Applied Intelligence (2021) 51:586–615
https://doi.org/10.1007/s10489-020-01811-3

A new user similarity measure in a new prediction model for collaborative filtering

S. Manochandar[1] · M. Punniyamoorthy[1]

Published online: 22 August 2020

Abstract

The Recommender Systems (RSs) based on the performance of Collaborative filtering (CF) depends on similarities among users or items obtained by a user-item rating matrix. The conventional measures such as the Pearson correlation coefficient (*PCC*), cosine (*COS*), and Jaccard (*JACC*) provide a varied and dissimilar value when the ratings between the users lie in the positive and negative side of the rating scale. These measures are also not very effective when there is sparsity in the rating matrix of the user-item. These problems are addressed by the Proximity-Impact-Popularity (*PIP*) similarity measure. Even though the *PIP* method provides an improved solution for this problem, the range of values for each component in *PIP* is very high. To address this issue and to improve the performance of a CF-based RS, a modified proximity-impact-popularity (*MPIP*) similarity measure is introduced. The expression is designed to get *PIP* values within the range of 0 to 1. A modified prediction expression is proposed to predict the available and unavailable ratings by combining user- and item-related components. The proposed method is tested by using various benchmark datasets. The size of the user-item sparse matrix varies to compare the performance of the methods in terms of mean absolute error, root mean squared error, precision, recall, and F_1-measure. The performance of the proposed method is statistically tested through the Friedman and McNemer test. The results obtained by using the evaluation criteria indicate that the proposed method provides a better solution than the conventional methods. The statistical analysis reveals that the proposed method provides minimum *MAE* and *RMSE* values. Similarly, it also provides a maximum F_1-measure for all the sub-problems.

Keywords Collaborative filtering · Prediction · Proximity-impact-popularity · Similarity measure · Recommender systems

1 Introduction

A majority of people spend an increasing amount of time on the Internet because of the excessive quantity of information on various fields it provides [1–5]. A Recommender System (RS) refers to the collection of information on websites related to the user preferences for a set of items. The RS employs different online information sources to predict the users' preferences for these items [6, 7]; therefore, it plays a vital role in the sale of products or services [8, 9]. Many users prefer to buy the products based on the recommendations of other users; thus, the preferences of the users for different products should be analyzed. From the perspective of a company, this helps to maximize profits and promote its products or services.

The RS methods based on the Collaborative filtering (CF) are incredibly popular among the researchers and practitioners. This is endorsed by a vast number of articles published in the journals and also real-life implementation cases [10–13]. CF recommends products or services based on the similarities in the preferences of a group of customers or online users, known as neighbors [14–16]. The advantage of CF-based RS is that it is domain-independent and comparatively more accurate than content-based filtering (CBF) [17]. With an increase in online purchase and the development of electronic commerce, the automated product recommendation is observed as an essential tool for enhancing the sales of products and services through Internet-based stores [18]. Conversely, CF recommends products to a customer based on the similarities between the users or products. Customers' past or historical preferences enable them to find out the similarities [16, 19–21]. Therefore, the critical component of CF effectively identifies the similarities between the users or items [18].

The Pearson Correlation Coefficient (*PCC*), Cosine (*COS*), Jaccard (*JACC*) and Jaccard Mean Squared Difference

✉ M. Punniyamoorthy
punniya@nitt.edu; mpuniya@yahoo.co.in

[1] National Institute of Technology, Tiruchirappalli, Tamilnadu 620 015, India

(*JMSD*) are the conventional methods used to compute the similarities between the users or items. The *PCC*, *COS*, and mean squared difference is the statistical metrics adopted in CF-based RS; the main advantage of these measures is that they are easy to implement to interpret the similarity values [22]. Similarly, *JACC* is the ratio of intersection by the union, which helps to find out the similarity between the users based on the number of co-rated items rated by two users. These similarity measures provide high accuracy for CF-based RS. However, these methods suffer from cold-start problems; in other words, only a few items or users are rated. This leads to an extremely sparse user-item rating matrix for the RS [7, 16, 23–25]. A similarity matrix computed by using this sparse input matrix misleads an RS [18, 26, 27]. The cold-start is an example of a sparsity problem that occurs when a new user or item is introduced; it becomes difficult to compute the similarities among users or items because of insufficient rating information [25, 28, 29]. The sparse input matrix is a significant issue in an RS, which decreases the performance of CF-based RS [30]. Another problem is that the rating values belong to both the positive and negative side of the rating scale; then, the conventional methods provide different similarity values. This is another drawback of conventional similarity measures. This misleading similarity values, eventually lower the accuracy of the CF-based RS. Therefore, a more effective similarity measure is required to improve the performance of a CF-based RS. To address these issues, Ahn [18] proposed the Proximity-Impact-Popularity (*PIP*) measure, predominantly for use in the CF to provide a better solution to the sparsity problem. In this method, two agreement conditions are included, and similarity is computed by considering positive and negative ratings. The range of values for the *PIP* is so wide that the three components (proximity, impact, popularity) are not equally treated; in different scenarios, each component has a different weight; similarly, the values of the components are not normalized [26]. If users provide an extremely positive rating for the co-rated items, then proximity has a greater weight than the impact and popularity. Similarly, if users provide an opposite rating for the co-rated items, then proximity and popularity are treated in the same manner but the impact value is very small. Each component contributes some important information in the *PIP* calculation; however, as each component is treated in unequal proportion, in reality, this affects prediction accuracy in CF-based RS. This is one of the limitations of the *PIP* measure. To overcome this, a detailed analysis of the *PIP* measure has been performed and the shortcomings of the existing similarity measure has been identified. Based on this analysis, a modified PIP (*MPIP*) similarity measure has been developed to overcome the limitations of *PIP* measure.

Generally, a similarity-based prediction expression is used to predict the rating, based on user-related average and its weighted average deviation or item-related average and its weighted average deviation. Anyone of these is considered for the prediction process. The user and item-related information are providing important information to make an accurate prediction; in the existing prediction expression, any one of the information is used for the rating prediction. This is one of the shortcomings of the existing prediction expression. Therefore, to improve the accuracy of prediction, a modified prediction expression is devised by adding the user-related deviation in the user-based prediction and item-related deviation in the item-based prediction; further, the predicted rating is the average values of user-and item-based prediction.

In this study, we have modified the *PIP* similarity measure by converting the range of each component into the range of 0 to 1. The *PIP* value is the product of proximity, impact and popularity values. These three components are weighed in different proportions in different scenarios. The deviation between the minimum and maximum values of each component is very high. If two users provide extreme positive ratings, then the resultant *PIP* values are in the magnitude of 10^3. If users provide different rating values i.e., one user provides a positive rating and another user offers a negative rating, then the *PIP* values are 10^{-1}. The variation between the two conditions is very high. Direct normalization procedures such as z-score, and min-max normalization may be adopted to get normalized *PIP* values. However, different normalization procedures provide a different range of values. For different similarity matrix, the normalization techniques provide different values. To overcome this issue, the expression itself has changed into the modified *PIP* measure to compute improved similarity values between users or items. In the existing similarity measures such as *PCC*, *COS*, *JACC*, and *JMSD*, the agreement conditions are not adopted. The main problem in existing similarity measures is flat and single value problem. In *PIP* measure provides better solution for flat and single value problems. In *MPIP* expression the same agreement conditions in *PIP* are used to differentiate the similar and dissimilar between two ratings. Similarly, it resolves the flat and single value problems. In *MPIP* values are designed to get higher magnitude values for agreement conditions and very minimal values for disagreement conditions. If two users are in disagreement with the ratings, then *MPIP* provides very lesser similarity values for this condition. In the existing *PIP* similarity measure, constant penalty value is multiplied with the proximity component. Instead of constant value, variable penalty is used in the proposed proximity expression. This helps to compute an improved similarity matrix for CF-based RS. A modified prediction expression has been introduced, which evolves by combining the user-and item-related information to enhance the effectiveness of prediction by using a spare user-item rating matrix. The modified prediction expression is also derived for predicting unavailable ratings. Based on this modified prediction expression, one can find an accurate rating prediction for the users. The modified *PIP* similarity measure and prediction expression are combined as a proposed framework for CF-based RS. This improved CF-based RS recommends more relevant products or services to the customer, which in turn enhances the level of customers' satisfaction in e-commerce services.

Experiments are conducted by using MovieLens100KB (ML100KB), and Netflix datasets, which are also used by Ahn [18]. Benchmark datasets such as Epinions, CiaoDVD, MovieTweet, FilmTrust, and MovieLens1MB (ML1MB) are used to validate the proposed framework. Each dataset is divided into different sub-problems, to test the proposed framework under sparse conditions. The performance criteria such as Mean Absolute Error (*MAE*), Root Mean Squared Error (*RMSE*), precision, recall and F_1-measure are used to measure the effectiveness of the CF-based RS. McNemer test is conducted to statistically test the performance of the different methods. Friedman rank test is performed to test the performance of the methods in all the sub-problems. The results obtained from the proposed framework are compared with the eleven existing similarity measure. The proposed method provides a minimum of *MAE* and *RMSE* values for all the dataset. Similarly, it provides higher accuracy than conventional methods. The statistical tests are conducted on *MAE*, *RMSE* and confusion matrix. The results show that the proposed framework can improve prediction performance and quality with the sparse input matrix. The symbols used throughout the paper are listed in Table 1.

The main contribution of this study is as follows: The Modified PIP (*MPIP*) similarity measure is introduced to provide an improved similarity between users or items. In existing *PIP* measure, the minimum and maximum value of each component varies at different range. The resultant *PIP* value heavily depends on any one of the component value. This issue is addressed by the proposed similarity measure (*MPIP*) by converting each component values into zero to one. In *MPIP* the variable penalty value is introduced in the proposed proximity expression to differentiate the values at various scenario. Further, Combined user and item-based prediction (*CUIP*) expression is proposed to get better predicted rating. The *CUIP* is tuned to forecast for unavailable rating. The *MPIP* and *CUIP* both are combined as a new framework to overcome sparsity problem in CF-based RS. A schema is developed to generate different level of sparse input matrix by providing equal importance to all the elements in the input user-item rating matrix. Under this schema number of users, items and ratings size vary in a different proposition. Finally, the McNemer test is explained with the graphical representation for better understating of each component used in the McNemer table.

The remainder of this paper is organized as follows: Section 2 discusses the literature related to CF-based RS and the similarity measures adopted for CF-based RS. Section 3 presents a detailed analysis of the PIP and the issues identified in the *PIP* expression. Section 4 describes the proposed method, which is a combination of the modified *PIP* similarity measure and the modified prediction expression. Section 5 presents the experimental results, and Section 6 discusses our conclusions.

2 Related literature

RS is broadly classified into three sectors; they are Content-based Filtering, Collaborative Filtering, and hybrid approach. The content-based filtering predominantly utilizes text-mining concepts; collaborative filtering, which is further subdivided into model-and memory-based filtering; and a hybrid method that combines the text and rating preferences provided by the user. [31]

CF is a type of personalized recommendation technique, which is widely used in many domains [13, 30, 32]. However, CF also suffers from several issues; for example, the cold-start problem, data sparsity [6, 33, 34], and scalability. These problems considerably reduce the user experience. Memory-based filtering is further divided into user-and item-based methods. In this study, literature related to the user-based similarity methods are collected and listed below.

Many studies have been conducted to improve prediction accuracy, resulting in the development of new similarity measures. *PCC* is often used to compute a linear relationship (i.e., correlation) between a pair of objects; it ranges from −1 to 1, where −1 indicates a negative relationship between the users, mid-value 0 indicates that there is no relationship between users, and + 1 indicates a strong relationship between users.

$$sim\left(u_j,u_h\right)^{PCC}=\frac{\sum_{i=1}^{m'}\left(r_{u_j,I_i}-\overline{r}_{u_j}\right)\left(r_{u_h,I_i}-\overline{r}_{u_h}\right)}{\sqrt{\sum_{i=1}^{m'}\left(r_{u_j,I_i}-\overline{r}_{u_j}\right)^2}\sqrt{\sum_{i=1}^{m'}\left(r_{u_h,I_i}-\overline{r}_{u_h}\right)^2}} \tag{1}$$

The *COS* similarity is a vector space model, which is highly used in information retrieval domains, where the cosine value of the angle between two vectors is used as the similarity between the users. This is calculated by using the following equation:

$$sim\left(u_j,u_h\right)^{COS}=\frac{\sum_{i=1}^{m'}\left(r_{u_j,I_i}\right)\times\left(r_{u_h,I_i}\right)}{\sqrt{\sum_{i=1}^{m'}r_{u_j,I_i}^{2}}\times\sqrt{\sum_{i=1}^{m'}r_{u_h,I_i}^{2}}} \tag{2}$$

Bobadilla et al. [6] have proposed a new similarity measure by combining Jaccard similarity and mean squared difference (*MSD*). Another new metric, Jaccard Mean Squared Difference (*JMSD*), is proposed to solve the cold user problem.

Table 1 Description of the symbols and abbreviations used in this paper

Symbols	Descriptions
u_j, u_h	User j and h, $j \neq h$, j, $h = \{1, 2, \ldots., n\}$
I_i, I_q	Items i and q, $i \neq q$, i, $q = \{1, 2, \ldots., m\}$
r_{u_j,I_i}	Rating of the user 'j' for the item 'i'
$\overline{r_u}$	The average rating for the user
$\overline{r_I}$	The average rating for the item
r_1, r_2	The ratings provided by the users
R_{ran}	Range value of the rating scale(i.e., R_{max}- R_{min})
R_{med}	The median value of the rating scale
R_{max}	The maximum value of the rating scale
R_{min}	The minimum value of the rating scale
$D(r_1, r_2)$	The absolute difference between the two rating
med^+	The median value of the positive side of the rating scale
med^-	The median value of the negative side of the rating scale
δ	penalty value
$\overline{r}_{I_i}$, $\overline{r}_{I_q}$	Average rating of the item 'i' and item 'q'
$\overline{r}_{u_j}$, $\overline{r}_{u_h}$	Average rating of the user 'j' and user 'h'
P_{u_j,I_i}	Predicted for the user 'j' for the item 'i'
$sim(u_j, u_h)$	Similarity for the user 'j' and user 'h'
$sim(I_i, I_q)$	Similarity for the item 'i' and item 'q'
$\lvert I_{u_j} \rvert$	Frequency of the items rated by the user 'j'
$\lvert I_{u_h} \rvert$	Frequency of the items rated by the user 'h'
$\lvert I_{u_j} \cap I_{u_h} \rvert$	Total Number of co-rated items between user 'j' and 'h'
$\overline{I_{u_j}}$, $\overline{I_{u_h}}$	Number of un-co-rated items of user j' and 'h'
n	Total number of Users
m	Total number of Items
n'	number of items co-rated by both users
m'	number of users rated both of the items
$\#R$	Total number of ratings
F_r	Friedman calculated value
PCC	Pearson Correlation Coefficient
COS	Cosine similarity measure
JACC	Jaccard similarity measure
JMSD	Jaccard Mean Squared
PIP	Proximity-Impact-Popularity
MPIP	Modified Proximity-Impact-Popularity
PSS	Proximity-Significance-Singularity
URP	User Rating Preference
NHSM	New Heuristic Similarity Measure
BCF_{Corr}	Bhattacharyya coefficient with a correlation-based similarity measure
BCF_{Med}	Bhattacharyya coefficient with a median-based similarity measure
RJACC	Relevant Jaccard similarity measure
RJMSD	Relevant Jaccard mean squared difference
SOQN	Sub one quasi norm-based similarity measure
UP	User-based predication
IP	Item-based prediction
CUIP	Combined user and item-based prediction

$$sim\left(u_j, u_h\right)^{JACC} = \frac{\left|I_{u_j}\right| \cap \left|I_{u_h}\right|}{\left|I_{u_j}\right| \cup \left|I_{u_h}\right|} \tag{3}$$

$$sim\left(u_j, u_h\right)^{MSD} = 1 - \left(\frac{\sum_{i=1}^{m'} \left(r_{u_j,I_i} - r_{u_h,I_i}\right)^2}{\left|I_{u_j} \cap I_{u_h}\right|}\right) \tag{4}$$

$$sim\left(u_j, u_h\right)^{JMSD} = \left(\frac{\left|I_{u_j}\right| \cap \left|I_{u_h}\right|}{\left|I_{u_j}\right| \cup \left|I_{u_h}\right|}\right) \times \left(1 - \frac{\sum_{i=1}^{m'} \left(r_{u_j,I_i} - r_{u_h,I_i}\right)^2}{\left|I_{u_j} \cap I_{u_h}\right|}\right) \tag{5}$$

The Spearman Rank Correlation (*SRC*) is another similarity measure that relies on the rank of the items instead of the rating provided by the users as in the *PCC*. The rankings are based on the higher to lower-rated items [35]. Ahn [18] has proposed the PIP similarity measure for CF for both agreement and disagreement situations.

