCF	0.633	0.603	0.495	0.442	0.430	0.578	0.557	0.451	0.395	0.386
IGPE	0.608	0.580	0.520	0.472	0.461	0.569	0.546	0.489	0.431	0.421
PPLSA	0.657	0.624	0.530	0.478	0.471	0.605	0.594	0.501	0.441	0.431
UPCSim	0.662	0.631	0.545	0.489	0.480	0.615	0.599	0.512	0.455	0.443
PP/NP	0.672	0.644	0.556	0.502	0.489	0.629	0.617	0.531	0.469	0.456
GSHF	0.682	0.653	0.569	0.516	0.501	0.638	0.628	0.549	0.486	0.473
MF-LOD	0.690	0.661	0.582	0.528	0.515	0.654	0.640	0.566	0.501	0.492
GHRS	0.701	0.676	0.596	0.551	0.531	0.673	0.654	0.580	0.521	0.511
GDPE	**0.733**	**0.710**	**0.631**	**0.582**	**0.569**	**0.713**	**0.691**	**0.610**	**0.552**	**0.541**

The best results are shown in bold

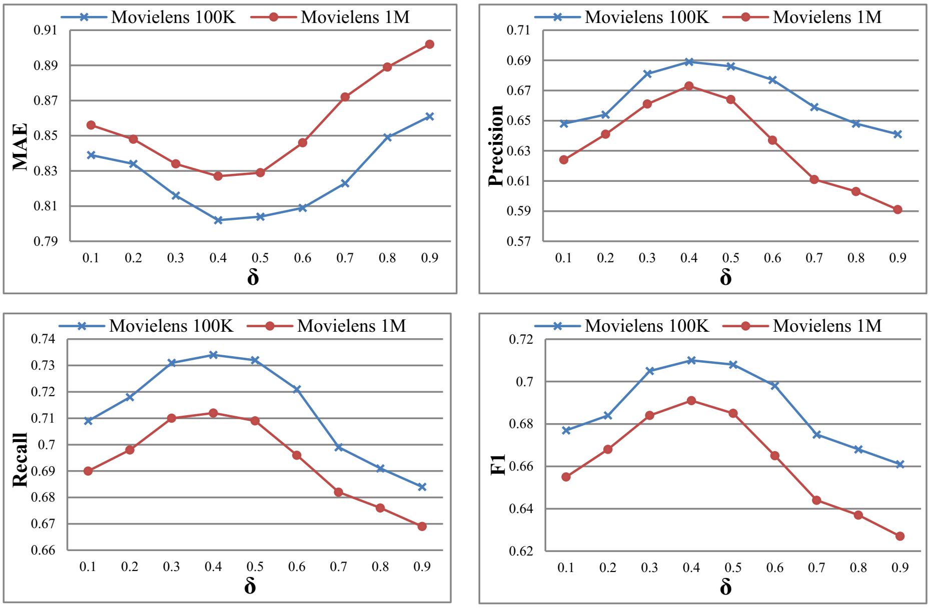
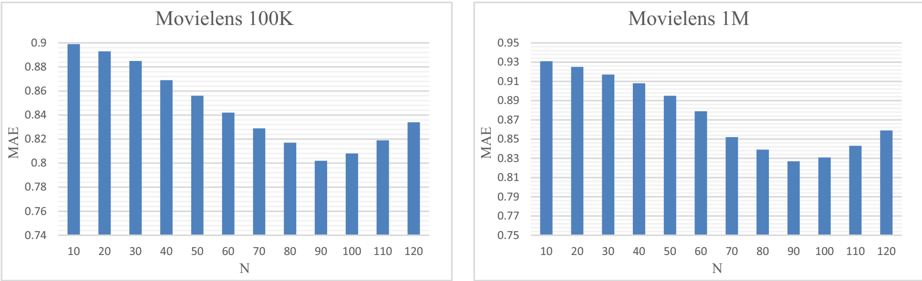


Fig. 3 The values of MAE, precision, recall, and F1 metrics for the proposed method based on different values of parameter δ. The sparsity level is set to $\theta = 20\%$

recommender systems. The main reasons behind this achievement include adding virtual ratings to the user's rating profile and also utilizing the genres of movies and demographic information of users in the proposed method.

