IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 9, 2020, accepted June 22, 2020, date of publication June 30, 2020, date of current version July 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3005953

An Improved Product Recommendation Method for Collaborative Filtering

ARTA IFTIKHAR[1], MUSTANSAR ALI GHAZANFAR[1], MUBBASHIR AYUB[1], ZAHID MEHMOOD[2], AND MUAZZAM MAQSOOD[3]

[1]Department of Software Engineering, University of Engineering and Technology Taxila, Taxila 47050, Pakistan
[2]Department of Computer Engineering, University of Engineering and Technology Taxila, Taxila 47050, Pakistan
[3]Department of Computer Science, COMSATS University Islamabad, Attock Campus, Attock 43600, Pakistan

Corresponding authors: Mubbashir Ayub (mubbashir.ayub@uettaxila.edu.pk) and Arta Iftikhar (arta.iftikhar@uettaxila.edu.pk)

ABSTRACT Collaborative filtering (CF) is the most commonly used technique for online recommendations. CF works by computing the interests of a user by gathering preferences or taste information of other users. In this technique, similar users or items are discovered by exploring the user-item rating matrix. Based on the computed similarity, a prediction is made for the unknown or new product. There are many similarity computation methods, such as the Pearson correlation coefficient (PCC), Jaccard, Mean square difference, Cosine, etc. however, the accuracy of product recommendations using these methods is not very promising. This work introduces an improved product recommendation method for collaborative filtering, which is based on the triangle similarity. However, the downside of triangle similarity is that it only considers the common ratings of users. The proposed similarity measure not only focuses on common ratings but also consider the ratings of those items that are not commonly rated from pairs of users. The obtained similarity is further complemented with the user rating preference (URP) behavior in giving rating preferences. To evaluate the accuracy, experiments are performed on the six commonly used datasets in the field of CF. Experimental results prove that the proposed similarity measure performs well as compared to the existing similarity measures.

INDEX TERMS Collaborative filtering, recommender systems, triangle similarity, user preferences.

I. INTRODUCTION

The modern world is relying significantly on the world wide web (WWW) that has millions of pages carrying an incredible amount of information. Progress in technology and availability of gadgets (such as smartphones, laptops, and tablets) allows people to spend time in search of objects of their interests. Information overload, as a result of these developments, gets problematic among users to reach a point of relevance [1], [2]. The randomness of this large amount of data, frustrate the users to choose what they want. These issues might be resolved if the most relevant recommendations are made to the users based on their preferences. Recommender systems do this job by analyzing the already provided data on the web by the user. These systems then recommend something the user might prefer to check out. By using such systems, both the service provider and user will be benefited mutually. A service provider would attain recognition of his products and possibly increased sales. On the other hand, the user's time would be saved in discovering products and services of his interest [3].

Collaborative filtering (CF) is one of the most widely used technique in recommender systems, owing to its simplicity and accuracy [4]–[7]. It is a personalization technique that counts on the set of ratings the user has provided for certain products and services [8]. These ratings can be achieved based on feedback provided by the user. One way to get this feedback is implicit feedback in which a user clicks on a particular object. Another way is the explicit behavior of a user, where user rates a particular object in terms of numbered or starred values. CF uses two techniques for a recommendation. One is a model-based technique, which uses attributes of items/users [9]. The second technique is memory-based, in which the similarity between the active user and other users in the user-item rating matrix is calculated. Based upon computed similarity, users are categorized as neighbor users of the active user, after which the recommendation is made according to the preferences of the

The associate editor coordinating the review of this manuscript and approving it for publication was Tomasz Trzcinski.

active user [10], [11]. CF requires a lot of computational analysis for calculating the similarity between users. The user ratings are recorded and a comparison is applied to those ratings with the help of various similarity measurement methods. These methods include Jaccard similarity [12], Pearson correlation coefficient [13], Cosine [8], Euclidean similarity [14], PIP method [15], Heuristics based similarity measure method [35], TMJ similarity [32] etc. CF is being used successfully in many web systems such as Netflix, Amazon, etc. Besides enormous success, CF encounters problems of cold start, data sparsity, and scalability.