Further, this similarity measure is modified into a non-linear model by Liu et al. [26], which is termed as a new heuristic similarity measure (*NHSM*). It comprises the proximity-significance-singularity (*PSS*) combined with the modified Jaccard (*JACC′*) function and the user rating preference (*URP*).

$$Proximity\left(r_{u_j,I_i}, r_{u_h,I_i}\right) = 1-\left(\frac{1}{1+exp\left(-\left|r_{u_j,I_i}-r_{u_h,I_i}\right|\right)}\right) \quad (6)$$

$$Significance\left(r_{u_j,I_i}, r_{u_h,I_i}\right)=\frac{1}{1+exp\left(-\left|r_{u_j,I_i}-R_{med}\right| \times \left|r_{u_h,I_i}-R_{med}\right|\right)} \quad (7)$$

$$Singularity\left(r_{u_j,I_i}, r_{u_h,I_i}\right) = 1-\left(\frac{1}{1+exp\left(-\left|\frac{r_{u_j,I_i}-r_{u_h,I_i}}{2}-\overline{r}_{I_i}\right|\right)}\right) \quad (8)$$

$$PSS\left(r_{u_j,I_i}, r_{u_h,I_i}\right) = Proximity\left(r_{u_j,I_i}, r_{u_h,I_i}\right) \times Significance\left(r_{u_j,I_i}, r_{u_h,I_i}\right) \times Singularity\left(r_{u_j,I_i}, r_{u_h,I_i}\right) \quad (9)$$

$$sim\left(u_j, u_h\right)^{PSS} = \sum_{i=1}^{m'} PSS\left(r_{u_j,I_i}, r_{u_h,I_i}\right) \quad (10)$$

$$sim\left(u_j, u_h\right)^{Jacc'} = \frac{\left|I_{u_j} \cap I_{u_h}\right|}{\left|I_{u_j}\right| \times \left|I_{u_h}\right|} \quad (11)$$

$$sim\left(u_j, u_h\right)^{URP} = 1-\frac{1}{1+\exp\left(-\left|\overline{r}_{u_j}-\overline{r}_{u_h}\right| \times \left|\sigma_{u_j}-\sigma_{u_h}\right|\right)} \quad (12)$$

$$sim\left(u_j, u_h\right)^{NHSM} = sim\left(u_j, u_h\right)^{PSS} \times sim\left(u_j, u_h\right)^{Jacc'} \times sim\left(u_j, u_h\right)^{URP} \quad (13)$$

Bhattacharyya Coefficient in CF is proposed by Patra et al. [17] by combining global and local similarity measures; this coefficient is treated as a global measure, and the local measure is calculated based on the correlation or cosine similarity measure.

$$sim\left(u_j, u_h\right) = JACC\left(u_j, u_h\right) + \sum_{i \in I_{u_j}} \sum_{q \in I_{u_h}} BC\left(I_i, I_q\right) \times loc\left(r_{u_j}, r_{u_h}\right) \quad (14)$$

$$BC\left(I_i, I_q\right) = \sum_{j=1}^{n} \sqrt{Pr\left(u_j, I_i\right) \times Pr\left(u_h, I_q\right)} \quad (15)$$

$$loc\left(u_j, u_h\right)^{corr} = \frac{\left(r_{u_j,I_i}-\overline{r}_{u_j}\right) \times \left(r_{u_h,I_i}-\overline{r}_{u_h}\right)}{\sqrt{\sum_{i=1}^{m'}\left(r_{u_j,I_i}-\overline{r}_{u_j}\right)^2}\sqrt{\sum_{i=1}^{m'}\left(r_{u_h,I_i}-\overline{r}_{u_h}\right)^2}} \quad (16)$$

$$loc\left(u_j, u_h\right)^{med} = \frac{\left(r_{u_j,I_i}-R_{med}\right) \times \left(r_{u_h,I_i}-R_{med}\right)}{\sqrt{\sum_{i=1}^{m'}\left(r_{u_j,I_i}-R_{med}\right)^2}\sqrt{\sum_{i=1}^{m'}\left(r_{u_h,I_i}-R_{med}\right)^2}} \quad (17)$$

Where '*BC*' is the Bhattacharyya Coefficient, '*loc*' is the local similarity measure. Eqs. 16 and 17 are used for calculating local similarity. Eq. 16 is calculated by using the mean as a reference; similarly, Eq.17 is calculated by using the median as a reference.

Generally, the Jaccard similarity measure only counts the frequency of co-rated ratings between the users. This is one of the shortcomings in the Jaccard measure. To overcome this issue, Bag et al., [31] have proposed the Relevant Jaccard Similarity Measure (*RJACC*), which includes the frequency of un-co-rated items. Besides, a new similarity measure for CF-based RS is evolved by combining the *RJACC* and *MSD* which is termed as Relevant Jaccard Mean Squared Deviation (*RJMSD*) [31].

$$sim\left(u_j, u_h\right)^{RJACC} = \frac{1}{1+\frac{1}{\left|I_{u_j} \cap I_{u_h}\right|}+\frac{\left|\overline{I_{u_j}}\right|}{1+\left|\overline{I_{u_j}}\right|}+\frac{1}{1+\left|\overline{I_{u_h}}\right|}} \quad (18)$$

$$sim\left(u_j, u_h\right)^{RJMSD} = \frac{1}{1+\frac{1}{\left|I_{u_j} \cap I_{u_h}\right|}+\frac{\left|\overline{I_{u_j}}\right|}{1+\left|\overline{I_{u_j}}\right|}+\frac{1}{1+\left|\overline{I_{u_h}}\right|}} \times \left(1-\frac{\sum_{i=1}^{m'}\left(r_{u_j,I_i}-r_{u_h,I_i}\right)^2}{\left|I_{u_j} \cap I_{u_h}\right|}\right) \quad (19)$$

Where '$\overline{I_{u_j}}$ ' and '$\overline{I_{u_h}}$ ' are the number of un-co-rated items of user '*j*' and '*h*'. Sub one quasi norm-based similarity measure for CF-based RS is introduced to overcome the issues of similarity measure based on the Euclidean distance. [7]

$$sim\left(u_j, u_h\right)^{SOQN} = \sum_{i=1}^{m'} 1-\left(\frac{\left|r_{u_j,I_i}-r_{u_h,I_i}\right|^g}{\left(\frac{R_{ran}}{2}\right)^g}\right) \quad (20)$$

Where 'R_{ran}' is the range value, which is deviation between R_{max} and R_{min}. 'R_{max} ' is the maximum value on the rating scale. 'R_{min}' is the minimum value in the rating scale. '*g*' is the parameter which varies from 0 to 1.

2.1 Drawbacks in the existing similarity measure

PCC and *COS* are widely used methods in CF-based RS. The limitations related to PCC and COS are as follows: Flat value problem [18, 36], Single-value problem [36], Equal ratio problem [36], and Opposite value problem [36]. Specifically, Cosine similarity is one of the popular similarity measures used in many research applications. This is highly used in the area of text clustering to find out the similarity between the two documents [37], which is also used in the *k*-means clustering algorithm to compute the similarity between the data objects and the corresponding centroids. Based on the similarity values, the data objects are grouped into different clusters. It is also used in a multi-objective function for the Krill Herd algorithm. This similarity measure provides higher accuracy than conventional methods. [38]. Jaccard similarity measure is very highly known as a binary similarity measure. This is mainly adopted to find a similarity between the binary variables. This measure provides a better solution in the evolutionary algorithm, such as fitness function for the genetic algorithm [39]. The research, as mentioned above, clearly shows that cosine and Jaccard based similarity measures are highly used in real-life applications to solve various problems. In the case of CF-based RS, the rating values (ordinal scale) are used. The *PCC* and *COS* provide equal weightage for both the positive and negative sides of a rating scale which leads to misleading similarity values. This is one of the shortcomings in the *PCC* and Cosine based similarity measures. In Jaccard similarity measure, only the number of co-rated ratings is considered, not the intensity of the rating. These are some of the shortcomings in existing similarity measures. To overcome this shortcoming, Ahn [18] has proposed a *PIP* similarity measure. The magnitude of the *PIP* values is higher and each component in *PIP* measures treated in different proportions in different scenarios. *PSS* is the extension of the *PIP* expression with a non-linear assumption but the agreement conditions used in *PIP* measure is not included in this expression; thus, the correlation between *PIP* and *PSS* is very low. *NHSM* is a combination of the *PSS*, modified Jaccard (*JACC′*) and User Preference Rating (*URP*) measure. In *URP*, the deviation of the mean is multiplied with the standard deviation. If both users have the same mean but a different standard variation in this situation, standard variation becomes negligible. *BCF* similarity measure is a combination of global and local similarity measures. The Bhattacharya coefficient is treated as a global measure. In this expression, only the rating number is used in the calculation; co-rated items are not considered. Relevance Jaccard (*RJACC*) is the extension of *JACC* based similarity measure, in this measure, only the frequency of the co-rated and un-co-rated items are considered, the intensity (i.e., magnitude of the rating) is not considered for the similarity computation. *SQON* is the improved version of Euclidean distance, if the number of co-rated items gets increased then the magnitude of this similarity values also get increased. If two users having higher number of co-rated items, then the *SQON* provides higher similarity values for the users and vice-versa. These are the shortcomings in the existing similarity measure used in CF-based RS. To overcome these shortcomings, a modified *PIP* similarity measure has been proposed to get an improved similarity matrix for CF-based RS.

3 Detailed analysis of *PIP*

The *PIP* similarity measure [18] comprises two agreement conditions, which in turn include three components: proximity, impact, and popularity. The similarity between the two users ratings, 'r_{u_j,I_i}' and 'r_{u_h,I_i}' is calculated as follows:

$$\begin{aligned} PIP\left(r_{u_j,I_i}, r_{u_h,I_i}\right) &= proximity\left(r_{u_j,I_i}, r_{u_h,I_i}\right) \\ &\times impact\left(r_{u_j,I_i}, r_{u_h,I_i}\right) \\ &\times popularity\left(r_{u_j,I_i}, r_{u_h,I_i}\right) \end{aligned} \tag{21}$$

$$sim\left(u_j, u_h\right)^{PIP} = \sum_{i=1}^{m'} PIP\left(r_{u_j,I_i}, \mathrm{r}_{u_h,I_i}\right) \tag{22}$$

The agreement conditions play a vital role in the *PIP* measure. Compared to other similarity measures, the *PIP* is the only similarity measure that differentiates the positive and negative ratings by using the agreement conditions. This agreement conditions are used to discriminate the ratings based on pattern of ratings provided by two users. Let us consider the set of the users $U = \{u_1, u_2, \ldots, u_n\}$ and set of items $I = \{I_1, I_2, \ldots, I_m\}$, Where '$n$' is the total number of users, and 'm' denotes the total number of items. They are associated with the rating matrix called $\langle user \times item \rangle$ rating matrix

$$\begin{array}{c} \\ u_1 \\ \vdots \\ u_n \end{array} \begin{array}{c} \begin{array}{ccc} I_1 & \cdots & I_m \end{array} \\ \begin{bmatrix} r_{u_1,I_1} & \cdots & r_{u_1,I_m} \\ \vdots & \ddots & \vdots \\ r_{u_n,I_1} & \cdots & r_{u_n,I_m} \end{bmatrix} \end{array} \tag{23}$$

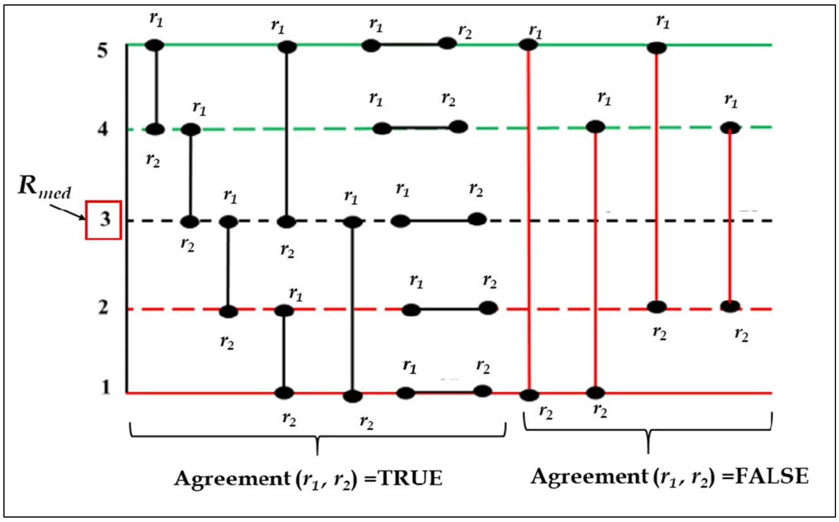
To understand better, let us consider 'r_1' and 'r_2' are the two ratings, where r_1 is the rating provided by user 'j' for item 'i' and r_2 is the rating provided by user 'h' for item 'i'. Where 'R_{max}' is the maximum rating in the rating scale and 'R_{min}' is the minimum rating in the rating scale; further let $R_{med} = \frac{R_{max}+R_{min}}{2}$, if both rating '$r_1$' and '$r_2$' are lesser or greater than 'R_{med}'. This means $Agreement(r_1, r_2)$ belongs to *TRUE* situation. Similarly, if the rating is in the opposite direction, i.e., a rating is greater than 'R_{med}' and another rating is lesser than 'R_{med}' the situation is *FALSE*. The main use for this agreement conditions is to discriminate against the rating. A Boolean function for the agreement conditions of ratings 'r_1' and 'r_2' is defined as follows:

$$\begin{array}{c} Agreement(r_1, r_2) = FALSE \\ if\ (r_1 > R_{med}\ and\ r_2 < R_{med})\ or\ (r_1 < R_{med}\ and\ r_2 > R_{med}) \end{array} \tag{24}$$

$$\begin{array}{l} Agreement(r_1, r_2) = TRUE \\ \qquad otherwise \end{array} \tag{25}$$

The pictorial representation of agreement conditions by considering r_1 as a reference is shown in Fig. 1. (A rating scale1 to 5 is considered for plotting this figure). The vertical lines denote the rating inside the plot, and the vertical black line indicates the Agreement of r_1 and r_2 is *TRUE*; similarly, the vertical red line denotes the Agreement of r_1 and r_2 is *FALSE*.

Fig. 1 Graphical representation of Agreement conditions


5
4
3
2
1
R_{med}
r_1
r_2
Agreement (r_1, r_2) =TRUE
Agreement (r_1, r_2) =FALSE

If both the users provide similar kind of ratings, the *Agreement*(r_1, r_2) = *TRUE*. The minimum and maximum deviation of ratings in *Agreement*(r_1, r_2) = *TRUE* situation are 0 and 2; this shows that both the users agree with the rating values. If both the users provide different kinds of rating for the same item then the *Agreement*(r_1, r_2) = *FALSE*. The minimum deviation of *Agreement*(r_1, r_2) = *FALSE* is two and the maximum deviation value is four. This clearly shows that the deviation between the two ratings provides useful information to calculate the similarity between the users. This agreement condition helps to differentiate the users based on the variation of the ratings provided to the co-rated item.

3.1 Proximity

Generally, proximity is used to determine the closeness of the two objects. Here proximity is the absolute difference between two rating r_1 and r_2.

$$D(r_1, r_2) = |r_1 - \mathrm{r}_2| \quad if\ Agreement(r_1, r_2) = TRUE \tag{26}$$

$$D(r_1, r_2) = 2 \times |r_1 - r_2| \quad if\ Agreement(r_1, r_2) = FALSE \tag{27}$$

$$proximity(r_1, r_2) = \{(2 \times (R_{max} - R_{min}) + 1) - D(r_1, r_2)\}^2 \tag{28}$$

Figure 2 (a) is the graphical representation of the proximity values plotted for the rating scale of 1 to 5. The minimum proximity value for *Agreement* (r_1, r_2) = *TRUE* condition is 49. This situation may occur when two ratings are not equal ($r_1 \neq r_2$). If both the ratings are same (i.e., $r_1 = r_2$), then the proximity value is 81. Similarly, for the *Agreement* (r_1, r_2) = *FALSE*, condition the minimum value is 1 and the maximum value is 25. The graphical representation of the proximity values for different combinations is shown in Fig. 2 (a).

3.2 Impact

The impact is the second critical component computed in the *PIP* formula, which provides the information regarding how strong the user preferred or disliked the particular item.

$$Impact(r_1, r_2) = (|r_1 - R_{med}| + 1) \times (|r_2 - R_{med}| + 1) \quad if\ Agreement(r_1, r_2) = TRUE \tag{29}$$

$$Impact(r_1, r_2) = \frac{1}{(|r_1 - R_{med}| + 1) \times (|r_2 - R_{med}| + 1)} \quad if\ Agreement(r_1, r_2) = FALSE \tag{30}$$

Impact in the range 0.11 to 9. In the impact factor, if *Agreement* (r_1, r_2) = *TRUE* the minimum is 1, and the maximum value is 9. For the *Agreement* (r_1, r_2) = *FALSE* the minimum is 0.11, and the maximum value is 0.25. The impact value for different combinations is shown in Fig. 2(b).