4.5 Parameter analysis

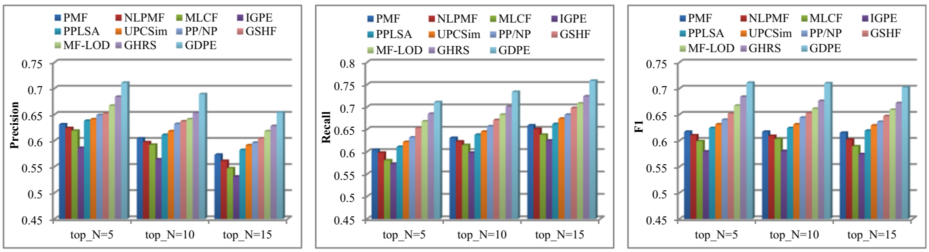
The effect of different values of the input parameters on the efficiency of the proposed GDPE method is investigated in this section according to different evaluations metrics. To do this, a number of experiments have been conducted on the Movielens 100 K and Movielens 1 M datasets. Fig. 3 shows the values of MAE, precision, recall, and F1 metrics for GDPE in terms of different values of parameter δ for both Movielens 100 K and Movielens 1 M datasets. δ is an input parameter utilized in Eq. (3) as a predefined threshold value to determine the level of reliability in producing accurate predictions. It is worth noting that we use $\theta = 20\%$ for the sparsity level in these experiments. As can be seen from these results, the MAE values for GDPE decrease by increasing δ from $\delta = 0.1$ to $\delta = 0.4$. Whilst, the MAE values increase when δ is higher than 0.4. Moreover, increasing δ from $\delta = 0.1$ to $\delta = 0.4$ results in boosting GDPE's performance according to precision, recall, and F1 metrics while these metrics will be declined for δ values higher than 0.4. Accordingly, we can conclude that $\delta = 0.4$ is the best value which makes the best performance of the proposed method in terms of all evaluation metrics and both Movielens 100 K and Movielens 1 M datasets. The higher values of parameter δ result in increasing the number of virtual ratings added to the unreliable rating profiles. Hence, when δ exceeds 0.4, a large number of virtual ratings will be incorporated into the unreliable profiles leading to an increment in the probability of adding irrelevant ratings and declining the recommendation accuracy of the proposed method.



(a) (b)

Fig. 4 The MAE values for the proposed method based on different number of nearest neighbors (N): (**a**) Movielens 100 K dataset, (**b**) Movielens 1 M dataset. The sparsity level is set to $\theta = 20\%$

Fig. 4 demonstrates the effect of different number of nearest neighbors (N) on the GDPE's performance measured by the MAE metric for both Movielens 100 K and Movielens 1 M datasets. N is utilized in Eq. (24) as the number of nearest neighbors employed in the rating prediction process. In these results, the value of N changes from $N = 10$ to $N = 120$ with step 10. These results reveal that the MAE values are declined by increasing N from 10 to 90 for both datasets. Therefore, increasing the value of parameter N from $N = 10$ to $N = 90$ results in enhancing the GDPE's performance. Moreover, the performance is decreased when N exceeds 90. This is due to the fact that when N exceeds 90, the probability of considering ineffective neighbors in the recommendation process will be increased which this issue reduces the accuracy of predictions. Another important parameter used in the recommendation process is top_N referring to the number of items considered in the recommendations list. Figs. 5 and 6 show the precision, recall, and F1 values of different recommendation methods for $top\_N = 5,10,15$ based on Movielens 100 K and Movielens 1 M datasets, respectively. As it is shown, the values of all metrics for GDPE are higher than other compared models in terms of all lengths of recommendations list and both datasets. Therefore, GDPE significantly outperforms other recommenders according to precision, recall, and F1 metrics and also different top_N values. Figs. 5 and 6 demonstrate that the precision values of all compared models are declined when top_N increases from 5 to 15. Moreover, increasing the value of top_N results in increasing the value of recall metric for all recommendation methods.



(a) (b) (c)

Fig. 5 Performance comparison on Movielens 100 K dataset in terms of different top_N values: (**a**) Precision metric, (**b**) Recall metric, and (**c**) F1 metric. The sparsity level is set to $\theta = 20\%$

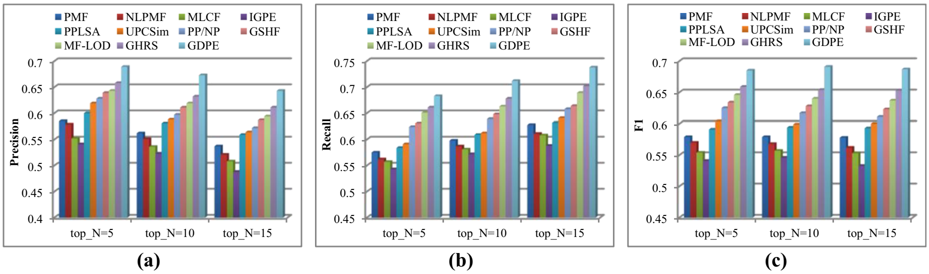


Fig. 6 Performance comparison on Movielens 1 M dataset in terms of different top_N values: (**a**) Precision metric, (**b**) Recall metric, and (**c**) F1 metric. The sparsity level is set to $\theta = 20\%$