Our proposed work aims to improve the accuracy of CF-based systems. The fundamental of CF is to evaluate the similarities between the users or items that have the same interest. The conventional measures of similarity, for example, Pearson (PCC), Cosine, Jaccard, Mean square difference(MSD), are unable to gather fruitful similar users or items, particularly for cold start systems or users, who rate only a few items. This research introduces an improved product recommendation method for CF. Our aim in this research is to improve the triangle similarity method. Triangle similarity works only on items that are rated by both users and ignore items that are rated by either user. The proposed improved triangle similarity measure not only focuses on common ratings but also considers the rating of those items, which are not rated by any user from a pair of users. After that, we complement improved triangle similarity measure with user rating preferences (URP) behavior, which triangle similarity also ignores. We carried experiments on six datasets, namely, Filmtrust, CiaoDVD, Epinions, MovieLens-100K, and MovieLens-1M, to evaluate the performance of the proposed similarity measurement method. The list of notations used in this study is mentioned in Table 1. This research paper is divided into the following sections. I. Introduction (which already cover-up), (II) Literature Review, (III) Proposed Method and Formalization (IV) Experimental setup (V) Experimental results and discussion (VI) Conclusion.

TABLE 1. List of notations.

Notation	Description
m, n	Represents two distinct users
i	Set of items
i_m	Items rated by a particular user m
$r_{m,i}$	Rating of item i given by user m
$I = \{i_m \cap i_n\}$	Set of co-rated items
$\overline{r}_m$	User m average rating
$P = \{i_m \cup i_n\}$	Items rated by either user m or n

II. LITERATURE REVIEW

Despite being widely used recommendation techniques, CF faces a significant reduction in user practice. It is due to specific problems like data sparsity, where users hardly rate an item. Thus, there is not much information about user ratings in the database. Other issues like scalability and cold start are also present. Table 1 present the list of different notations, which are used for the proposed similarity measure method.

A. RELATED WORK AND DRAWBACKS OF EXISTING SIMILARITY MEASURES

Researchers have proposed various similarity measures. One such measure is Pearson's correlated coefficient (PCC) [16]. PCC measures the correlation between two user ratings by identifying their rating preferences as differences in ratings from average rating and user standard deviation in ratings. The mathematical equation for PCC is given in Eq. (1).

$$sim^{PCC}_{(m,n)} = \frac{\sum_{i\in I}(r_{m,i} - \overline{r}_m)(r_{n,i} - \overline{r}_n)}{\sqrt{\sum_{i\in I}(r_{m,i} - \overline{r}_m)}^2 \sqrt{\sum_{i\in I}(r_{n,i} - \overline{r}_n)}^2} \quad (1)$$

Weighted Pearson's correlated coefficient (WPCC) [16], was suggested to overcome the shortcomings of traditional PCC. It apprehends the reliability of neighbors by giving weight to the number of common rated items. PCC ignores the set size of common rated items. The mathematical equation for WPCC is given in Eq. (2).

$$sim^{WPCC}_{(m,n)} = \begin{cases} sim^{PCC}_{(m,n)} \cdot \frac{1}{H}, & |I| \leq H \\ sim^{PCC}_{(m,n)}, & otherwise \end{cases} \quad (2)$$

The value of H is set to 50 in [16], which is an experimental value.