3.3 Popularity

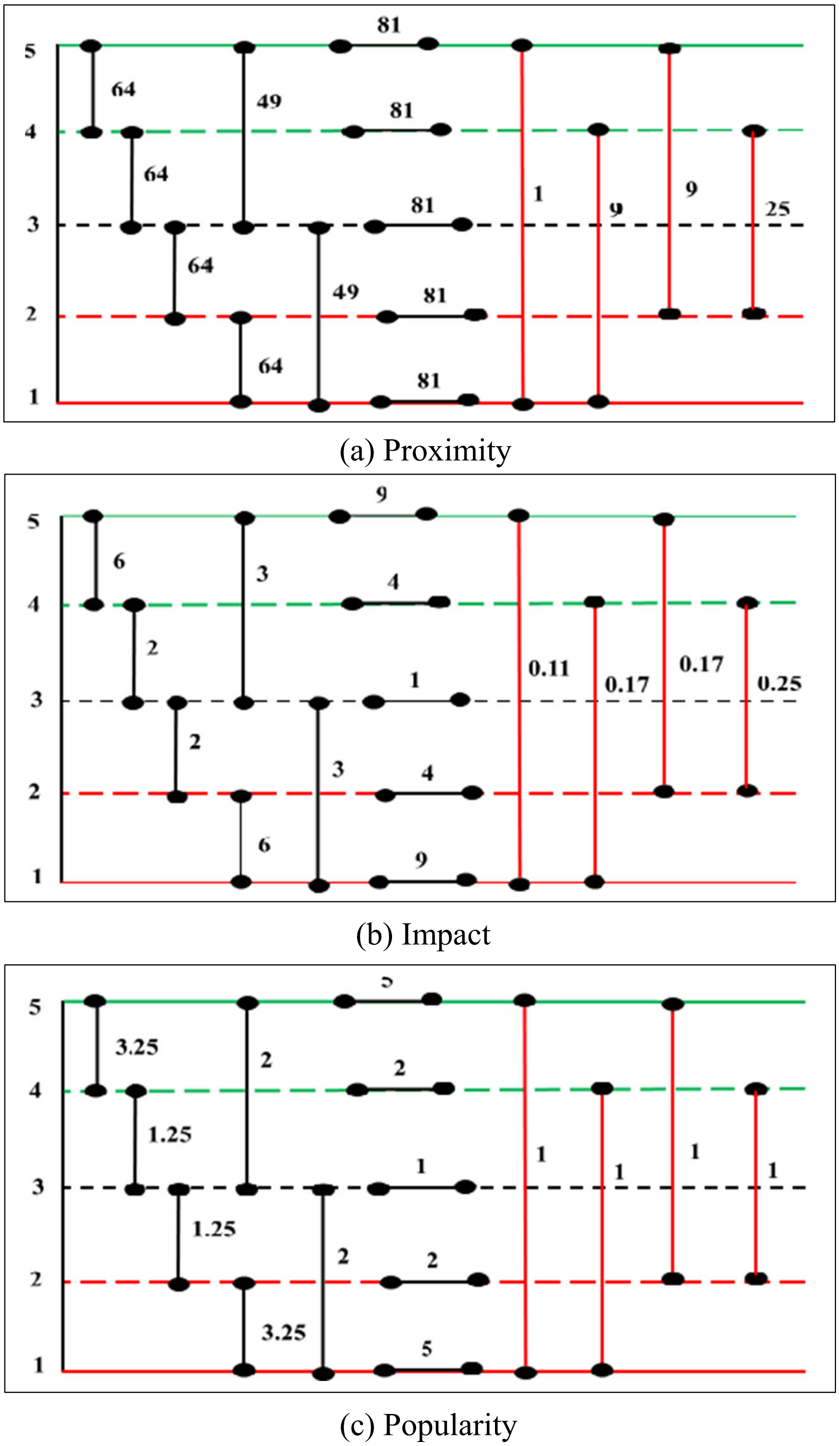
Popularity is calculated based on the deviation of the average rating for the item '*i*'. Let '$\bar{r}_{I_i}$ ' be the average rating of item '*i*' by all users.

$$Popularity(r_1, r_2) = 1 + \left(\left(\frac{r_1 + r_2}{2}\right) - \bar{r}_{I_i}\right)^2 \quad if\left(r_1 > \bar{r}_{I_i}\ and\ r_2 > \bar{r}_{I_i}\ or\ r_1 < \bar{r}_{I_i}\ and\ r_2 < \bar{r}_{I_i}\right) \tag{31}$$

$$Popularity(r_1, r_2) = 1 \quad otherwise \tag{32}$$

In popularity factor, if *Agreement* (r_1, r_2) = *TRUE*, the minimum value is 1, and the maximum value is 5. For computation purpose, the value of $\bar{r}_{I_i}$ is considered as 3. For the *Agreement* (r_1, r_2) = *FALSE* the minimum, and the maximum

Fig. 2 Graphical comparison of *PIP* values for rating scale 1–5 (**a**) proximity values; (**b**) impact values; (**c**) popularity values



(a) Proximity

(b) Impact

(c) Popularity

value is 1. The popularity value for different combinations is represented in Fig. 2(c).

The graphical comparison indicates that the proximity has higher magnitude values than the impact and popularity for *Agreement* $(r_1, r_2) = TRUE$ condition. In the case of *Agreement* $(r_1, r_2) = FALSE$ condition, the impact value is very minimal when compared to the other two components. So that the similarity value is highly dependent on the proximity and popularity values than the impact in the *FALSE* condition. A detailed explanation is given in the below sub-section.

3.4 Issues identified in *PIP*

In *PIP*, each component value lies in the wider range. Proximity, impact, and popularity are treated in different proportions., The calculation is listed for the rating scale 1 to 5.

The maximum value is calculated by considering r_1 as 5 and r_2 as 5 for *Agreement* $(r_1, r_2) = TRUE$ situation. The minimum value is computed by considering r_1 as 1 and r_2 as 5 for *Agreement* $(r_1, r_2) = FALSE$ condition, which is the extreme rating provided by both users (u_j, u_h) for item 'I_i'.

The minimum and maximum proximity values for *Agreement* $(r_1, r_2) = TRUE$ are 49, and 81, respectively. Similarly, if *Agreement* $(r_1, r_2) = FALSE$ the minimum and maximum values are 1 and 9. The range value for proximity is 80. If *Agreement* $(r_1, r_2) = TRUE,$ the minimum and maximum impact values are 1 and 9. Similarly, if *Agreement* $(r_1, r_2) = FALSE$ the minimum and maximum values are 0.11 and 0.25, respectively. The impact range is 8.89. If *Agreement* $(r_1, r_2) = TRUE$, the minimum and maximum values for popularity are 1 and 5. Similarly, if *Agreement* $(r_1, r_2) = FALSE$ the minimum and maximum values are both 1. The range for popularity is 4.

If the 'u_j' rating is 5, and the 'u_h' rating is 5; this situation leads to higher values for all the three components; i.e., 81 for proximity, 9 for impact, and 5 for popularity. The *PIP* is the product of proximity, impact and popularity; the resultant value is 3645. In this calculation, the proximity has a greater weight than the impact and popularity. Each component results in a different proportion. Similarly, for the worst scenario; i.e., if the user 'u_j' rating is 5, and the 'u_h' rating is 1, then proximity is 1, the impact is 0.11, and popularity is 1. The final '$PIP(r_1, r_2)$' value is 0.11, which is very small. Here, we see that proximity and popularity have equal weight, but the impact has much less value. The maximum *PIP* is 3645, and the minimum *PIP* is 0.11. This leads to a greater difference between the minimum and maximum value. We conclude that each factor provides important information for computing the similarity between users. Yet, their different range of values shows that any one of the components provides higher values in different scenarios. This unequal scaling method provides different ranges of values for different scenarios, and the values of each component are non-normalized. Suppose direct normalization procedure is adopted to convert the values into particular range, then the values get changed for different scenario. This leads to lesser prediction accuracy [40], which constitutes a major drawback of the existing *PIP* similarity measure.

4 Proposed method

Our proposed framework aims to provide an improved solution for the CF-based RS with a sparse data matrix. It consists of a modified PIP (*MPIP*) measure and a combined user and item-based prediction (*CUIP*) expression.

4.1 Modified PIP (*MPIP*) similarity measure

In the existing *PIP* expression, each component has different values in different scenarios, and greater priority can be given to anyone of the components in different situations. To avoid this situation, the component ranges are converted into zero to one in our modified similarity measure by changing the expressions.

4.1.1 Proposed proximity

Proposed proximity is the normalized value that ranges from zero to one, calculated using absolute deviation between the two ratings r_1 and r_2.

$$D(r_1, r_2) = |r_1 - r_2| \tag{33}$$

$$Proposed\ Proximity(r_1, r_2) = \left(\frac{D(r_1, r_2) - \frac{(med^{+} + med^{-})}{2}}{R_{max} - R_{min}}\right)^2 \quad if\ Agreement(r_1, r_2) = TRUE \tag{34}$$

Where med^{+} is the median value of the positive rating (i.e., the rating above or equal to the median value of the rating scale), med^{-} is the median value of the negative rating (i.e., the rating value below the median value of the rating scale). $Agreement(r_1, r_2) = TRUE$ condition the absolute difference between two ratings is subtracted from the average of positive and negative median values. To get a normalized value within the range of 0 to 1 the deviation of R_{max} and R_{min} values are used. The positive and negative median value is included in the expression to find the closeness of the rating.

$$Proposed\ Proximity(r_1, r_2) = \delta * \left(\frac{\frac{1}{D(r_1, r_2)}}{R_{max} - R_{min}}\right)^2 \quad if\ Agreement(r_1, r_2) = FALSE \tag{35}$$

$Agreement(r_1, r_2) = FALSE$ the inverse of the deviation term is used to calculate the proximity value. Our reframed expression provides higher weightage for $Agreement(r_1, r_2) = TRUE$ condition and less weightage for $Agreement(r_1, r_2) = FALSE$ condition. $Agreement(r_1, r_2) = FALSE$ condition, the positive and negative median values are not included because both the users are on the different side of the rating scale values. In both situations, the values are within the range of 0 to 1.

$$\delta = \begin{cases} 0.75 & if\ D(r_1, r_2) > R_{med} \\ 0.5 & else\ if\ D(r_1, r_2) = R_{med} \\ 0.25 & otherwise \end{cases} \tag{36}$$

In *MPIP* variable penalty (δ) is multiplied with the proximity value. The higher penalty is weighed for higher deviation value, and a lesser penalty is given for lesser deviation values. This conversion reduces the magnitude of the value.

4.1.2 Proposed impact

The exponential based expression is used to compute the impact value for the *Agreement* $(r_1, r_2) = TRUE$ case. This helps to get a normalized impact value for the *TRUE* situations.

$$proposed\ impact(r_1, r_2) = exp\left(-\frac{1}{(|r_1 - R_{med}| + 1) \times (|r_2 - R_{med}| + 1)}\right) \quad if\ Agreement(r_1, r_2) = TRUE \tag{37}$$

If r_1 and r_2 are far away from the median, the impact value is higher. When both the ratings are nearer to the median, the impact provides lesser value. This shows that both the users agree with the median values, so the impact of the ratings is very less.

$$proposed\ impact\ (r_1, r_2) = \frac{1}{(|r_1 - R_{med}| + 1) \times (|r_2 - R_{med}| + 1)} \quad if\ Agreement(r_1, r_2) = FALSE \tag{38}$$

In the existing *PIP* measure the impact values for *Agreement* $(r_1, r_2) = FALSE$ condition, are lie within the range of zero to one. Therefore, the same expression is used in the proposed impact for *FALSE* condition .

4.1.3 Proposed popularity

Popularity is the third component in *PIP* similarity measure, which includes both positive and negative popularity in this expression.

$$proposed\ popularity(r_1, r_2) = log_{10}\left(2 + \left(\frac{r_1 + r_2}{2} - \overline{r}_{I_i}\right)^2\right) \quad if\ \left(r_1 > \overline{r}_{I_i}\ and\ r_2 > \overline{r}_{I_i}\ or\ r_1 < \overline{r}_{I_i}\ and\ r_2 < \overline{r}_{I_i}\right) \tag{39}$$

$$Proposed\ popularity(r_1, r_2) = 0.3010 \quad otherwise \tag{40}$$

If two ratings are on the same side of the rating scale and the average of two ratings is far away from the item mean, then the popularity is very high. For popular or unpopular items, two users provide similar kinds of ratings, and then, they have high similarity in the type of rating. In the existing method, the minimum value for the *Agreement* $(r_1, r_2) =$ *TRUE* situation is 1. Based on this, the popularity value is set to 1 for *Agreement* $(r_1, r_2) = FALSE$ condition. In the proposed method, the minimum value for *Agreement* $(r_1, r_2) =$ *TRUE* is 0.3010; So this value is chosen for all the *Agreement* $(r_1, r_2) = FALSE$ condition. The proposed popularity values range from 0.3010–0.778 for the 1–5 rating scale, so that the minimum popularity value is chosen for all *Agreement* $(r_1, r_2) = FALSE$ situations.

According to Fig. 3, the proposed proximity, proposed impact and proposed popularity values are within the range of zero to one; in contrast to the existing method, which provides different weights for each component.

4.1.4 Similarity measure computation

The *MPIP* expression is the product of proposed proximity, proposed impact, and proposed popularity, as shown in Eq. 41

$$MPIP\left(r_{u_j,\mathrm{I}_i}, r_{u_h,I_i}\right) = \left\{\begin{array}{c} Proposed\ proximity\left(r_{u_j,I_i}, r_{u_h,I_i}\right) \times \\ Proposed\ impact\left(r_{u_j,I_i}, r_{u_h,I_i}\right) \times \\ Proposed\ popularity\left(r_{u_j,I_i}, r_{u_h,I_i}\right) \end{array}\right\} \tag{41}$$

The similarity between users is computed as follows:

$$sim\left(u_j, u_h\right)^{uMPIP} = \sum_{i=1}^{m'} MPIP\left(r_{u_j,I_i}, r_{u_h,I_i}\right) \tag{42}$$

Similarity between 'u_j' and 'u_h' ($sim(u_j, u_h)$) is the item-wise summation of *MPIP* values.

The same procedure is used for computing the similarity between items, which helps to find a similar kind of items. The maximum similarity values show that most of the users provide a similar pattern of rating for the two items. Likewise, the minimum similarity value shows that most of the users provide a high positive rating for one item and a higher negative rating for another item.

$$sim\left(I_i, I_q\right)^{iMPIP} = \sum_{j=1}^{n'} MPIP\left(r_{u_j,I_i}, r_{u_j,I_q}\right) \tag{43}$$

A comparison of the minimum and maximum values is shown in Table 2.

Case 1: Positive or Negative rating.

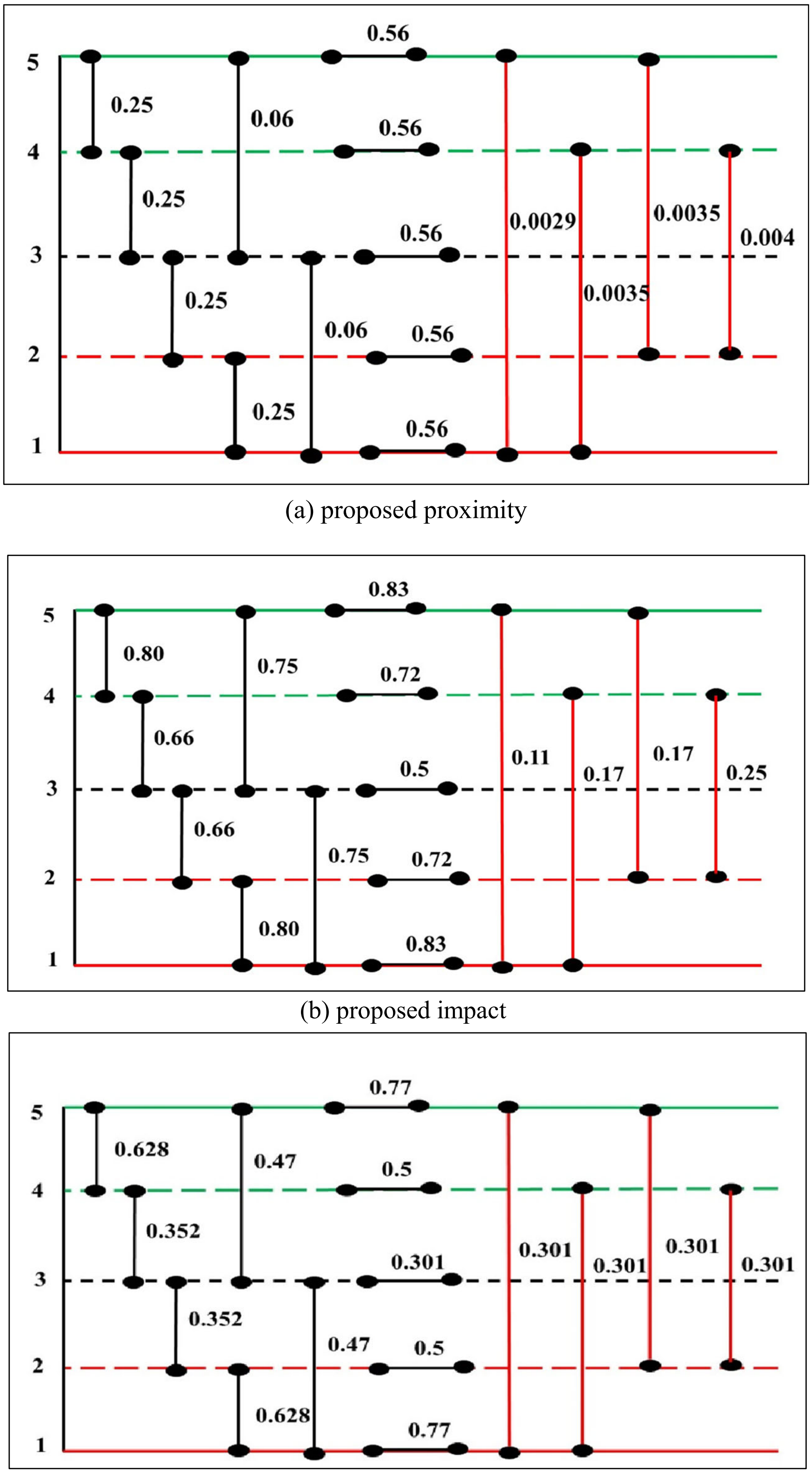
Both u_j and u_h have a higher positive or negative rating; i.e., the extreme positive rating means both users u_j and u_h gives rating 5 for the item 'i' otherwise extreme negative rating means both the users u_j and u_h offers rating 1 for the item 'i'; the existing *PIP* method provides a proximity value of 81, an impact of 9, popularity of 5, and a *PIP* value of 3645. Approximately 85% of weightage is given for the proximity components. When using the *MPIP*, the proposed proximity is 0.56, the proposed impact is 0.89, the proposed popularity is 0.778, and the *MPIP* value is 0.389.