5 Conclusions

Movie recommendation methods have many applications in real-world systems that the main purpose of these methods is to help users find their interested movies. These methods mainly use the preferences of users in the past which are represented as the ratings assigned to different movies by users. However, users tend to provide ratings only for a small portion of movies leading to raise a problem called data sparsity. We proposed a movie recommendation method in this paper in which an effective profile expansion technique is utilized for the aim of empowering user's ratings profile through additional virtual ratings. A probabilistic model is utilized in the proposed method to determine how many virtual ratings should be added to each rating profile. The genres of movies and demographic information of users are used to compute these virtual ratings. To demonstrate that how much the proposed method is effective, several experiments have been conducted by utilizing two movie recommendation datasets and their results showed that our model is more efficient than other competitors. For the future work, other side data such as social relationships and tag information can be employed to enhance the ability of the proposed method in generating more accurate recommendations. Also, the proposed rating profile expansion model can be utilized in other types of recommender systems such as music recommendation methods.

Data availability All the datasets used in this paper are publicly available. The links to access these datasets are provided in this paper.

Declarations

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Ahmadian S, Moradi P, Akhlaghian F (2014) An improved model of trust-aware recommender systems using reliability measurements. In: 2014 6th Conference on Information and Knowledge Technology (IKT). pp. 98–103
2. Ahmadian S, Meghdadi M, Afsharchi M (2018) Incorporating reliable virtual ratings into social recommendation systems. Appl Intell 48:4448–4469
3. 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). pp. 1139–1144

4. Ahmadian S, Afsharchi M, Meghdadi M (2019) An effective social recommendation method based on user reputation model and rating profile enhancement. J Inf Sci 45:607–642
5. Ahmadian S, Afsharchi M, Meghdadi M (2019) A novel approach based on multi-view reliability measures to alleviate data sparsity in recommender systems. Multimed Tools Appl 78:17763–17798
6. Ahmadian S, Joorabloo N, Jalili M, Ahmadian M (2021) Alleviating data sparsity problem in time-aware recommender systems using a reliable rating profile enrichment approach. Expert Systems with Applications. p. 115849
7. Alonso S, Bobadilla J, Ortega F, Moya R (2019) Robust model-based reliability approach to tackle shilling attacks in collaborative filtering recommender systems. IEEE Access 7:41782–41798
8. Bishop CM (2006) Pattern recognition and machine learning: springer-Verlag New York
9. Breitfuss A, Errou K, Kurteva A, Fensel A (2021) Representing emotions with knowledge graphs for movie recommendations. Futur Gener Comput Syst 125:715–725
10. Cauteruccio F, Terracina G, Ursino D (2020) Generalizing identity-based string comparison metrics: framework and techniques. Knowl-Based Syst 187:104820
11. Chen B, Ding Y, Xin X, Li Y, Wang Y, Wang D (2021) AIRec: attentive intersection model for tag-aware recommendation. Neurocomputing 421:105–114
12. Chen X, Liu D, Xiong Z, Zha ZJ (2021) Learning and fusing multiple user interest representations for Micro-video and movie recommendations. IEEE Trans Multimed 23:484–496
13. Chen YL, Yeh YH, Ma MR (2021) A movie recommendation method based on users' positive and negative profiles. Inf Process Manag 58:102531
14. 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
15. Gan M, Cui H (2021) Exploring user movie interest space: a deep learning based dynamic recommendation model. Expert Syst Appl 173:114695
16. Huang X, Fang Q, Qian S, Sang J, Li Y, Xu C (2019) Explainable Interaction-driven User Modeling over Knowledge Graph for Sequential Recommendation. In: Proceedings of the 27th ACM International Conference on Multimedia. pp. 548–556
17. Huang L, Tan W, Sun Y (2019) Collaborative recommendation algorithm based on probabilistic matrix factorization in probabilistic latent semantic analysis. Multimed Tools Appl 78:8711–8722
18. Huang L, Ma H, He X, Chang L (2021) Multi-affect(ed): improving recommendation with similarity-enhanced user reliability and influence propagation. Front Comput Sci 15:155331
19. Indira K, Kavithadevi MK (2019) Efficient machine learning model for movie recommender systems using multi-cloud environment. Mobile Netw Appli 24:1872–1882
20. Jiang Y, Ma H, Liu Y, Li Z (2020) Exploring user trust and reliability for recommendation: A hypergraph ranking approach. In: International Conference on Neural Information Processing. pp. 333–344
21. Katarya R, Verma OP (2017) An effective collaborative movie recommender system with cuckoo search. Egyptian Inform J 18:105–112
22. Khan ZY, Niu Z, Nyamawe AS, Haq I (2021) A deep hybrid model for recommendation by jointly leveraging ratings, reviews and metadata information. Eng Appl Artif Intell 97:104066
23. Lawrence ND, Urtasun R Non-linear matrix factorization with Gaussian processes. In: ICML '09 proceedings of the 26th annual international conference on machine learning, Montreal, Quebec, Canada, 2009, pp. 601–608.
24. Li H, Li D (2021) A time-aware hybrid recommendation scheme combining content-based and collaborative filtering. Front Comput Sci 15:154613
25. Matoussek J, Vondrak J (2001) The probabilistic method: Charles University
26. Moradi P, Ahmadian S (2015) A reliability-based recommendation method to improve trust-aware recommender systems. Expert Syst Appl 42:7386–7398
27. Moradi P, Ahmadian S, Akhlaghian F (2015) An effective trust-based recommendation method using a novel graph clustering algorithm. Physica A: Statistical Mech Appl 436:462–481
28. Moradi P, Rezaimehr F, Ahmadian S, Jalili M (2016) A trust-aware recommender algorithm based on users overlapping community structure. In: 2016 sixteenth international conference on advances in ICT for emerging regions (ICTer). pp. 162–167
29. Natarajan S, Vairavasundaram S, Natarajan S, Gandomi AH (2020) Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data. Expert Syst Appl 149:113248
30. Ortega F, Cabrera RL, Prieto ÁG, Bobadilla J (2021) Providing reliability in recommender systems through Bernoulli matrix factorization. Inf Sci 553:110–128
31. Polatidis N, Georgiadis CK (2016) A multi-level collaborative filtering method that improves recommendations. Expert Syst Appl 48:100–110