Sigmoid function PCC (SPCC) was introduced that weakens the similarity of a small set of common rated items between users [17]. (2)

$$sim^{SPCC}_{(m,n)} = sim^{PCC}_{(m,n)} \cdot \frac{1}{1 + exp(-\frac{|I|}{2})} \quad (3)$$

In PCC, the value of similarity among users is the difference among the co-ratings of the items by these users. Generally, the similarity tends to be higher when the value of co-rated items is closed. To acquire better recommendations, many manipulative calculation techniques have been evolved that ultimately rely on absolute rating differences. The multi-level (ML) division of PCC [34] considers the similarity obtained from PCC and the number of co-rated items. Authors of [34] claimed that accuracy improves if we split PCC into multiple levels. Working of ML involves checking the number of co-rated items, if it exceeds the pre-defined threshold, only then it proceeds to the next level to calculate the similarity between the users. If the number of co-rated items is quite sufficient and the similarity value obtained from PCC is higher than the pre-set threshold, then the recommendations are returned. In the opposite case, where the number of co-rated items is not enough, zero recommendations return. Despite improving the accuracy of recommendations, this method suffered a limitation that of not being able to provide recommendations where users do not have enough number of co-rated items or a specific PCC value. Moreover inherent problems of [15] exist in ML also.

Different efforts are made to improve conventional CF techniques, such as the Pearson correlation coefficient (PCC), to enhance the accuracy of the recommendations. Systems that utilize PCC, technically consider the absolute ratings between the users while giving the recommendations [33]. However, the modification involved the division of user similarity into multiple levels and adding constraints to each level. In this way, the accuracy and quality of recommendation were improved [34].

To overcome the shortcomings of cosine similarity, the adjusted cosine similarity measure used to include preferences of user rating [15].

$$sim^{ACOS}_{(m,n)} = \frac{\sum_{i \in I}(r_{m,i} - \overline{r}_i)(r_{n,i} - \overline{r}_i)}{\sqrt{\sum_{i \in I}(r_{m,i} - \overline{r}_i)}^2 \sqrt{\sum_{i \in I}(r_{n,i} - \overline{r}_i)}^2} \tag{4}$$

Here I is the set of total items, rated by both users m and n.

Jaccard and Mean Square Difference (MSD) are also the two techniques of similarity measures. Jaccard only counts the number of common ratings between the two users and does not count the absolute values of ratings. This leads to a complication while differentiating between users. MSD uses absolute values that are rated by users, and not count the number of common ratings. The disadvantage of MSD is that it reduces the weight of similarity. JMSD, which is a combination of both Jaccard and MSD solves the partial problems of both Jaccard and MSD. JMSD measure integrates Jaccard similarity into mean squared difference to overcome the cold user issue. In this measure, the prediction is made by initially selecting six similarity measures, and then weights of these measures are learned by neural network learning [18].

$$sim^{JACCARD}_{(m,n)} = \frac{|i_m \cap i_n|}{|i_m \cup i_n|} \tag{5}$$

$$sim^{MSD}_{(m,n)} = 1 - \frac{\sum_{i \in I}(r_{m,i} - r_{n,i})^2}{|I|} \tag{6}$$

$$sim^{JMSD}_{(m,n)} = sim^{JACCARD}_{(m,n)} * sim^{MSD}_{(m,n)} \tag{7}$$

$r_{m,i}$, $r_{n,i}$ is the set of items that are rated by user m and n. From Table 3, it can be seen that there are two types of similarity values that occurred in Jaccard, which are 0.6 and 1.0. This is due to the consideration of the number of common ratings between two users and assigning the same weight to them. The similarity between the $item_2$ and $item_3$, $item_2$ and $item_4$ are 0.6 and the similarity between $item_1$ and $item_2$, $item_1$ and $item_5$, $item_2$ and $item_5$ is 1.0.

PIP measure computes proximity, impact, and popularity of users' ratings [15]. This work initially evaluated the drawbacks of Cosine and PCC, followed by figuring out that these methods are confined to the local information of ratings and ignore global preferences of ratings. It also provides a better solution in both the cold start and non-cold start systems.