Case 2: Median rating.

If u_j and u_h have a median rating, i.e., both the users u_j and u_h gives rating 3 for the item 'i', then the proximity is 81, the impact is 1, the popularity is 1, and the *PIP* value is 81. For the *MPIP*, the proposed proximity is 0.56, the proposed impact is 0.50, the proposed popularity is 0.301, and the *MPIP* is 0.084.

Case 3: Difference of opinion.

If u_j and u_h provides different extreme ratings for the item 'i', i.e., u_j provides rating 1and u_h gives rating 5 or vice versa, then the proximity is 1, the impact is 0.11, popularity is 1, and the PIP value is 0.11. For the *MPIP*, the proposed proximity is 0.002, the proposed impact is 0.11, the proposed popularity is 0.301, and the *MPIP* value is 0.0001. *MPIP* values nearer to zero for difference of opinion situation, i.e., very minimal weightage is given for the ratings because both the ratings lie on the extreme value of the rating scale. There is no relationship existing between the two users. This comparison proves that the *MPIP* values lie within a range of 0 to 1, but the existing method has a vast range.



(c) proposed popularity

Fig. 3 Graphical representation of *MPIP* the values are calculated for rating scale 1–5 (**a**) proposed proximity (**b**) proposed impact and (**c**) proposed popularity

4.2 Validation for *MPIP* similarity measure

In the *MPIP*, the maximum and minimum values range from 0 to 1. To validate the proposed similarity measure (i.e., Modified PIP), a rank correlation test is conducted. A set of rating pairs is generated for the rating scale 1 to 5. The *PIP*, *PSS* and *MPIP* values are computed for each pair of ratings. We generated the set of *PIP*, Proposed proximity, impact, popularity values and *PSS* values. The values generated for each pair of ratings are listed in the Table 3.

Table 2 Comparison of *PIP* and *MPIP* similarity measures showing the minimum and maximum value for each component

	Agreement(r_1, r_2) = *TRUE*		*Agreement*(r_1, r_2) = *FALSE*	
	Minimum	Maximum	Minimum	Maximum
Proximity	49	81	1	9
Proposed Proximity	**0.06**	**0.56**	**0.0029**	**0.004**
Impact	1	9	0.11	0.25
Proposed Impact	**0.37**	**0.89**	**0.11**	**0.25**
Popularity	1	5	1	1
Proposed Popularity	**0.301**	**0.78**	**0.301**	**0.301**
PIP	49	3645	0.11	2.25
MPIP	**0.007**	**0.389**	**0.0001**	**0.0003**

The magnitude of the values gets changed in the case of *MPIP*. In *PSS* method, each component is ranged from zero to one, but the position of the values gets changed because it provides equal weightage for both agreement and disagreement conditions. The rank correlation test is conducted for this set of values; the results are shown in Table 4.

The correlation values of *PIP* and *MPIP* components are one. This shows that a strong positive relationship exists between the *PIP*, and *MPIP* components. Owing to the violation of the agreement conditions, the *PSS* measure has much lower correlation values. The correlation between proximity and PSS proximity is higher. In the case of impact and popularity, the values are minimal. At the same time, this shows that a negative relationship exists between impact and PSS significance term. Similarly, there is a negative relationship existing between popularity and singularity. These results clearly show that the violation of agreement condition provides misleading information for the similarity between two ratings. Due to this reason, the same agreement conditions are used in our proposed method. The high correlation between the *PIP* and *MPIP* components reveals that the magnitude of the values gets changed in our proposed method, but the equivalent proportion is retained.

Table 3 Comparison of *PIP*, *MPIP* and *PSS* values set of rating pairs

r_1	r_2	Proximity	Impact	Popularity	Proposed Proximity	Proposed Impact	Proposed Popularity	Proximity	Significance	Singularity
5	5	81.00	9.00	5.00	0.563	0.833	0.778	0.500	0.982	0.119
4	5	64.00	6.00	3.25	0.250	0.800	0.628	0.269	0.881	0.182
3	5	49.00	3.00	1.00	0.063	0.750	0.301	0.119	0.500	0.269
2	5	9.00	0.17	1.00	0.0035	0.167	0.301	0.047	0.881	0.378
1	5	1.00	0.11	1.00	0.0029	0.111	0.301	0.018	0.982	0.500
5	4	64.00	6.00	3.25	0.250	0.800	0.628	0.269	0.881	0.182
4	4	81.00	4.00	2.00	0.563	0.750	0.477	0.500	0.731	0.269
3	4	64.00	2.00	1.00	0.250	0.667	0.301	0.269	0.500	0.378
2	4	25.00	0.25	1.00	0.004	0.250	0.301	0.119	0.731	0.500
1	4	9.00	0.17	1.00	0.0035	0.167	0.301	0.047	0.881	0.622
5	3	49.00	3.00	1.00	0.063	0.750	0.301	0.119	0.500	0.269
4	3	64.00	2.00	1.00	0.250	0.667	0.301	0.269	0.500	0.378
3	3	81.00	1.00	1.00	0.563	0.500	0.301	0.500	0.500	0.500
2	3	64.00	2.00	1.00	0.250	0.667	0.301	0.269	0.500	0.622
1	3	49.00	3.00	1.00	0.063	0.750	0.301	0.119	0.500	0.731
5	2	9.00	0.17	1.00	0.0035	0.167	0.301	0.047	0.881	0.378
4	2	25.00	0.25	1.00	0.004	0.250	0.301	0.119	0.731	0.500
3	2	64.00	2.00	1.00	0.250	0.667	0.301	0.269	0.500	0.622
2	2	81.00	4.00	2.00	0.563	0.750	0.477	0.500	0.731	0.731
1	2	64.00	6.00	3.25	0.250	0.800	0.628	0.269	0.881	0.818
5	1	1.00	0.11	1.00	0.0029	0.111	0.301	0.018	0.982	0.500
4	1	9.00	0.17	1.00	0.0035	0.167	0.301	0.047	0.881	0.622
3	1	49.00	3.00	1.00	0.063	0.750	0.301	0.119	0.500	0.731
2	1	64.00	6.00	3.25	0.250	0.800	0.628	0.269	0.881	0.818
1	1	81.00	9.00	5.00	0.563	0.833	0.778	0.500	0.982	0.881

Table 4 Correlation *MPIP*, *PIP*, and *PSS*

Combinations	Correlation Value
ρ(Proximity, Proposed Proximity)	**1****
ρ(Impact, Proposed Impact)	**1****
ρ(Popularity, Proposed Popularity)	**1****
ρ(PSS-Proximity, Proposed Proximity)	**0.99****
ρ(PSS-Significance, Proposed Impact)	**−0.01**
ρ(PSS-Singularity, Proposed Popularity)	**−0.07**

*** denotes 0.01 significance level*

4.3 Existing prediction expression

The main objective of the CF-based RS is to predict user ratings based on the similarity measure. Many expressions are used in the prediction process. This user mean-based prediction expression, which is a widely used method that provides a better solution for CF-based RS [5, 18, 36, 41–43], is shown as follows:

$$P_{u_j,I_i} = \overline{r}_{u_j} + \frac{\sum_{h=1}^{n'} sim\left(u_j,u_h\right)\left(r_{u_h,I_i} - \overline{r}_{u_h}\right)}{\sum_{h=1}^{n'} sim\left(u_j,u_h\right)} \tag{44}$$

In this expression, 'P_{u_j,I_i} ' is computed based on the mean of the user '*j*' which is added with the weighted average deviation of user '*h*'. The weighted average deviation of user '*j*' is not considered in the prediction process. In weighted deviation, only user-related information is used, item-related information is not included in the prediction expression.

From Fig. 4, the user-related mean is considered for user-based prediction, and the item-related mean is considered for item-based prediction. In the sparse user-item rating matrix, both the mean and deviation provide important information for the rating prediction. This is one of the shortcomings of conventional user-related or item-related prediction expression.

4.3.1 Combined user and item-based prediction expression (*CUIP*)

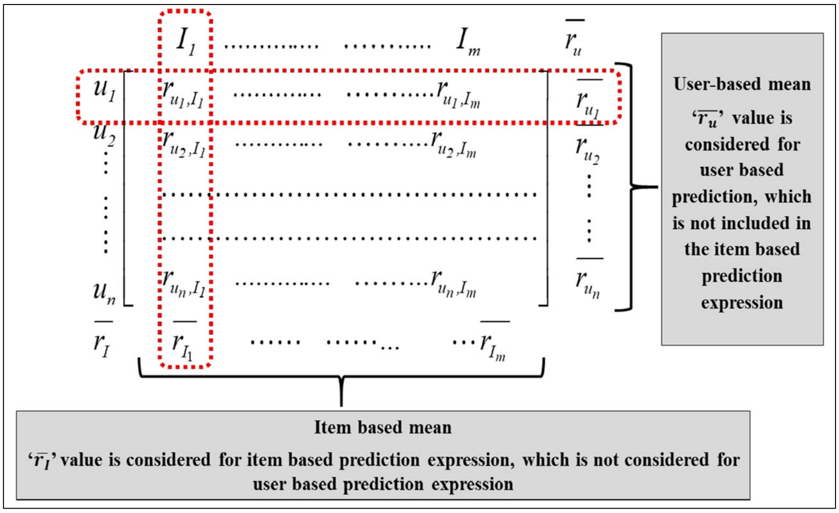
A modified prediction expression is derived by incorporating user '*j*' related components in user-based prediction. Similarly, the item '*i*' related components are included in the item-based prediction.

Ma and Hu [44] used a hybrid prediction expression for predicting the rating. In this expression, the additional weight parameter 'λ' is multiplied by the user based prediction. Similarly, '1- λ' weight is multiplied for the item based prediction. 'λ' value varies between 0 to 1. If 'λ' equals one, it purely depends on user-based prediction; if 'λ' equals zero, it purely depends on item-based prediction. Thus, 'λ' is an additional parameter that should be optimized for a different problem.

In the modified prediction expression, the average values of user-based and item-based prediction expression are introduced by providing equal weightage for both the expressions. The modified prediction expression is shown below.

$$UP_{u_j,I_i} = \overline{r}_{u_j} + \frac{\sum_{h=1}^{n'} sim\left(u_j,u_h\right) \times \left(\frac{\left(r_{u_h,I_i} - \overline{r}_{u_h}\right) + \left(r_{u_j,I_i} - \overline{r}_{u_j}\right)}{2}\right)}{\sum_{h=1}^{n'} sim\left(u_j,u_h\right)} \tag{45}$$

Where '*UP*' is the user based prediction, $j \neq h, j = \{1, 2\ldots.., n\}, h = \{1, 2\ldots\ldots, n'\}$, and n' is the number of items co-rated by both the users '*j*' and '*h*'.



User-based mean '$\overline{r}_u$' value is considered for user based prediction, which is not included in the item based prediction expression

Item based mean '$\overline{r}_I$' value is considered for item based prediction expression, which is not considered for user based prediction expression

Fig. 4 Issues in user and item-based prediction expression

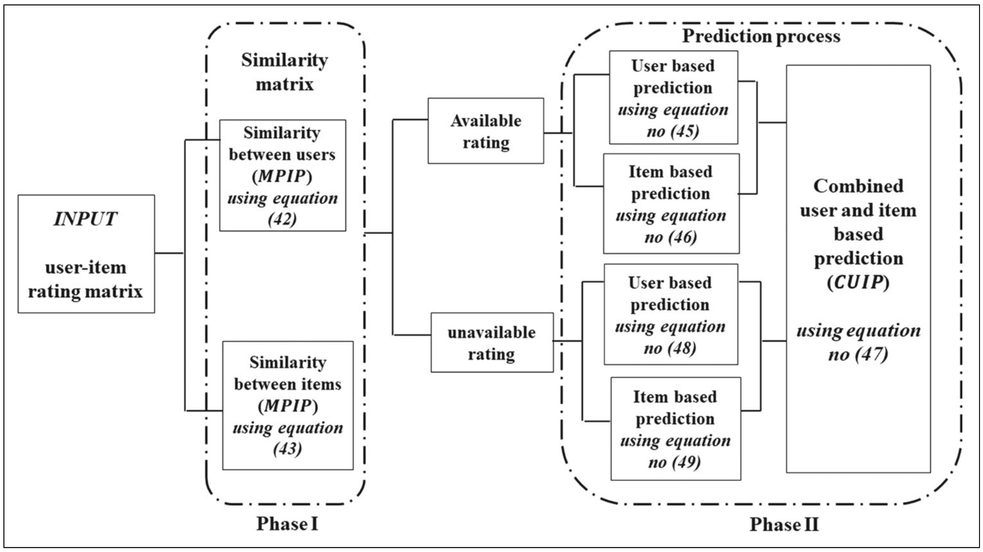

INPUT
user-item rating matrix
Similarity matrix
Similarity between users (MPIP) using equation (42)
Similarity between items (MPIP) using equation (43)
Phase I
Available rating
unavailable rating
Prediction process
User based prediction using equation no (45)
Item based prediction using equation no (46)
User based prediction using equation no (48)
Item based prediction using equation no (49)
Combined user and item based prediction (CUIP) using equation no (47)
Phase II

Fig. 5 The proposed framework for CF-based RS

$$IP_{u_j,I_i} = \overline{r_{I_i}} + \frac{\sum_{q=1}^{m'} sim\left(I_i, I_q\right) \times \left(\frac{\left(r_{u_j,I_q} - \overline{r}_{I_q}\right) + \left(r_{u_j,I_i} - \overline{r}_{I_i}\right)}{2}\right)}{\sum_{q=1}^{m'} sim\left(I_i, I_q\right)} \tag{46}$$

Where '*IP*' is the item based prediction $i \neq q$, $i = \{1, 2\ldots.., m\}$,$q = \{1, 2\ldots\ldots, m'\}$, and m' is the number of users who rated both items '*i*' and '*q*'.

$$CUIP_{u_j,I_i} = \left(\frac{\left(UP_{u_j,I_i}\right) + \left(IP_{u_j,I_i}\right)}{2}\right) \tag{47}$$

The similarity values and deviation play a vital role in prediction expression. Either the user-related mean and its weighted average deviation or the item-related mean and its weighted average deviation are used in the existing prediction expressions. The user-related mean $\overline{r}_{u_j}$ and item-related mean $\overline{r}_{I_i}$ are calculated for user '*j*' and item '*i*', but the corresponding deviation is not included in the existing prediction expression. This is one of the shortcomings of the existing prediction expression. In the modified prediction expression (*CUIP*), the above-mentioned shortcoming is removed by including the user and item-related deviation term.