32. Qu T, Wan W, Wang S (2021) Visual content-enhanced sequential recommendation with feature-level attention. Neurocomputing 443:262–271
33. Rahmanic HA, Aliannejadi M, Ahmadian S, Baratchi M, Afsharchi M, Crestani F (2019) LGLMF: local geographical based logistic matrix factorization model for POI recommendation. In: AIRS 2019: Information Retrieval Technology. pp. 66–78
34. Rezaeimehr F, Moradi P, Ahmadian S, Qader NN, Jalili M (2018) TCARS: time-and community-aware recommendation system. Futur Gener Comput Syst 78:419–429
35. Roy A, Ludwig SA (2021) Genre based hybrid filtering for movie recommendation engine. J Intell Inf Syst 56:485–507
36. Salakhutdinov R, Mnih A (2007) Probabilistic matrix factorization. In: NIPS'07: Proceedings of the 20th International Conference on Neural Information Processing Systems. pp. 1257–1264
37. Sang L, Xu M, Qian S, Martin M, Li P, Wu X (2021) Context-dependent propagating-based video recommendation in multimodal heterogeneous information networks. IEEE Trans Multimed 23:2019–2032
38. Tahmasebi F, Meghdadi M, Ahmadian S, Valiallahi K (2021) A hybrid recommendation system based on profile expansion technique to alleviate cold start problem. Multimed Tools Appl 80:2339–2354
39. Tahmasebi H, Ravanmehr R, Mohamadrezaei R (2021) Social movie recommender system based on deep autoencoder network using twitter data. Neural Comput & Applic 33:1607–1623
40. Thakker U, Patel R, Shah M (2021) A comprehensive analysis on movie recommendation system employing collaborative filtering. Multimed Tools Appl 80:28647–28672
41. Walek B, Fojtik V (2020) A hybrid recommender system for recommending relevant movies using an expert system. Expert Syst Appl 158:113452
42. Wang D, Yih Y, Ventresca M (2020) Improving neighbor-based collaborative filtering by using a hybrid similarity measurement. Expert Syst Appl 160:113651
43. Wei G, Wu Q, Zhou M (2021) A hybrid probabilistic multiobjective evolutionary algorithm for commercial recommendation systems. IEEE Transac Comput Social Syst 8:589–598
44. Widiyaningtyas T, Hidayah I, Adji TB (2021) User profile correlation-based similarity (UPCSim) algorithm in movie recommendation system. J Big Data 8:1–21
45. Xia H, Wei X, An W, Zhang ZJ, Sun Z (2021) Design of electronic-commerce recommendation systems based on outlier mining. Electron Mark 31:295–311
46. Xie H, Lui JCS (2015) Mathematical modeling and analysis of product rating with partial information. ACM Trans Knowl Discov Data 9:1–33
47. Yuan X, Han L, Qian S, Zhu L, Zhu J, Yan H (2021) Preliminary data-based matrix factorization approach for recommendation. Inf Process Manag 58:102384
48. Zamanzadeh Darban Z, Valipour MH (2022) GHRS: graph-based hybrid recommendation system with application to movie recommendation. Expert Syst Appl 200:116850
49. Zayani CA, Ghorbel L, Amous I, Mezghanni M, Péninou A, Sèdes F (2020) Profile reliability to improve recommendation in social-learning context. Online Inf Rev 44:433–454