The mathematical equation used for PIP similarity is defined as follows:

$$sim^{PIP}_{(m,n)} = \sum_{i \in I} PIP(r_{m,i}, r_{n,i}) \tag{8}$$

where $r_{m,i}$ and $r_{n,i}$ are the ratings of user m and n on a common item i. PIP can also be defined as follows:

$$\begin{aligned}\sum_{i \in I} PIP(r_{m,i}, r_{n,i}) = {} & Proximity(r_{m,i}, r_{n,i}) \\ & * Impact(r_{m,i}, r_{n,i}) \\ & * Popularity(r_{m,i}, r_{n,i})\end{aligned} \tag{9}$$

Proximity examines whether the two ratings are in agreement or not, penalizing cases where ratings are in disagreement. Two ratings are considered to be in disagreement if one of them lies at the lower half of the rating scale, while the other one lies at the upper half. Remaining two factors, impact and popularity rely heavily on the user's agreement. Impact considers how far from the center of the rating scale the two ratings are, under the rationale that ratings that are towards the upper or lower end of the rating scale convey stronger indications of preference. The impact metric takes into consideration both the arithmetic distance between the ratings and whether the ratings are in agreement or not: if the values of the ratings are close, then the value of the impact metric is amplified, whereas if the values of the ratings are distant, then the value of the impact metric is attenuated. Similarly, if the ratings are in agreement, the value of the impact metric is amplified, whereas if the ratings are in disagreement, then the value of the impact metric is attenuated. Popularity examines how close the user's ratings are to the mean rating for the particular item in the database, under the rationale that when two users agree on a rating that deviates from the global view on the item, this offers a stronger indication regarding user-to-user similarity. These three factors are computed for each common rated item between two users. The drawback of PIP is that it only considers the absolute value of rating and some time lead to the wrong results where the similarity between two similar users may be lower as compared to the dissimilar once.

A system called singularity based similarity measure is suggested in [19] to improve the results of conventional similarity measures. It has been practically verified that this measure effectively calculated singularity values instead of similarity values for individual users and items followed by designating their ratings as positive and non-positive [19].

A significance based similarity measure is also presented in [20]. This method computes the significance of the item itself, the significance of an item for other users, and the significance of one user to recommend to another user. After calculating these three significances, the conventional PCC and Cosine similarity were applied to evaluate users' similarities following the significances [20].

Another widely used technique is the data smoothing technique. It utilizes several sparsity measures to boost the efficiency of CF based on similarities on local as well as global levels. It also suggests the utility of certain definitive parameters to weigh sparsity measures. Experiments have shown that these parameters surpass the efficiency of the prediction of ratings achieved by keeping certain parameters constant [11].

Another algorithm that took under consideration the data provided both by the user and the item was proposed in [21]. In this algorithm, similarity among items and users was adjusted to predict missing information, if the intersection of neighbors of users as well as the items are not left vacant. It is called the partial missing data prediction algorithm. Another method called the iterative prediction method uses explicit ratings to concentrate the sparse matrix achieved from user and item ratings. A spectral clustering algorithm is used for that purpose [22].

A combination of singular value decomposition (SVD) and random indexing as a hybrid method was also proposed in the literature. This method incorporates the management of data by monitoring the item description and users' behavior [23]. Moreover, probabilistic matrix factorization [24], principal component analysis (PCA) [25], singular value decomposition [26], cluster-based smoothing method [27], Neural Networking, are among the significant approaches used to enhance the effectiveness of CF through solving various computational setbacks.

Another heuristic-based similarity measure method that simultaneously embodies Proximity-Significance-Singularity (PSS) measures along with modified Jaccard similarity is formalized in [35]. Traditional PIP similarity faces serious shortcomings such as it is not normalized and use. Consider the absolute rating of the users only and computation is too complicated. Improved measure in [35] entails Jaccard features, thus not only considers the absolute rating but also takes into account the proportion of common ratings, providing high accuracy. Moreover, the similarity is determined not only by considering the local context only but also by the global preferences of user behavior.