Forecasting is a vital process in CF-based RS. The forecasting values help the company to identify potential customers and then to promote sales. In *CUIP*, if the user gives rating for that particular item, i.e., $r_{u_j,I_i} \neq \emptyset$; then the deviations carry some values. Suppose the rating is unavailable i.e., $r_{u_j,I_i} = \emptyset$, then computing user-related and item-related deviations become complex. The average of the user (u_j) related mean and item (I_i) related mean is replaced in the unavailable rating to avoid this situation. For unavailable rating, the user based prediction is calculated using Eq. (48).

$$UP_{u_j,I_i} = \overline{r}_{u_j} + \frac{\sum_{h=1}^{n'} sim\left(u_j, u_h\right) \times \left(\frac{\left(r_{u_h,I_i} - \overline{r}_{u_h}\right) + \left(\frac{\overline{r}_{u_j} + \overline{r}_{I_i}}{2} - \overline{r}_{u_j}\right)}{2}\right)}{\sum_{h=1}^{n'} sim\left(u_j, u_h\right)} \tag{48}$$

Table 5 Summary of the dataset used for this study

Dataset	Description	*n*	*m*	#R	Sparsity	Rank
Epinions [d]	Rating of products in scale 1 to 5	49,290	1,39,738	6,64,824	99.99	1
CiaoDVD [e]	Rating of DVD products in scale of 1 to 4	17,615	16,121	72,665	99.97	2
MovieTweet [f]	Rating of Movies in scale of 1 to 10	51,503	29,564	6,56,276	99.95	3
FlimTrust [g]	Rating of Movies in scale of 0.5 to 4	1508	2071	35,497	98.64	4
NetFlix [c]	Rating of movies in scale of 1–5	50,000	17,770	1,03,75,459	98.83	5
MovieLens1MB (ML1MB) [a]	Rating of Movies in scale of 1–5	6040	3706	10,00,209	95.53	6
MovieLens100KB (ML100KB) [b]	Rating of Movies in scale of 1–5	943	1682	1,00,000	93.69	7

Similarly, the item-based prediction is calculated using Eq. (49)

$$IP_{u_j,I_i} = \overline{r}_{I_i} + \frac{\sum_{q=1}^{m'} sim\left(I_i, I_q\right) \times \left(\frac{\left(r_{u_j,I_q} - \overline{r}_{I_q}\right) + \left(\left(\frac{\overline{r}_{u_j} + \overline{r}_{I_i}}{2}\right) - \overline{r}_{I_i}\right)}{2}\right)}{\sum_{q=1}^{m'} sim\left(I_i, I_q\right)} \tag{49}$$

The Eq. (47) is used for computing the final prediction value for the user '*j*' for the item '*i*'. The combination of modified PIP (*MPIP*) similarity measure and modified prediction expression (*CUIP*) is treated as the proposed method. The algorithm for the proposed method is shown below:

Algorithm: Proposed Framework for CF-based RS

```
Input   : User-Item rating matrix
Output : Predicted rating matrix

            //Initialize//
$sim(u_j, u_h)^{uMPIP} = [\,]$,
            //where $sim(u_j,, u_h)^{uMPIP}$ is the similarity values between users//
$sim(I_i, I_q)^{iMPIP} = [\,]$,
           //where $sim(I_i, I_q)^{iMPIP}$ is the similarity values between items//
$CUIP_{u_j,I_i} = [\,]$,
          // where $CUIP_{u_j,I_i}$ is the predicted rating for the user 'j' and item 'i'//

          //Similarity Matrix computation//
For user $j = 1{:}\,n$ do
    For user $h = 1{:}\,n$ do
        For item $i = 1{:}\,m$ do
            For item $q = 1{:}\,m$ do
                If $\left|r_{u_j,I_i} \cap r_{u_h,I_i}\right| > 0$
                    $sim(u_j, u_h)^{uMPIP}$ using Eq (42)
                    $sim(I_i, I_q)^{iMPIP}$ using Eq (43)
                Else
                    $sim(u_j, u_h)^{uMPIP} = 0$
                    $sim(I_i, I_q)^{iMPIP} = 0$
                End if
            End For
        End For
    End For
End For

    // prediction for the $j^{th}$ user rating for the $i^{th}$ item //
For user $j = 1{:}\, n$ do
    For item $i = 1{:}\, m$ do
        if $r_{u_j,I_i} \neq \emptyset$
            $UP_{u_j,I_i}$ using Eq (45)
            $IP_{u_j,I_i}$ using Eq (46)
            $CUIP_{u_j,I_i}$ using Eq (47)
        Else
            $UP_{u_j,I_i}$ using Eq (48)
            $IP_{u_j,I_i}$ using Eq (49)
            $CUIP_{u_j,I_i}$ using Eq (47)
        End if
    End For
End For
Return predicted rating matrix
```

The block diagram for our proposed framework is shown in Fig. 5.

The proposed framework consists of two phases: The first phase is the computation of the similarity matrix (user similarity and item similarity matrix), and the second phase is the prediction process. Initially, it requires an input user-item rating matrix for computing similarity values. The similarity between users is computed by using *MPIP* expression (eq.(42)). Similarly, item similarity is computed by using *MPIP* expression (eq.(43)). Both user and item related similarity values provide valuable information for rating prediction. For available rating, the Eqs. (45), (46) and (47) are used to compute prediction values. In the case of unavailable ratings, the Eqs. (48), (49), and (47) are adopted for predicting the user's ratings. Based on the predicted values, the items are sorted in descending order; the top 'k' items are recommended to the users.

5 Experiments

Datasets such as MovieLens1MB, Netflix, Epinions, CiaoDVD, MovieTweet, FilmTrust, and MovieLens100KB are used for comparing the conventional and proposed methods. We used MovieLens100KB and Netflix, which are used in [18], and other benchmark datasets that are often used by many researchers for CF-based RS. To validate our proposed method eleven state of the art methods (*PCC*, *COS*, *JACC*, *JMSD*, *PIP*, *NHSM*, BCF_{corr}, BCF_{Med}, *RJACC*, *RJMSD*, and *SOQN*) are used for comparison.

5.1 Characteristics of sub-problems generated from the dataset

The dataset description, number of users, number of items, number of ratings, and sparsity are listed in Table 5. The dataset is arranged based on the rank of the sparsity (i.e., higher sparsity to lower sparsity).

[a] http://grouplens.org/datasets/movielens/1m
[b] http://grouplens.org/datasets/movielens/100k
[c] http://www.netflixprize.com
[d] http://www.trustlet.org/downloaded epinions.html.
[e] http://www.librec.net/datasets/CiaoDVD.zip
[f] https://github.com/sidooms/MovieTweetings
[g] http://www.librec.net/datasets/filmtrust.zip

The above-mentioned dataset size is large. Computing the similarity matrix and predicting the rating for this nature of the dataset requires higher computational complexity. Because of this, different levels of sparse subsets $\left(\left(1-\left(\frac{\#R}{n*m}\right)\right) *100\right)$ are generated from the user-item rating matrix to validate the proposed method. The schema used for creating different sub-problems is shown in Fig. 6.

The schema comprises four levels. Level 1 relates to the dataset; level 2 is for users, level 3 is for items, and level 4 is for rating. The user-level consists of two sub levels. One sub-level of users varies without any constraints. Another sub-level of users is restricted as 25%, 50%, and 75% of the users in the dataset. The next level is the item level, which consists of two sub-levels. One sub-level of items varies without any constraints. The item sub-level is restricted to 25%, 50%, and 75% of the items in the dataset. The final level consists of the rating. This level varies from 1% to 50% in the number of ratings with increments of 1% in the dataset.

For each dataset, one can create 800 different sub-problems by using the above schema. The number of sub-problems created by each path is shown in Table 6.

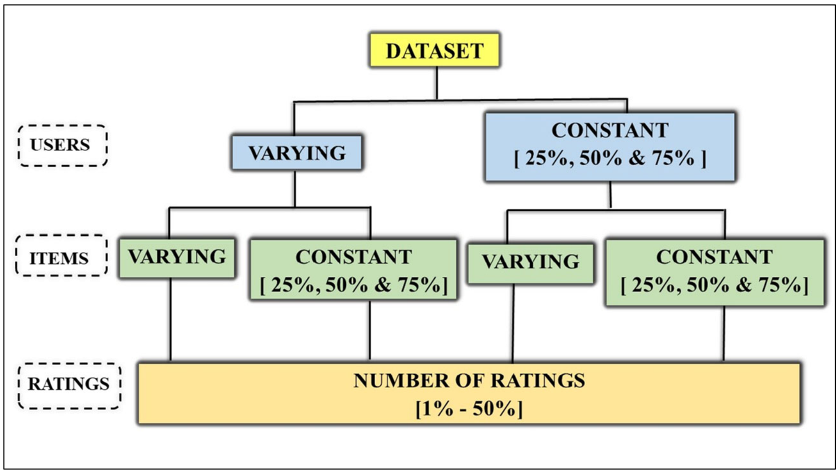


Fig. 6 Schema for generating different sub-problems

5.2 Limitation of the above schema

In this schema, the maximum percentage of users and items is 75%. For this specific combination, it is difficult to create sub-problems for all the ratings because the required percentage of ratings (1% to 50%) could not use for the higher-order matrix. For all the sub-problems, it is ensured that each user and item should have a minimum of two co-rated values. In each sub-problem, it is challenging to generate an exact 1% to 50% of ratings. To overcome this problem, we have an approximate percentage of rating values extracted. For each dataset, this results in lesser than 800 sub-problems, as shown in Table 6.

Table 7 provides the number of feasible sub-problems created for each dataset using the schema.

Table 6 Number of sub-problems generated by each path

Paths	Users	Items	Rating	No. of Sub-problems
1	Varying	Varying	1% to 50%	50
2	Varying	Constant	1% to 50%	
		25%		50
		50%		50
		75%		50
3	Constant	Varying	1% to 50%	
	25%			50
	50%			50
	75%			50
4	Constant	Constant	1% to 50%	
	25%	25%		150
	50%	50%		150
	75%	75%		150
Total				800

5.3 Performance criteria

The Mean Absolute Error (*MAE*) and Root Mean Squared Error (*RMSE*) are most commonly used evaluation criteria to validate the CF-based RS performance. [45, 46]

MAE is the mean of absolute deviation between the actual and the predicted rating.

$$MAE = \frac{1}{n}\sum_{j=1}^{n}\frac{1}{m}\sum_{i=1}^{m}\left|r_{u_j,I_i}-P_{u_j,I_i}\right| \tag{50}$$

Where ‘n’ is the total number of users, ‘m’ is the total number of items. ‘P_{u_j,I_i} ’ is the predicted rating for the j^{th} user and i^{th} item, r_{u_j,I_i} is the actual rating for the j^{th} user and i^{th} item, $j = \{1, 2, \ldots. n\}$, $i = \{1, 2, \ldots. . m\}$.

RMSE is the square root of the deviation between the actual and predicted rating. *RMSE* is also known as standard deviation of residuals or forecasting error. The formula is given below:

$$RMSE = \frac{1}{n}\sum_{j=1}^{n}\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(r_{u_j,I_i}-P_{u_j,I_i}\right)^2} \tag{51}$$

The lesser *MAE* and *RMSE* help to choose the better method. The predicted rating ($P_{u_j\ ,I_i}$) for the existing similarity measures such as *PCC*, *COS*, *JACC*, *JMSD*, *PIP*, *NHSM*,

Table 7 Number of feasible sub-problems

Dataset	No. of feasible Sub-problems
Epinions	296
CiaoDVD	305
MovieTweet	314
FlimTrust	323
NetFlix	327
ML1MB	341
ML100KB	349

Table 8 Contingency table to compare two methods

Method 2 Method 1	Correct	Incorrect
Correct	*A*	*B*
Incorrect	*C*	*D*

BCF_{corr}, BCF_{Med}, *RJACC*, *RJMSD*, and *SOQN* are calculated by using Eq. (44). The predicted rating values (P_{u_j} $,_{I_i}$) for the proposed method, are calculated by using Eqs. (45–47). Finally, the predicted ratings are rounded off to the nearest integer for both the existing and proposed method.

5.4 Friedman rank test

Friedman rank test is a non-parametric test for finding differences in methods across multiple replications [47, 48]. This test does not assume the given data that comes from particular distribution (i.e., normal distributions). The Friedman values are calculated by using the given equation

$$F_r = \frac{12*TS}{l(l+1)}\left(\sum_{y=1}^{l} R_y^2 - \frac{l(l+1)^2}{4}\right) \qquad (52)$$

Where F_r is the calculated Friedman value, R_y^2 is the squared average rank values of method '*y*', $(1 \le y \le l)$, '*l*' is the total number of methods, '*sp*' is the number of sub-problems generated, $sp = \{1, 2.... TS\}$, '*TS*' is the total number of sub-problems generated. The hypotheses framed for this test are: Null hypothesis (H_0) denotes there is no significant difference that occurs between the performance of the different methods, whereas Alternate hypothesis (H_1) denotes there is significant difference that occurs between the performance of the different methods. [49]

Table 9 Confusion matrix format for CF-based RS

Prediction Actual	Recommended (Good Rating)	Not Recommended (Bad Rating)
Good Rating	True Positive (*TP*)	False Negative (*FN*)
Bad Rating	False Positive (*FP*)	True Negative (*TN*)

5.5 McNemar's test

McNemar's test is a non-parametric test used to analyze the performance of the two methods has a statistically significant difference [50, 51]. The contingency table used to compare the two models is given in Table 8.

Where '*A*' is the number of ratings correctly predicted by Methods 1 and 2, *B* is the number of ratings correctly predicted by Method 1 but incorrectly predicted by Method 2, *C* is the number of rating correctly predicted by Method 2 but incorrectly predicted by Method 1, *D* is the number of ratings incorrectly predicted by both methods.

$$\chi^2 = \frac{(|B-C|-1)^2}{B+C} \qquad (53)$$

The Null hypothesis (H_0) for this χ^2 test is the probabilities, *Pr*(*B*) and *Pr*(*C*) are equal. The alternative hypothesis (H_1) is the performances of the two methods are not equal.

For clarity, a graphical representation is presented in Fig. 7 for the McNemer contingency table values. The *A* value is used to find out the number of ratings correctly predicted by method 1 and method 2. The *D* value is used to find out number of ratings that are incorrectly predicted by method 1 and method 2. The *B* and *C* values are used to find the best method. If *B* is

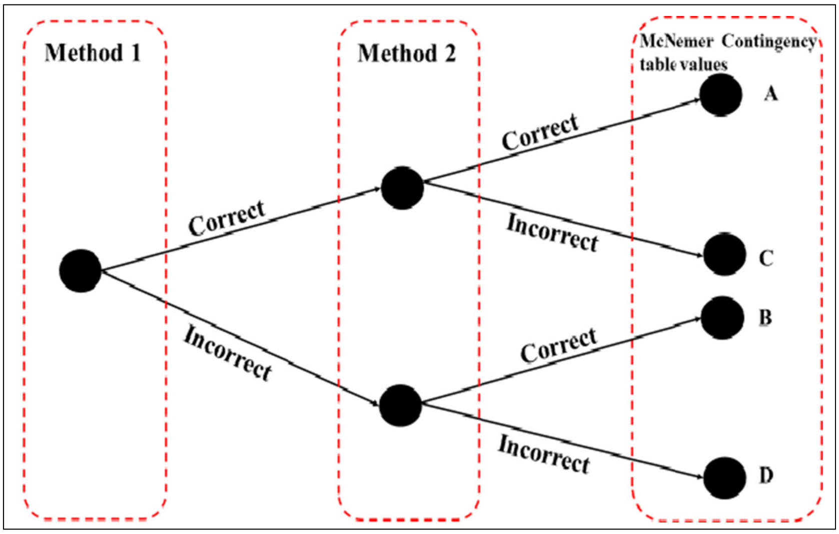
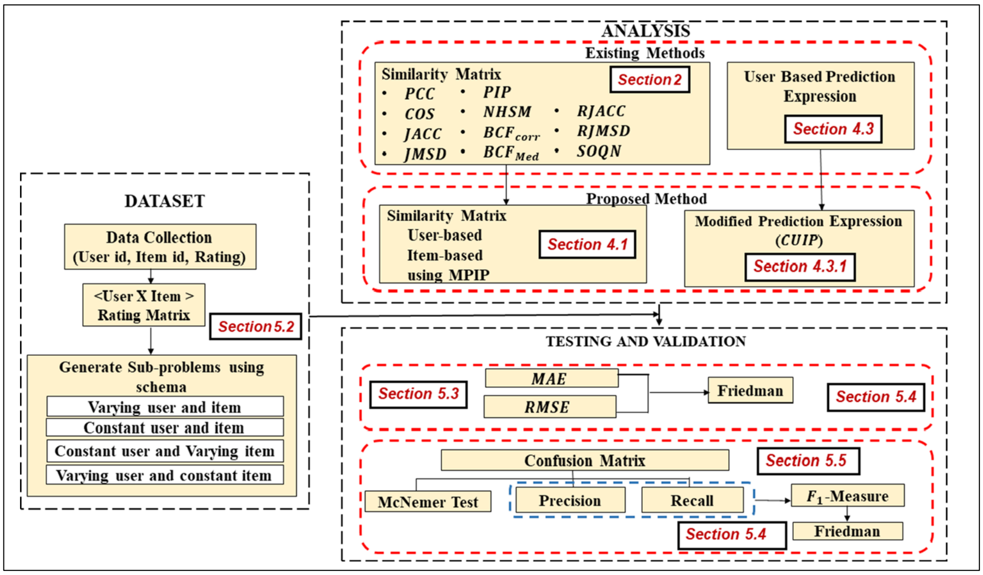


Fig. 7 Graphical Representation of McNemer table values

Fig. 8 Overall process carried out in this study



greater than C value this indicates that method 2 performs better than method 1. Similarly, if C is greater than B value then, the performance of method 1 is better than that of method 2.

In addition to McNemer test, the precision, recall, and F_1-measure are calculated. The measures are computed from the confusion matrix by converting the rating into a good or bad recommendation. The format of the confusion matrix used for computing precision, recall, and F_1 measure is given in Table 9.

Where True Positive (TP) is the number of actual good ratings provided by the user and correctly predicted by the CF-based RS, False Negative (FN) is the number of actual good ratings predicted as bad ratings, False Positive (FP) is the number of actual bad ratings predicted as good ratings, True Negative (TN) is the number of actual bad ratings correctly predicted.

The rating values are classified into two groups i.e., good and bad rating. The rating lies between R_{med} to R_{max} which are treated as good ratings; similarly, the values which are less than R_{med} are treated as bad ratings. For example, if rating scale is 1 to 5, where good rating values are 3,4, and 5, and bad ratings are 1 and 2. This similar process is carried out for various rating scale.

Precision is the ratio of the number of good ratings (i.e., relevant rating) recommendations made to the total number of recommendations.

$$precision = \frac{TP}{TP + FP} \tag{54}$$

Table 10 Comparison of minimum *MAE* values conventional and proposed method

Datasets Methods	Epinions	CiaoDVD	MovieTweet	FlimTrust	NetFlix	ML1MB	ML100KB
PCC	0.563	1.153	2.313	0.552	0.568	0.427	0.283
COS	0.663	1.276	2.515	0.571	0.545	0.430	0.285
JACC	0.615	1.252	2.466	0.515	0.524	0.416	0.275
JMSD	0.552	1.184	2.430	0.509	0.552	0.507	0.336
PIP	0.601	1.166	2.721	0.607	0.506	0.387	0.256
NHSM	0.546	1.203	2.693	0.534	0.497	0.367	0.219
BCF_{Med}	0.509	1.236	2.846	0.585	0.528	0.408	0.270
BCF_{corr}	0.530	1.215	2.650	0.472	0.564	0.394	0.261
RJACC	0.497	1.086	2.626	0.479	0.460	0.403	0.267
RJMSD	0.554	0.929	2.162	0.418	0.473	0.382	0.253
SOQN	0.464	1.012	2.595	0.413	0.454	0.330	0.243
Proposed	**0.299**	**0.509**	**1.430**	**0.201**	**0.245**	**0.172**	**0.114**

The best results are boldfaced

Table 11 Comparison of minimum *RMSE* values conventional and proposed method

Datasets Methods	Epinions	CiaoDVD	MovieTweet	FlimTrust	NetFlix	ML1MB	ML100KB
PCC	0.813	1.820	3.404	0.902	0.891	0.661	0.402
COS	0.926	1.851	3.411	0.908	0.865	0.704	0.428
JACC	0.949	1.859	3.356	0.871	0.853	0.683	0.415
JMSD	0.927	1.867	3.319	0.853	0.924	0.666	0.405
PIP	0.914	1.780	3.903	1.018	0.819	0.624	0.379
NHSM	0.841	1.994	4.032	0.896	0.761	0.571	0.311
BCF_{Med}	0.789	1.857	3.963	0.976	0.804	0.557	0.339
BCF_{corr}	0.773	1.933	3.957	0.879	0.808	0.534	0.325
RJACC	0.791	1.804	3.902	0.939	0.675	0.571	0.348
RJMSD	0.883	1.456	3.209	0.874	0.706	0.622	0.378
SOQN	0.747	1.509	3.387	0.853	0.727	0.511	0.347
Proposed	**0.432**	**0.850**	**1.845**	**0.328**	**0.347**	**0.283**	**0.172**

The best results are boldfaced

A recall is the ratio of the number of good ratings recommended to the number of actual good ratings.

$$Recall = \frac{TP}{TP + FN} \qquad (55)$$

F_1-Measure is the combination of precision and recall

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \qquad (56)$$

The higher precision, Recall and F_1-measure shows the better performance. The overall process of this comparison is shown in Fig. 8.

5.6 Results and discussion

The comparison of results obtained for the proposed framework and other methods using the performance criteria are discussed below.

The number of feasible sub-problems generated using the schema varies for each dataset. Each sub-problem, the conventional methods, and the proposed framework are adopted to predict the rating values. The *MAE* and *RMSE* values are calculated for all the sub-problems by using Eqs. (50) and (51), respectively. The set of *MAE* and *RMSE* values are generated for each dataset. From the set of *MAE* and *RMSE* values the minimum *MAE* and *RMSE* are chosen and listed in Tables 10 and 11.

Table 12 Comparison of maximum *MAE* values of the conventional and proposed method

Datasets Methods	Epinions	CiaoDVD	MovieTweet	FlimTrust	NetFlix	ML1MB	ML100KB
PCC	1.287	2.636	5.287	1.262	1.187	1.184	0.917
COS	1.484	2.917	5.750	1.304	1.234	1.276	1.013
JACC	1.334	2.861	5.638	1.178	1.198	1.234	0.955
JMSD	1.262	2.643	5.987	1.164	1.262	1.360	0.893
PIP	1.374	2.664	6.220	1.388	1.156	1.150	0.868
NHSM	1.248	2.566	6.156	1.220	1.136	1.088	0.821
BCF_{Med}	1.164	2.551	6.547	1.151	1.178	1.117	0.843
BCF_{corr}	1.094	2.712	5.585	1.055	1.164	1.079	0.708
RJACC	1.136	2.482	5.861	1.094	1.052	0.995	0.751
RJMSD	1.143	2.426	5.647	0.892	0.986	0.944	0.805
SOQN	1.060	2.260	5.793	0.945	1.038	0.982	0.741
Proposed	**0.624**	**1.062**	**2.988**	**0.453**	**0.541**	**0.494**	**0.373**

The best results are boldfaced

Table 13 Comparison of maximum *RMSE* values of the conventional and proposed method

Datasets Methods	Epinions	CiaoDVD	MovieTweet	FlimTrust	NetFlix	ML1MB	ML100KB
PCC	1.859	4.160	7.334	2.061	1.900	1.831	1.446
COS	2.283	4.231	7.797	2.076	1.977	1.949	1.540
JACC	2.024	4.249	7.671	1.991	1.949	1.893	1.495
JMSD	1.977	4.401	7.409	1.949	1.907	1.977	1.562
PIP	2.090	4.069	8.920	2.328	1.872	1.851	1.462
NHSM	1.921	4.254	8.929	2.047	1.739	1.693	1.337
BCF_{Med}	1.804	4.377	8.847	2.230	1.794	1.562	1.234
BCF_{corr}	1.767	4.314	8.833	1.949	1.630	1.538	1.037
RJACC	1.809	4.026	8.919	2.095	1.506	1.564	1.236
RJMSD	1.823	3.804	8.891	1.821	1.424	1.461	1.215
SOQN	1.707	3.369	8.849	2.010	1.501	1.515	1.197
Proposed	**0.902**	**1.776**	**4.692**	**0.725**	**0.766**	**0.812**	**0.674**

The best results are boldfaced

Our proposed method provides lesser *MAE* for all the datasets. The second best results are obtained by *SOQN* for Epinions, FlimTrust, NetFlix, and ML1MB. For CiaoDVD and MovieTweet datasets *RJMSD* attains the next better solution. For ML100KB datasets *NHSM* provides the next better result. The percentage of improvement for our proposed method from the second best solution are 35.56% for Epinions; 45.21% for CiaoDVD; 33.8% for MovieTweet; 51.33% for FlimTrust; 46.03% for NetFlix, 47.78% for ML1MB and 47.47% for ML100KB datasets. The existing similarity measures with user-based prediction expression, the high variations exist between the actual and predicted ratings. This higher variations leads to maximum *MAE* values, which decreases the prediction quality. In our proposed approach, the variation is minimal, and the predicted ratings coincide with the actual rating for most cases; this improves the effectiveness of the CF-based RS.

The table values indicate that the standard deviation of the prediction error is minimum for the proposed method compared to the other methods. The percentage of improvement of our proposed method from the next lesser *RMSE* values is 42.16%, 41.62%, 42.50%, 61.54%, 48.59%, 44.49%, and 45.35% for Epinions, CiaoDVD, MovieTweet, FlimTrust, NetFlix, ML1MB and ML100KB respectively. The comparison of maximum *MAE* and *RMSE* values for all the datasets are arranged in Tables 12 and 13.

Tables 12 and 13 show the maximum *MAE* and *RMSE* chosen for each method. It is observed that the proposed method achieves more improved solutions than the conventional methods in terms of lesser *MAE* and *RMSE* values. Average of

Table 14 Comparison of average *MAE* values of the conventional and proposed method

Datasets Methods	Epinions	CiaoDVD	MovieTweet	FlimTrust	NetFlix	ML1MB	ML100KB
PCC	0.996	2.060	4.130	0.986	0.950	0.925	0.724
COS	1.124	2.279	4.492	1.019	0.964	0.997	0.800
JACC	1.042	2.235	4.404	0.920	0.936	0.964	0.755
JMSD	0.986	2.115	4.339	0.909	0.986	1.063	0.705
PIP	1.074	2.082	4.859	1.085	0.904	0.898	0.686
NHSM	0.975	2.005	4.810	0.953	0.887	0.850	0.649
BCF_{Med}	0.909	1.993	4.744	0.975	0.942	0.947	0.666
BCF_{corr}	0.855	2.169	4.733	0.844	0.909	0.914	0.560
RJACC	0.887	1.939	4.689	0.855	0.822	0.844	0.593
RJMSD	0.893	1.895	4.412	0.697	0.789	0.800	0.636
SOQN	0.828	1.808	4.634	0.738	0.811	0.767	0.585
Proposed	**0.533**	**0.908**	**2.554**	**0.387**	**0.462**	**0.422**	**0.294**

The best results are boldfaced

Table 15 Comparison of average *RMSE* values of the conventional and proposed method

Datasets Methods	Epinions	CiaoDVD	MovieTweet	FlimTrust	NetFlix	ML1MB	ML100KB
PCC	1.452	3.250	5.730	1.611	1.484	1.431	1.130
COS	1.654	3.305	6.092	1.621	1.545	1.523	1.203
JACC	1.581	3.320	5.993	1.556	1.523	1.479	1.168
JMSD	1.545	3.334	5.927	1.523	1.490	1.545	1.221
PIP	1.632	3.179	6.969	1.819	1.463	1.446	1.142
NHSM	1.501	3.324	7.143	1.600	1.359	1.323	1.045
BCF_{Med}	1.409	3.316	7.078	1.742	1.435	1.291	1.020
BCF_{corr}	1.380	3.451	7.067	1.523	1.304	1.238	0.935
RJACC	1.413	3.221	6.968	1.676	1.205	1.326	1.048
RJMSD	1.424	2.972	6.946	1.457	1.139	1.304	1.030
SOQN	1.334	2.695	6.913	1.570	1.172	1.183	0.978
Proposed	**0.771**	**1.518**	**4.010**	**0.620**	**0.655**	**0.694**	**0.548**

The best results are boldfaced

46.71% of improved *MAE* and 45.18% of improved *RMSE* values are arrived at from the next best results. The comparison of average *MAE* and *RMSE* values are listed in Tables 14 and 15.

From Tables 14 and 15, it is noticed that among all the methods listed, the proposed method attains the smaller *MAE* and *RMSE* values for all the datasets. The proposed framework is the combination of *MPIP* similarity measure

Table 16 Comparison of percentage of improvement of average *MAE* and *RMSE* for the proposed method

Methods	Performance criteria	Epinions	CiaoDVD	MovieTweet	FlimTrust	NetFlix	ML1MB	ML100KB
PCC	*MAE*	46.49	55.92	**38.16**	60.75	51.37	54.38	59.39
	RMSE	46.90	53.29	**30.02**	61.51	55.86	51.50	51.50
COS	*MAE*	52.58	60.16	43.14	62.02	52.07	57.67	63.25
	RMSE	53.39	54.07	34.18	61.75	57.61	54.43	54.45
JACC	*MAE*	48.85	59.37	42.01	57.93	50.64	56.22	61.06
	RMSE	51.23	54.28	33.09	60.15	56.99	53.08	53.08
JMSD	*MAE*	45.94	57.07	41.14	57.43	53.14	60.30	58.30
	RMSE	50.10	54.47	32.34	59.29	56.04	55.08	55.12
PIP	*MAE*	50.37	56.39	47.44	64.33	48.89	53.01	57.14
	RMSE	52.76	52.25	42.46	65.92	55.23	52.01	52.01
NHSM	*MAE*	45.33	54.71	46.90	59.39	47.91	50.35	54.70
	RMSE	48.63	54.33	43.86	61.25	51.80	47.54	47.56
BCF_{Med}	*MAE*	41.36	54.44	46.16	60.31	50.96	55.44	55.86
	RMSE	45.28	54.22	43.35	64.41	54.36	46.24	46.27
BCF_{corr}	*MAE*	37.66	58.14	46.04	54.15	49.17	53.83	**47.50**
	RMSE	44.13	56.01	43.26	59.29	49.77	43.94	**41.39**
RJACC	*MAE*	39.91	53.17	45.53	54.74	43.80	50.00	50.42
	RMSE	45.44	52.87	42.45	63.01	45.64	47.66	47.71
RJMSD	*MAE*	40.31	52.08	42.11	**44.48**	**41.44**	47.25	53.77
	RMSE	45.86	48.92	42.27	**57.45**	**42.49**	46.78	46.80
SOQN	*MAE*	**35.63**	**49.78**	44.89	47.56	43.03	**44.98**	49.74
	RMSE	**42.20**	**43.67**	41.99	60.51	44.11	**41.34**	43.97

The minimum values are boldfaced

Table 17 Comparison of F_r values using *MAE* as reference for all the dataset

Dataset	F_r	*p*-values	Results
Epinions	229.88	2.15×10^{-43}	H_1 Accepted
CiaoDVD	233.78	3.30×10^{-44}	H_1 Accepted
MovieTweet	247.27	4.99×10^{-47}	H_1 Accepted
FlimTrust	242.57	4.82×10^{-46}	H_1 Accepted
NetFlix	270.90	5.57×10^{-52}	H_1 Accepted
ML1MB	288.15	1.32×10^{-55}	H_1 Accepted
ML100KB	288.87	9.31×10^{-56}	H_1 Accepted

and *CUIP* prediction expression. In *MPIP* all the three components are converted into to range of 0 to 1 values, this converted value helps to find a better similarity value between users or items. This goes as an input for prediction expression; in the proposed method, the item related components and the user '*j*' deviation is included; this leads to accurate rating prediction. For calculating *MAE* and *RMSE* the actual range of rating scales are used. When compared to the existing method, the predicted ratings of our proposed method are nearer to the actual ratings, which reduce the deviation between actual and predicted ratings.

The average *MAE* and *RMSE* values are computed by considering all the sub-problems. The percentage of improvement for the proposed method over other methods is calculated and listed in Table 16.

The results listed in Table 16 shows that the *SOQN* method holds the second position by obtaining lesser average *MAE* and *RMSE* for Epinions, CiaoDVD, and ML1MB datasets. Similarly, *RJMSD* provides the second best solution for FlimTrust and NetFlix datasets. *PCC* provides the next better solution for the MovieTweet dataset. BCF_{corr} provides the second best solution for the ML100KB dataset. Our proposed framework yields 53.81% and 51.60% of the average percentage of improved *MAE* and *RMSE* than the conventional similarity measures such as *PCC*, *COS*, *JACC* and *JMSD*. Similarly, the average percentage of improved *MAE* and *RMSE* for the proposed framework is 49.44% and 49.24% from specific similarity measures used for CF-based RS are *PIP*, *NHSM*, BCF_{corr}, BCF_{Med}, *RJACC*, *RJMSD*, and *SOQN*. This improved solution enhances the accuracy of the CF-based RS. The above-mentioned tables indicate that the proposed method exhibits superiority over other methods.

The Friedman rank test is conducted to test whether the performance of all the methods is the same. The *MAE* values are computed for all the sub-problems. For each sub-problem (*sp*), rank the *MAE* values from 1 (best results) to '*l*' (worst results). F_r values are calculated based on the squared average rank values of each method. Each dataset the F_r is calculated and listed in Table 17.

The calculated F_r values are higher than the χ^2 table value, which is 19.67 for the degrees of freedom 11 with a significance level of 0.05. The *p*-values obtained from the Friedman test are nearer to zero, which strongly accepts the alternate hypothesis for all the datasets. The performance of all the methods is not equal. The comparison of average rank values for the conventional method and proposed method for all datasets are shown in Table 18.

The average rank for our proposed method is 1 for all the datasets. This infers that the proposed framework provide minimum *MAE* values for all the sub-problems generated by using the schema. Similarly, a comparison of F_r values for *RMSE* is listed in Table 19.