The formalization implies building a similarity measure with a non-linear function based on initial PIP similarity, which is linear mainly. This new similarity measure, named as NHSM, is normalized and can be easily combined with other similarity measures. Besides, this novel similarity measure effectively overcome the shortcomings and drawbacks of traditional ones [15].

$$sim^{PSS}_{(m,n)} = Proximity * Significance * Singularity \tag{10}$$

$$sim^{Jaccard'}_{(m,n)} = \frac{|I_m \cap I_n|}{|I_m| * |I_n|} \tag{11}$$

$$sim^{JPSS}_{(m,n)} = sim^{PSS}_{(m,n)} * sim^{Jaccard'}_{(m,n)} \tag{12}$$

$$sim^{urp}_{(m,n)} = 1 - \frac{1}{1 + exp(-|\overline{r}_m - \overline{r}_n|.|\sigma_m - \sigma_n|)} \tag{13}$$

$$sim^{NHSM}_{(m,n)} = sim^{JPSS}_{(m,n)} * sim^{urp}_{(m,n)} \tag{14}$$

In Eq. (10) proximity factor corresponds to a sigmoid difference in user ratings, normalized in the range of zero to one. The significance factor corresponds to sigmoid difference of user ratings form median rating, normalized in the range zero to one. Singularity factor computes the sigmoid difference of both users' average rating forms the average rating of the target item. The singularity factor is also normalized in the range zero to one.

Sparse rating datasets have always been a severe issue for the traditional similarity measures. Another approach known as a modified heuristic similarity measurement model has been proposed to tackle data sparsity. It employed the combination of three different similarity methods. One of them is Jaccard similarity; for the computational analysis of ratings assigned by two users. Second is the modified Bhattacharya coefficient measure; to find out computationally, the divergence between the ratings assigned by two users. The third one is the Proximity-Significance-Singularity (PSS); for absolute ratings given by two users during the similarity computation. This improvising employed the idea of combining the local context along with the global preferences of the user demeanor. The model thus suggested having an improved performance of various recommender systems by allowing more accurate prediction of adjacent neighbors [36].

Another study proposed a hybrid model combining triangle and Jaccard similarities, collectively called triangle multiplying Jaccard (TMJ) similarity [32]. The performance of this approach is tested using item-based CF [28], [29]. Since triangle similarity faces a limitation of considering co-rating of users solely. TMJ entails the qualities and benefits of both triangle and Jaccard similarities [30], [31]. This new hybrid measure is thus designed to provide more information about co-rated as well as non-co-rated users effectively. It has been shown that TMJ allows the system to surpass all the counterparts in reference [32]. The mathematical formulation of TMJ similarity is given below in Eq. (15), Eq. (16), and Eq. (17).

$$sim^{TRIANGLE}_{(m,n)} = 1 - \frac{\sqrt{\sum_{i \in I}(r_{m,i} - r_{n,i})}^2}{\sqrt{\sum_{i \in I} r_{m,i}}^2 + \sqrt{\sum_{i \in I} r_{n,i}}^2} \tag{15}$$

$$sim^{JACCARD}_{(m,n)} = \frac{|i_m \cap i_n|}{|i_m \cup i_n|} \tag{16}$$

$$sim^{TMJ}_{(m,n)} = sim^{TRIANGLE}_{(m,n)} * sim^{JACCARD}_{(m,n)} \tag{17}$$

Different efforts are made to improve conventional CF techniques, such as the Pearson correlation coefficient (PCC), to enhance the accuracy of the recommendations. Systems that utilize PCC, technically consider the absolute ratings between the users while giving the recommendations [33]. However, the modification involved the division of user similarity into multiple levels and adding constraints to each level. In this way, the accuracy and quality of recommendation were improved [34].

An improvement of the Jaccard index has also been proposed in [37], which manifested the frequency of ratings given by users as well as the number of items co-rated by users. The number of commonly rated items by two users can be calculated by this index, regardless of the rating values, anyhow it is a fact that the rating of the common items is normal or extreme. This modification proved to overcome the major setbacks associated with the conventional Jaccard index, especially the ignorance of actual rating value [37].

From the previous discussion, it can be concluded that Jaccard similarity considers only common rating items of users and ignore the absolute value of rating. The researchers consider the absolute rating values, which are equal in value, to the total no of co-rated items. Another parameter which included is the average rating value of users. After this, the researcher compared the performance of the proposed method with many traditional similarity measures. The recommendation results show that the proposed method performance is found well in terms of several evaluation metrics as compared to other traditional methods [38].

Another novel similarity measure method was proposed in [39] which considers user RPB while calculating the similarities among users. This method state user RPB as a function of cosine taking user average rating value, and variance or standard deviation as an input value. The user RPB is then combined with an improved model of standard PCC. Improved PCC weighted with RPB (IPWR) is the name given to this method. The qualitative and quantitative analysis of the IPWR similarity measure method is tested on six datasets i.e. MovieLens-1M, Epinions, CiaoDVD, MovieLens-100K, FilmTrust, and MovieTweetings. The result shows that the performance of the IPWR similarity measure method was better than the traditional similarity measure methods in terms of several evaluation metrics.

III. PROBLEM FORMALIZATION AND METHODOLOGY

A. THE MOTIVATIONS

The proposed similarity measure introduces an improved product recommendation method for CF, which is based on improvement in the triangle similarity. The results of triangle similarity are although better as compared to some other similarity measurement methods developed up-to-date, but there is still a space to improve the result accuracy of this method. The downside of triangle similarity is that it only considers the common ratings of users. The proposed similarity measure not only focuses on common ratings but also find the rating of those items, which are not commonly rated. For this purpose, we built a union set of items rated by either user. In this way, we aim to incorporate ratings of those items, which are not commonly rated.

As discussed in section II, TMJ was applied on item-based CF, due to its better performance as compared to user-based CF. There is no doubt TMJ performance is tremendous when we compare the results with the other similarities. TMJ similarity is the product of Jaccard and Triangle similarity, due to which it takes the advantages of both similarities while producing recommendations [32]. Triangle similarity recommendations are based on common ratings of users and ignore the remaining ratings of users. On the other hand, Jaccard's similarity is the ratio of an intersection of the items to the unions of the items rated by the user. Jaccard's recommendation is based on the non-common ratings, but it ignores the absolute values which are rated by the user. This hybrid measure was designed to provide more information about co-rating as well as non-co-rating items, but accuracy is still an issue. In our work, we improve recommendations by eliminating the downside of the triangle. The obtained similarity is further complemented with the URP of the user to achieve the best performance.

Similarly, if we look at Jaccard similarity as in Eq (16), there is a symbol of intersection in the numerator. This symbol enforces Jaccard's similarity to consider those items that are rated by both the users. The union symbol allows considering items rated by either user. As a result, the accuracy of the recommendation is affected. Another downside of the Jaccard similarity is that it does not consider the user's absolute value rating. Due to which, it assigns the same weightage to all users regardless of the number of common ratings. For example, if $user_1$ rates 1.0 to $item_1$ and $user_2$ rates 2.0 to $item_1$ then the similarity between two users will be 0.5 (i.e. 1/2). Similarly, if $user_3$ rates 2.0 to $item_1$ and $user_4$ rates 4.0 to $item_1$ then again the similarity between two users will be 0.5. Therefore, the same weightage is assigned, even though the ratings are different.

TABLE 2. Example user-item rating matrix.

Users	$item_1$	$item_2$	$item_3$	$item_4$	$item_5$
m	4	3	5	4	2
n	5	1	0	0	4
o	4	2	1	2	1
p	2	1	0	1	2
q	4	2	2	0	2

Table 3 shows the results of different similarity measurement methods applied to Table 2. Improved Triangle ($Triangle'$) and ITR are the results of the proposed similarity measure which takes lead on TMJ. Although TMJ performs well as compared to the other CF similarity measure methods.

The following are some points that motivate us to improve the original triangle similarity measure.