The *p*-values of Friedman rank test in Table 18 results in acceptance of alternate hypothesis for all the datasets, and hence it is revealed that the performance of all methods is

Table 18 Comparison of average rank values obtained using *MAE* as references for all dataset

	Epinions	CiaoDVD	MovieTweet	FlimTrust	NetFlix	ML1MB	ML100KB
PCC	8.61	6.89	2.10	9.52	7.69	7.89	9.47
COS	11.64	11.49	6.30	11.39	11.49	11.68	10.68
JACC	9.64	10.56	7.39	6.69	9.68	9.57	8.96
JMSD	8.84	9.63	5.35	6.48	10.46	11.40	11.46
PIP	10.45	7.78	11.42	10.79	5.90	5.74	5.80
NHSM	6.83	5.96	10.78	7.89	4.88	4.96	6.94
BCF_{Med}	6.08	5.42	10.42	9.42	10.87	9.69	9.80
BCF_{corr}	2.99	9.69	8.55	3.88	6.64	6.88	2.01
RJACC	3.98	3.64	7.63	4.90	3.72	3.87	3.92
RJMSD	5.88	3.81	3.83	2.24	2.12	3.22	4.85
SOQN	2.05	2.11	3.21	3.80	3.55	2.10	3.11
Proposed	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

The best results are boldfaced

Table 19 Comparison of F_r values using *RMSE* as reference for all the dataset

Dataset	F_r	*p*-values	Results
Epinions	222.83	6.36×10^{-42}	H_1 Accepted
CiaoDVD	229.77	2.27×10^{-43}	H_1 Accepted
MovieTweet	237.46	5.64×10^{-45}	H_1 Accepted
FlimTrust	276.42	3.86×10^{-53}	H_1 Accepted
NetFlix	259.90	1.13×10^{-49}	H_1 Accepted
ML1MB	256.59	5.60×10^{-49}	H_1 Accepted
ML100KB	285.03	5.99×10^{-55}	H_1 Accepted

not the same. The comparison of average rank values for obtaining using *RMSE* as a reference for all the dataset are listed in Table 20.

The rank values show that our proposed method has proved better than the existing methods. Compared to other methods, the proposed framework attains top rank for all the sub-problems. The proposed method yields a better solution for sparsity problems than the existing similarity measures with user-based prediction expression.

To further validate the proposed method, a McNemer test is conducted on the confusion matrix. The maximum and minimum *MAE* are chosen from the conventional methods for each dataset, and the corresponding confusion matrix is tested through a McNemer test. The proposed framework is treated as reference for conducting this test. The corresponding calculated χ^2 values for all the datasets are listed in Table 21.

The calculated χ^2 values are higher than the χ^2 table value; this results in acceptance of alternate hypothesis for McNemer test. This shows that performance of existing method and proposed method are not equal. The values of *B* and *C* are the primary components required for conducting McNemer test. The McNemer test is conducted by considering our proposed framework as a reference. The *B* values are related to the number of rating incorrectly predicted by our proposed framework but correctly predicted by the existing method. Similarly, *C* is related to the number of rating correctly predicted by our proposed framework but incorrectly predicted by the existing method. The comparison of *B* and *C* values are listed in Table 22.

The comparison table (Table 22) results clearly show that *C* values are higher than the *B* values i.e., the number of ratings predicted by our proposed framework is higher than the other exiting methods.

The maximum *MAE* values are chosen from the feasible number of sub-problems generated for each dataset corresponding predicted values are considered for conducting the McNemer test by using a confusion matrix. The calculated χ^2 and *p*-values are listed in Table 23.

The calculated χ^2 value is greater than the table value, which agrees with the alternate hypothesis, i.e., a significance difference exists between the conventional and proposed methods. The proposed method has proved that it is better than the existing method for all the datasets. Table 24 is the comparison of *B* and *C* values for maximum *MAE*.

The *B* and *C* values comparison for a maximum of *MAE* values reveal that the *C* is greater than the *B* values for all the datasets. When compared to the existing method, our proposed method correctly predicts good and bad ratings.

The precision, recall, and F_1-measure are calculated for each sub-problem. The summary of average precision, recall, and F_1-measure for the conventional and proposed methods are reported in Tables 25, 26, 27.

The comparison table (Table 25) indicates that the average precision value is higher for our proposed framework. These results show that from the total predicted ratings most of the

Table 20 Comparison of average rank values obtained using *RMSE* as references for all dataset

	Epinions	CiaoDVD	MovieTweet	FlimTrust	NetFlix	ML1MB	ML100KB
PCC	6.78	5.98	2.10	7.68	8.87	7.86	9.18
COS	11.04	7.69	5.26	9.75	11.20	11.37	8.84
JACC	9.86	8.55	7.29	4.79	10.68	9.98	10.65
JMSD	9.48	10.41	6.43	3.41	10.45	10.78	11.45
PIP	10.48	3.78	8.64	11.27	7.86	8.55	6.93
NHSM	7.85	9.87	11.51	6.87	5.88	5.88	5.94
BCF_{Med}	4.68	8.28	10.80	11.41	7.88	4.82	9.89
BCF_{corr}	2.89	11.49	9.67	2.93	4.84	2.85	2.04
RJACC	4.95	4.94	7.86	9.89	3.86	6.89	3.68
RJMSD	6.89	3.89	3.50	2.18	2.13	5.85	5.30
SOQN	2.08	2.06	3.87	6.78	3.30	2.08	3.07
Proposed	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

The best results are boldfaced

Table 21 The calculated χ^2 value for minimum *MAE* of the proposed method against the conventional method

	Epinions	CiaoDVD	MovieTweet	FlimTrust	NetFlix	ML1MB	ML100KB
PCC	54.80** (6.7×10^{-14})	49.52** (1.0×10^{-12})	32.01** (7.8×10^{-09})	74.74** (2.7×10^{-18})	69.49** (3.8×10^{-17})	41.38** (6.3×10^{-11})	30.69** (1.5×10^{-08})
COS	51.86** (3.0×10^{-13})	55.00** (6.1×10^{-14})	43.45** (2.2×10^{-11})	55.47** (4.8×10^{-14})	37.01** (6.0×10^{-10})	59.48** (6.2×10^{-15})	34.02** (2.8×10^{-09})
JACC	47.34** (3.0×10^{-12})	57.45** (1.7×10^{-14})	42.12** (4.3×10^{-11})	56.79** (2.4×10^{-14})	47.18** (3.3×10^{-12})	68.43** (6.6×10^{-17})	51.20** (4.2×10^{-13})
JMSD	34.27** (2.4×10^{-09})	53.85** (1.1×10^{-13})	52.92** (1.7×10^{-13})	58.07** (1.2×10^{-14})	49.01** (1.2×10^{-12})	57.77** (1.4×10^{-14})	61.79** (1.9×10^{-15})
PIP	42.35** (3.8×10^{-11})	47.92** (2.2×10^{-12})	45.79** (6.7×10^{-12})	55.80** (4.0×10^{-14})	32.14** (7.3×10^{-09})	53.02** (1.7×10^{-13})	55.65** (4.4×10^{-14})
NHSM	41.15** (7.2×10^{-11})	44.37** (1.3×10^{-11})	57.76** (1.5×10^{-14})	65.82** (2.5×10^{-16})	59.18** (7.2×10^{-15})	67.08** (1.3×10^{-16})	27.90** (6.5×10^{-08})
BCF_{Med}	43.32** (2.3×10^{-11})	31.45** (1.0×10^{-08})	49.46** (1.0×10^{-12})	55.62** (4.4×10^{-14})	38.98** (2.1×10^{-10})	63.82** (6.9×10^{-16})	38.77** (2.4×10^{-10})
BCF_{corr}	42.77** (3.1×10^{-11})	23.21** (7.5×10^{-07})	52.83** (1.8×10^{-13})	59.25** (7.0×10^{-15})	60.89** (3.0×10^{-15})	41.92** (4.8×10^{-11})	48.01** (2.1×10^{-12})
RJACC	40.15** (1.2×10^{-10})	28.93** (3.8×10^{-08})	42.55** (3.5×10^{-11})	40.69** (9.1×10^{-11})	55.75** (4.1×10^{-14})	63.69** (7.3×10^{-16})	47.68** (2.5×10^{-12})
RJMSD	35.57** (1.2×10^{-09})	22.81** (9.2×10^{-07})	39.45** (1.7×10^{-10})	40.52** (9.9×10^{-11})	40.37** (1.0×10^{-10})	49.45** (1.0×10^{-12})	35.50** (1.3×10^{-09})
SOQN	23.89** (5.2×10^{-07})	28.62** (4.5×10^{-08})	37.23** (5.3×10^{-10})	30.17** (2.0×10^{-08})	36.15** (9.3×10^{-10})	45.45** (7.9×10^{-12})	39.26** (1.9×10^{-10})

** significance level 5%; ** significance level 1%; () denotes p-values*

actual good rating items are correctly predicted by our proposed method.

The results arranged in Table 26 shows that our proposed method values are greater than the other methods. This

Table 22 Comparison of *B* and *C* values for minimum *MAE* of the conventional and proposed method

		Epinions	CiaoDVD	MovieTweet	FlimTrust	NetFlix	ML1MB	ML100KB
PCC	*B*	105	52	122	62	250	178	20
	C	**244**	**154**	**229**	**204**	**476**	**323**	**75**
COS	*B*	82	56	134	66	276	167	25
	C	**205**	**168**	**267**	**185**	**439**	**342**	**88**
JACC	*B*	102	47	120	78	323	166	22
	C	**228**	**156**	**245**	**206**	**524**	**356**	**103**
JMSD	*B*	121	41	119	80	377	172	24
	C	**232**	**141**	**262**	**211**	**596**	**346**	**119**
PIP	*B*	119	39	101	77	378	161	17
	C	**244**	**130**	**224**	**203**	**552**	**322**	**98**
NHSM	*B*	105	37	109	72	328	151	34
	C	**222**	**122**	**255**	**209**	**558**	**332**	**95**
BCF_{Med}	*B*	99	39	114	74	359	168	29
	C	**217**	**108**	**249**	**198**	**548**	**351**	**101**
BCF_{corr}	*B*	101	43	115	61	321	169	25
	C	**219**	**102**	**256**	**182**	**552**	**312**	**105**
RJACC	*B*	102	40	128	49	224	132	29
	C	**216**	**106**	**257**	**137**	**414**	**299**	**112**
RJMSD	*B*	112	31	122	43	263	152	35
	C	**222**	**83**	**243**	**127**	**431**	**303**	**107**
SOQN	*B*	124	34	117	37	218	172	41
	C	**215**	**96**	**232**	**103**	**364**	**323**	**122**

The best results are boldfaced

Table 23 The calculated χ^2 value for maximum *MAE* of the proposed method against the conventional method

	Epinions	CiaoDVD	MovieTweet	FlimTrust	NetFlix	ML1MB	ML100KB
PCC	63.14** (9.7×10^{-16})	49.28** (1.1×10^{-12})	30.15** (2.0×10^{-08})	61.75** (1.9×10^{-15})	87.78** (3.6×10^{-21})	47.30** (3.1×10^{-12})	42.68** (3.2×10^{-11})
COS	59.76** (5.4×10^{-15})	48.85** (1.4×10^{-12})	43.91** (1.7×10^{-11})	60.56** (3.6×10^{-15})	89.74** (1.3×10^{-21})	54.04** (1.0×10^{-13})	43.96** (1.7×10^{-11})
JACC	54.56** (7.6×10^{-14})	37.02** (5.9×10^{-10})	43.66** (2.0×10^{-11})	62.84** (1.1×10^{-15})	57.38** (1.8×10^{-14})	57.65** (1.5×10^{-14})	27.18** (9.5×10^{-08})
JMSD	39.51** (1.6×10^{-10})	33.26** (4.1×10^{-09})	58.57** (9.9×10^{-15})	55.61** (4.4×10^{-14})	57.92** (1.3×10^{-14})	43.98** (1.6×10^{-11})	36.67** (7.1×10^{-10})
PIP	48.81** (1.4×10^{-12})	33.78** (3.1×10^{-09})	51.00** (4.6×10^{-13})	49.96** (8.0×10^{-13})	33.64** (3.4×10^{-09})	39.93** (1.3×10^{-10})	43.38** (2.2×10^{-11})
NHSM	47.42** (2.9×10^{-12})	35.41** (1.3×10^{-09})	66.76** (1.5×10^{-16})	49.11** (1.2×10^{-12})	55.39** (5.0×10^{-14})	58.48** (1.0×10^{-14})	22.48** (1.1×10^{-06})
BCF_{Med}	49.92** (8.1×10^{-13})	34.11** (2.6×10^{-09})	54.41** (8.2×10^{-14})	26.56** (1.3×10^{-07})	43.94** (1.7×10^{-11})	51.73** (3.2×10^{-13})	28.96** (3.8×10^{-08})
BCF_{corr}	49.30** (1.1×10^{-12})	26.46** (1.3×10^{-07})	58.97** (8.1×10^{-15})	21.79** (1.5×10^{-06})	52.84** (1.8×10^{-13})	26.61** (1.2×10^{-07})	31.90** (8.3×10^{-09})
RJACC	46.28** (5.2×10^{-12})	31.36** (1.1×10^{-08})	43.35** (2.3×10^{-11})	34.40** (2.3×10^{-09})	38.79** (2.4×10^{-10})	57.52** (1.7×10^{-14})	40.96** (7.9×10^{-11})
RJMSD	41.06** (7.7×10^{-11})	27.33** (8.8×10^{-08})	39.86** (1.3×10^{-10})	26.95** (1.0×10^{-07})	39.86** (1.3×10^{-10})	37.03** (5.9×10^{-10})	26.57** (1.3×10^{-07})
SOQN	27.55** (7.8×10^{-08})	28.23** (5.5×10^{-08})	37.459** (4.7×10^{-10})	25.57** (2.2×10^{-07})	36.78** (6.7×10^{-10})	30.07** (2.1×10^{-08})	31.21** (1.1×10^{-08})

** significance level 5%; ** significance level 1%; () denotes p-values*

Table 24 Comparison of *B* and *C* values for maximum *MAE* of the conventional and proposed method

		Epinions	CiaoDVD	MovieTweet	FlimTrust	NetFlix	ML1MB	ML100KB
PCC	*B*	120	78	248	62	443	397	51
	C	**281**	**195**	**387**	**187**	**770**	**617**	**143**
COS	*B*	94	83	272	66	462	372	53
	C	**236**	**202**	**451**	**192**	**799**	**603**	**148**
JACC	*B*	117	89	244	77	649	370	61
	C	**262**	**192**	**414**	**213**	**953**	**609**	**135**
JMSD	*B*	139	83	242	81	758	384	63
	C	**267**	**177**	**443**	**209**	**1086**	**592**	**153**
PIP	*B*	137	81	205	78	760	359	56
	C	**281**	**175**	**379**	**196**	**1005**	**551**	**152**
NHSM	*B*	121	73	221	73	659	337	54
	C	**255**	**166**	**431**	**187**	**960**	**568**	**117**
BCF_{Med}	*B*	114	69	231	75	722	375	53
	C	**250**	**158**	**421**	**154**	**997**	**600**	**126**
BCF_{corr}	*B*	116	73	233	69	644	377	43
	C	**252**	**151**	**433**	**137**	**934**	**534**	**115**
RJACC	*B*	117	70	260	66	451	295	32
	C	**248**	**155**	**434**	**154**	**660**	**511**	**109**
RJMSD	*B*	129	79	248	71	528	339	39
	C	**255**	**161**	**411**	**149**	**755**	**518**	**101**
SOQN	*B*	143	66	238	82	440	384	43
	C	**247**	**144**	**392**	**162**	**641**	**552**	**114**

The best results are boldfaced

Table 25 Comparison of average precision values for the conventional and proposed method

Datasets Methods	Epinions	CiaoDVD	MovieTweet	FlimTrust	NetFlix	ML1MB	ML100KB
PCC	0.727	0.685	0.715	0.719	0.721	0.711	0.729
COS	0.725	0.712	0.662	0.717	0.731	0.726	0.747
JACC	0.740	0.722	0.645	0.721	0.729	0.750	0.759
JMSD	0.746	0.732	0.653	0.726	0.747	0.757	0.776
PIP	0.773	0.729	0.671	0.758	0.701	0.761	0.774
NHSM	0.763	0.716	0.681	0.748	0.767	0.750	0.784
BCF_{Med}	0.772	0.721	0.684	0.755	0.774	0.737	0.755
BCF_{corr}	0.785	0.735	0.698	0.755	0.782	0.754	0.773
RJACC	0.791	0.747	0.703	0.768	0.798	0.758	0.775
RJMSD	0.795	0.749	0.710	0.773	0.799	0.764	0.766
SOQN	0.791	0.752	0.650	0.785	0.795	0.772	0.783
Proposed	**0.839**	**0.828**	**0.810**	**0.841**	**0.837**	**0.867**	**0.881**

The best results are boldfaced

indicates more number of actual good rating items are correctly predicted by our proposed method. Table 27 is the comparison of average F_1 measure values for the conventional and proposed method.

The F_1 measure includes both the precision and recall values. The proposed method attains better results for all the datasets. To validate the effectiveness of the proposed method the Friedman rank test is conducted on F_1 -Measure values. The F_r values are listed in Table 28.