(1) Triangle similarity recommendations are only based on the common ratings of users and do not uses all ratings given by the user. Triangle similarity ignores the ratings of those items, which are not rated by the users commonly. Due to which accuracy of recommendations is lagging. For example, from Table 3 triangle similarity between item1 and item2 is 0.611, and item1 and item 3 are 0.698. The total rated values of item2 are more than item 3 but the similarity is vice versa. This is just because it ignores those ratings, which are not rated by common users.

(2) The results of TMJ are tremendous when we compare it with other similarity measures. As the TMJ similarity is the product of Jaccard and Triangle similarity and we already discussed the disadvantages of both similarities individually. If we talk about the results of TMJ, it mostly performs well, but sometimes it also leads to the wrong recommendations. For example, from Table 3 the similarity between item 1 and item 3 is 0.418, and item 1 and item 4 are 0.473. The sum of rated values of item 3 is more than item 4, but the similarity

TABLE 3. Computed values of different similarities.

Similarity		i2	i3	i4	i5
COS	i1	0.915	0.843	0.945	0.931
	i2	–	0.929	0.991	0.766
	i3		–	0.964	0.912
	i4			–	0.872
PCC	i1	0.218	0.000	0.677	0.458
	i2	–	0.641	0.946	−0.490
	i3		–	0.910	0.449
	i4			–	0.120
ACOS	i1	0.218	0.153	0.069	0.458
	i2	–	0.823	0.670	−0.490
	i3		–	0.836	−2.379
	i4			–	−0.383
CPCC	i1	−0.447	0.192	0.471	−0.125
	i2	–	0.707	0.912	0.335
	i3		–	1	0.408
	i4			–	0.5
Jaccard	i1	1	0.6	0.6	1
	i2	–	0.6	0.6	1
	i3		–	0.4	0.6
	i4			–	0.6
Triangle	i1	0.611	0.698	0.788	0.910
	i2	–	0.767	0.879	0.644
	i3		–	0.852	0.646
	i4			–	0.676
TMJ	i1	0.611	0.418	0.473	0.910
	i2	–	0.460	0.527	0.644
	i3		–	0.340	0.387
	i4			–	0.405
Triangle′	i1	0.611	0.539	0.492	0.700
	i2	–	0.731	0.726	0.644
	i3		–	0.736	0.504
	i4			–	0.488
URP	i1	0.386	0.306	0.403	0.5
	i2	–	0.304	0.433	0.476
	i3		–	0.462	0.416
	i4			–	0.491
ITR	i1	0.235	0.164	0.198	0.35
	i2	–	0.222	0.314	0.306
	i3		–	0.340	0.209
	i4			–	0.239

is vice versa. The same is the case with item 2 and item 3 and item 2 and 4.

(3) In TMJ, Jaccard considers those items, which are commonly rated by users, but it does not include the absolute value of the user's ratings. Jaccard is just a ratio of an intersection to the union between items. Therefore, when we combine Jaccard with another similarity then definitely it leads to the wrong recommendation at some places. These are the problems, which motivate us to improve the similarity of Triangle and TMJ. Therefore, in this paper, we will work on the triangle similarity and ignore Jaccard due to its various drawbacks.

B. METHODOLOGY

This section explains and formalizes our proposed similarity method ITR. In section III (A), we discuss the problems of existing similarity measurement methods. Section III also explains TMJ similarity methods issues in detail, which motivate us to present the new similarity method. To avoid those problems, which exist in the TMJ triangle similarity, we will discuss the proposed method below, which improves the accuracy of recommendations.