The calculated $F_r$values confirm that the performance of all the methods is not equal. Also, the average rank using for F_r calculation is shown in Table 29.

Table 29 indicate that the proposed method provides higher F_1-measure for all the feasible sub-problems. Furthermore, most of the actual good rating items are correctly recommended by the proposed method.

The main issue in CF-based RS is sparsity problem. A new framework is introduced to solve the sparsity problem and to enhance the prediction performance. The experiments are conducted on various level of sparse data. The *MAE* and *RMSE* results for our proposed approach is minimal compared to conventional similarity measures like *PCC*, *COS*, *JACC* and *JMSD* with user based prediction expression. The specific similarity methods used for CF-based RS are *PIP*, *NHSM*, BCF_{corr}, BCF_{Med}, *RJACC*, *RJMSD*, *SOQN* with user based prediction expression also attains higher *MAE* and *RMSE* values than the proposed methods for all the datasets. This clearly shows that the predicted ratings by the proposed

Table 26 Comparison of average recall values for the conventional and proposed method

Datasets Methods	Epinions	CiaoDVD	MovieTweet	FlimTrust	NetFlix	ML1MB	ML100KB
PCC	0.737	0.710	0.785	0.717	0.719	0.723	0.741
COS	0.736	0.725	0.720	0.724	0.714	0.743	0.765
JACC	0.746	0.729	0.722	0.715	0.725	0.739	0.748
JMSD	0.763	0.724	0.726	0.735	0.710	0.761	0.780
PIP	0.764	0.749	0.745	0.743	0.751	0.769	0.782
NHSM	0.776	0.718	0.736	0.718	0.759	0.741	0.795
BCF_{Med}	0.790	0.731	0.743	0.719	0.755	0.743	0.762
BCF_{corr}	0.788	0.730	0.759	0.727	0.776	0.756	0.775
RJACC	0.793	0.755	0.767	0.754	0.782	0.763	0.784
RJMSD	0.804	0.753	0.763	0.733	0.790	0.780	0.757
SOQN	0.802	0.759	0.713	0.745	0.791	0.782	0.800
Proposed	**0.842**	**0.833**	**0.814**	**0.819**	**0.845**	**0.856**	**0.871**

The best results are boldfaced

Table 27 Comparison of average F_1 measure for the conventional and proposed method

Datasets Methods	Epinions	CiaoDVD	MovieTweet	FlimTrust	NetFlix	ML1MB	ML100KB
PCC	0.730	0.696	0.745	0.715	0.716	0.719	0.737
COS	0.729	0.714	0.696	0.722	0.719	0.738	0.759
JACC	0.744	0.725	0.685	0.718	0.725	0.741	0.750
JMSD	0.758	0.727	0.685	0.729	0.717	0.758	0.777
PIP	0.763	0.735	0.707	0.745	0.711	0.763	0.776
NHSM	0.764	0.715	0.708	0.725	0.754	0.744	0.786
BCF_{Med}	0.777	0.729	0.716	0.738	0.760	0.740	0.759
BCF_{corr}	0.778	0.736	0.731	0.740	0.767	0.753	0.772
RJACC	0.792	0.750	0.737	0.742	0.790	0.752	0.774
RJMSD	0.796	0.744	0.739	0.757	0.791	0.770	0.760
SOQN	0.798	0.757	0.697	0.764	0.789	0.774	0.789
Proposed	**0.840**	**0.829**	**0.813**	**0.826**	**0.843**	**0.859**	**0.875**

The best results are boldfaced

Table 28 Comparison of F_r values using F_1 as a reference for all the dataset

Dataset	F_r	*p*-values	Results
Epinions	233.02	4.77×10^{-44}	H_1 Accepted
CiaoDVD	227.08	8.27×10^{-43}	H_1 Accepted
MovieTweet	239.69	1.92×10^{-45}	H_1 Accepted
FlimTrust	245.15	1.39×10^{-46}	H_1 Accepted
NetFlix	243.60	2.92×10^{-46}	H_1 Accepted
ML1MB	227.71	6.11×10^{-43}	H_1 Accepted
ML100KB	235.13	1.72×10^{-44}	H_1 Accepted

framework are closer to the actual ratings and this minimizes the error values. The average ranking value for *MAE* and *RMSE* of our proposed framework is 1, which denotes that for various sub-problems, our proposed method provides

Table 29 Comparison of average rank values obtained using F_1 as references for all dataset

	Epinions	CiaoDVD	MovieTweet	FlimTrust	NetFlix	ML1MB	ML100KB
PCC	10.84	11.57	2.53	11.76	10.78	11.34	11.56
COS	11.86	11.03	9.93	9.80	8.82	10.80	10.84
JACC	9.88	9.14	10.80	10.82	7.84	9.12	8.95
JMSD	8.94	7.79	11.56	7.93	9.80	4.95	4.99
PIP	8.09	5.78	7.84	3.95	11.76	3.92	5.14
NHSM	6.78	9.54	6.57	8.81	6.86	7.85	2.87
BCF_{Med}	5.88	6.83	6.73	6.92	5.92	8.44	8.89
BCF_{corr}	4.98	4.85	4.98	5.88	4.90	6.66	6.51
RJACC	3.92	3.26	3.95	4.93	2.94	6.98	5.86
RJMSD	3.26	2.98	3.22	2.48	3.46	4.00	7.75
SOQN	2.56	4.24	8.89	3.72	3.92	2.94	3.64
Proposed	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

The best results are boldfaced

better prediction rating. The McNemer test results in acceptance of alternate hypothesis, which shows that existing and proposed methods performance are not same. Each conventional method is tested with our proposed method and in all the comparison, our proposed approach offers improved solution in terms of accurate prediction. Finally, the Precision, Recall, and F_1-measure values for our proposed approach are above 0.8, higher than the other methods. The results obtained from the analysis indicate that our proposed framework improves the CF-based RS performance by correctly predicting more good ratings items and reducing the misclassification error. This improved prediction helps the company to identify and recommend products that are more relevant to online users.

6 Conclusion

CF-based RS depends on the similarity measures, of which *PIP* is one of the most popular techniques for calculating the similarity between the users. However, the ranges of proximity-impact-popularity values are wide and each component provides a different weight in different scenarios; i.e., the magnitude of each component differs. This is a serious limitation of existing *PIP* measures; therefore, we have developed a modified *PIP* similarity measure that provides a common value range between 0 and 1 for all the three components, resulting in equal priority. We have also developed a modified prediction expression to include the item-based average and weighted average deviation with user-based average and weighted average deviation in the expression to improve accuracy. Finally, a procedure for forecasting unrated items has been introduced to improve the recommendation performance. The proposed framework was tested by using various benchmark datasets, namely ML1MB, Netflix, ML100KB, CiaoDVD, Epinions, MovieTweet, and FlimTrust. The entire analysis was conducted for a different level of sparsity subproblems generated from the user-item rating matrix. The various level of sparse data is created by varying the number of users, items, and ratings. This is done for all datasets. The results obtained from the proposed method are compared with those of conventional methods. The proposed framework provides better results for all datasets yielding lower *MAE* and *RMSE* than the existing methods. The McNemer test is conducted to validate the proposed method in terms of good and bad rating recommendations. This analysis results in the acceptance of the alternate hypothesis (i.e., the performances of the conventional and proposed methods are not equal). Also, precision, recall, and F_1-measure are calculated to identify the best method based on the confusion matrix. The proposed method attains higher precision, recall, and F_1-measure. Finally, a Friedman rank test was conducted on the *MAE*, *RMSE*, and F_1 values. The statistical results show that our proposed framework outperformed the other methods.

References

1. Baker T, Mackay M, Randles M, Taleb-Bendiab A (2013) Intention-oriented programming support for runtime adaptive autonomic cloud-based applications. Comput Electr Eng 39:2400–2412. https://doi.org/10.1016/j.compeleceng.2013.04.019
2. Karam Y, Baker T, Taleb-Bendiab A (2012) Intention-oriented modelling support for socio-technical driven elastic cloud applications. In: 2012 international conference on innovations in information technology, IIT 2012
3. Baker T, Taleb-Bendiab A, Randles M (2009) Auditable intention-oriented web applications using PAA auditing/accounting paradigm. Front Artif Intell Appl. https://doi.org/10.3233/978-1-60750-052-0-61
4. Ozsoy MG, Polat F, Alhajj R (2016) Making recommendations by integrating information from multiple social networks. Appl Intell 45:1047–1065. https://doi.org/10.1007/s10489-016-0803-1
5. Zhang J, Lin Y, Lin M, Liu J (2016) An effective collaborative filtering algorithm based on user preference clustering. Appl Intell 45:230–240. https://doi.org/10.1007/s10489-015-0756-9
6. Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey. Knowledge-Based Syst. 46:109–132. https://doi.org/10.1016/j.knosys.2013.03.012
7. Jiang S, Fang SC, An Q, Lavery JE (2019) A sub-one quasi-norm-based similarity measure for collaborative filtering in recommender systems. Inf Sci (Ny) 487:142–155. https://doi.org/10.1016/j.ins.2019.03.011
8. Hwangbo H, Kim YS, Cha KJ (2018) Recommendation system development for fashion retail e-commerce. Electron Commer Res Appl 28:94–101. https://doi.org/10.1016/j.elerap.2018.01.012
9. Shi X, Luo X, Shang M, Gu L (2017) Long-term performance of collaborative filtering based recommenders in temporally evolving systems. Neurocomputing. 267:635–643. https://doi.org/10.1016/j.neucom.2017.06.026
10. Li Y, Lu L, Xuefeng L (2004) A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-commerce. Expert Syst Appl 28:67–77. https://doi.org/10.1016/j.eswa.2004.08.013
11. Cohen WW, Fan W (2000) Web-collaborative filtering: recommending music by crawling the web. Comput Netw 33:685–698. https://doi.org/10.1016/S1389-1286(00)00057-8
12. Wang Y, Deng J, Gao J, Zhang P (2017) A hybrid user similarity model for collaborative filtering. Inf Sci (Ny). 418-419:102–118. https://doi.org/10.1016/j.ins.2017.08.008
13. Bellogín A, Sánchez P (2017) Collaborative filtering based on subsequence matching: a new approach. Inf Sci (Ny). 418-419:432–446. https://doi.org/10.1016/j.ins.2017.08.016
14. Resnick P, Iacovou N, Suchak M, et al (1994) GroupLens: An open architecture for collaborative filtering of Netnews. In: CSCW
15. Shardanand U, Maes P (1995) Social information filtering: algorithms for automating ``word of mouth”. In: CHI ‘95: proceedings of the SIGCHI conference on human factors in computing systems
16. Bobadilla J, Hernando A, Ortega F, Gutiérrez A (2012) Collaborative filtering based on significances. Inf Sci (Ny) 185:1–17. https://doi.org/10.1016/j.ins.2011.09.014
17. Patra BK, Launonen R, Ollikainen V, Nandi S (2015) A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data. Knowledge-Based Syst. 82:163–177. https://doi.org/10.1016/j.knosys.2015.03.001

18. Ahn HJ (2008) A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Inf Sci (Ny) 178: 37–51. https://doi.org/10.1016/j.ins.2007.07.024
19. Basu C, Hirsh H, Cohen W (1998) Recommendation as classification: using social and content-based information in recommendation. In Recommender Systems. Papers from 1998 Workshop. Technical Report WS-98-08. AAAI Press
20. Krulwich B, Burkey C (1996) Learning user information interests through the extraction of semantically significant phrases. In: Proceedings of the AAAI spring symposium on machine learning in information access
21. Lang K (1995) NewsWeeder : Learning to Filter Netnews (To appear in ML 95). Proc 12th Int Mach Learn Conf
22. Sheugh L, Alizadeh SH (2015) A note on Pearson correlation coefficient as a metric of similarity in recommender system. In: 2015 AI and robotics, IRANOPEN 2015 - 5th conference on artificial intelligence and robotics
23. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. Proc 10th …. https://doi.org/10.1145/371920.372071
24. Huang Z, Chen H, Zeng D (2004) Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. ACM Trans Inf Syst 22:116–142. https://doi.org/10.1145/963770.963775
25. Papagelis M, Plexousakis D, Kutsuras T (2005) Alleviating the Sparsity problem of collaborative filtering using trust inferences
26. Liu H, Hu Z, Mian A, Tian H, Zhu X (2014) A new user similarity model to improve the accuracy of collaborative filtering. Knowledge-Based Syst 56:156–166. https://doi.org/10.1016/j.knosys.2013.11.006
27. Kurdija AS, Silic M, Vladimir K, Delac G (2018) Efficient global correlation measures for a collaborative filtering dataset. Knowledge-Based Syst. 147:36–42. https://doi.org/10.1016/j.knosys.2018.02.013
28. Yu K, Schwaighofer A, Tresp V et al (2004) Probabilistic memory-based collaborative filtering. IEEE Trans Knowl Data Eng 16:56–69. https://doi.org/10.1109/TKDE.2004.1264822
29. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 17:734–749
30. Su X, Khoshgoftaar TM (2009) A survey of collaborative filtering techniques. Adv Artif Intell 2009:1–19. https://doi.org/10.1155/2009/421425
31. Bag S, Kumar SK, Tiwari MK (2019) An efficient recommendation generation using relevant Jaccard similarity. Inf Sci (Ny) 483:53–64. https://doi.org/10.1016/j.ins.2019.01.023
32. Guo G, Zhang J, Yorke-Smith N (2013) A novel bayesian similarity measure for recommender systems. In: IJCAI International Joint Conference on Artificial Intelligence
33. Luo X, Xia Y, Zhu Q (2012) Incremental collaborative filtering recommender based on regularized matrix factorization. Knowledge-Based Syst 27:271–280. https://doi.org/10.1016/j.knosys.2011.09.006
34. Bobadilla J, Serradilla F (2009) The effect of Sparsity on collaborative filtering metrics. In: Conferences in Research and Practice in Information Technology Series
35. Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst 22:5–53. https://doi.org/10.1145/963770.963772
36. Tan Z, He L (2017) An efficient similarity measure for user-based collaborative filtering recommender systems inspired by the physical resonance principle. IEEE Access 5:27211–27228. https://doi.org/10.1109/ACCESS.2017.2778424
37. Abualigah LMQ, Khader AT (2017) Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. J Supercomput 73:4773–4795. https://doi.org/10.1007/s11227-017-2046-2
38. Abualigah L. M. Q (2019) Feature selection and enhanced krill herd algorithm for text document Clusteringitle, studies in. Springer International Publishing, Switzerland AG
39. Abualigah LMQ, Hanandeh SE (2015) Applying genetic algorithms to information retrieval using vector space model. Int J Comput Sci Eng Appl 5:19–28. https://doi.org/10.5121/ijcsea.2015.5102
40. Saranya KG, Sudha Sadasivam G (2017) Modified heuristic similarity measure for personalization using collaborative filtering technique. Appl Math Inf Sci 11:307–315. https://doi.org/10.18576/amis/110137
41. Konstan JA, Miller BN, Maltz D, Herlocker JL, Gordon LR, Riedl J (1997) GroupLens: applying collaborative filtering to Usenet news. Commun ACM 40:77–87. https://doi.org/10.1145/245108.245126
42. Sarwar BM, Karypis G, Konstan J, Riedl J (2002) Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In Proceedings of the fifth international conference on computer and information technology (Vol. 1, pp. 291–324)
43. Jamalzehi S, Menhaj MB (2016) A new similarity measure based on item proximity and closeness for collaborative filtering recommendation. In: 2016 4th international conference on control, instrumentation, and automation, ICCIA 2016
44. Ma H, King I, Lyu MR (2008) Effective missing data prediction for collaborative filtering
45. Chen Y, Wu C, Xie M, Guo X (2011) Solving the sparsity problem in recommender systems using association retrieval. J Comput https://doi.org/10.4304/jcp.6.9.1896-1902, 6
46. Singh S, Bag S, Jenamani M (2016) Relative similarity based approach for improving aggregate recommendation diversity. In: 12th IEEE international conference electronics, energy, environment, communication, computer, control: (E3-C3), INDICON 2015
47. Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32: 675–701. https://doi.org/10.1080/01621459.1937.10503522
48. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1:3–18. https://doi.org/10.1016/j.swevo.2011.02.002
49. Luo J, Liu Z (2020) Novel grey wolf optimization based on modified differential evolution for numerical function optimization. Appl Intell 50:468–486. https://doi.org/10.1007/s10489-019-01521-5
50. Roggo Y, Duponchel L, Huvenne JP (2003) Comparison of supervised pattern recognition methods with McNemar's statistical test: application to qualitative analysis of sugar beet by near-infrared spectroscopy. Anal Chim Acta 477:187–200. https://doi.org/10.1016/S0003-2670(02)01422-8
51. Roggo Y, Duponchel L, Ruckebusch C, Huvenne JP (2003) Statistical tests for comparison of quantitative and qualitative models developed with near infrared spectral data. J Mol Struct 654:253–262. https://doi.org/10.1016/S0022-2860(03)00248-5

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.