This research introduces an improved product recommendation method for CF. Our work is based on triangle similarity. The proposed similarity measure not only focuses on the common rating of users but it also considers the non-common rating of users. We named this similarity as improved triangle similarity. The obtained similarity is further complemented with rating preferences of users to achieve more accuracy in recommendations. User rating preferences are calculated by using the URP similarity as given in Eq (13). Here, we preferred to relinquish Jaccard similarity, which was used in TMJ, due to its multiple disadvantages. The improved triangle similarity, denoted by $sim_{(m,n)}^{TRIANGLE'}$, is computed as follows. Let P be a subset of items i, either rated by user m or n.

$$P = \{i \in i_m \cup i_n\}$$

$$sim_{(m,n)}^{TRIANGLE'} = 1 - \frac{\sqrt{\sum_{i \in P}(r_{m,i} - r_{n,i})}^2}{\sqrt{\sum_{i \in P} r_{m,i}}^2 + \sqrt{\sum_{i \in P} r_{n,i}}^2} \tag{18}$$

Here $i \in P$ indicates the set of items rated by either user m or n. If any user does not rate an item, then the rating of $r_{m,i}$ or $r_{n,i}$ is considered as zero. We will use that rating while computing the improved triangle similarity. For example, from Table 2, five users have rated item1 and three users rated item3. We will consider all five ratings of item1 while computing the similarity between item1 and item3. It can be also noticed that the similarity comparison in the improved triangle also provides fine results. From Table 3, the similarity between item1 and item2 is 0.611, and item1 and item3 are 0.539. The total ratings given to item2 are more than item3 and hence similarity value of item2 should be greater than item3. Similarly, the similarity of item1 and item2 is also greater than the similarity between item1 and item3. Consequently, by considering all ratings of items in an improved triangle, the accuracy of recommendations is improved.

To get better results, improved triangle similarity is complemented with the URP of users. URP is given in Eq (13). Therefore, the total similarity of the improved triangle complemented with URP is given in Eq (19), with the name given as *ITR*.

$$sim_{(m,n)}^{ITR} = sim_{(m,n)}^{TRIANGLE'} * sim_{(m,n)}^{urp} \tag{19}$$

For prediction generation, we used Resnick's formula [39], which is given below in Eq (20).

$$\hat{r}_{m,i} = \bar{r}_m + \frac{\Sigma_{n \in NN} sim^{ITR}_{(m,n)} \cdot \left(r_{n,i} - \bar{r}_n\right)}{\Sigma_{n \in NN} \left|sim^{ITR}_{(m,n)}\right|} \quad (20)$$

where $\hat{r}_{m,i}$ denotes the predicted rating of missing item i for user m. Here NN is a set of users having computed similarity above than a threshold.

Suppose we want to predict the value of $item_3$ for user n as shown in table 2 using item-based CF. Similarities of all items are already presented in table 3 for ITR similarity and other similarity computation methods. Only users m, o and q have rated $item_3$ in this specific example. As user n has not rated item i_4 so we will not use this item for rating prediction $\hat{r}_{n,i_3}$ and $\hat{r}_{n,i_3}$, as shown at the bottom of the page.

Algorithm 1 Procedure to Predict a Rating

Input: m (target user), i (target item), Ratings dataset

Output: Predicted rating $\hat{r}_{m,i}$

1. To predict the rating of target item i, for user m find set of users n in the training set who have rated item i.
2. Gather a set of items P which both users m and n have rated.
3. For each item in P, determine improved triangle similarity, $sim^{TRIANGLE'}_{(m,n)}$, using Eq. (18).
4. Determine the match of user rating preferences of users m and n using Eq. (13).
5. Obtain proposed ITR similarity using Eq. (19).
6. Ignore users having similarity less than a threshold.
7. Rank set of similar users according to obtained ITR similarity in step 7.
8. Find K nearest neighbors of target user m.
9. Make prediction $\hat{r}_{m,i}$ on target item i of target user m using Eq. *(20).*
10. Return $\hat{r}_{m,i}$

End procedure

Algorithm 1 describes the step by step working of the proposed similarity measure in the recommendation process. Discussion on computation time or time complexity of algorithm 1. The computation time of step 1 is $O(|N|)$ where n is the number of users, as this step is repeated for each pair of users. The complexity of step 2 to 5 will be $O(|N|*|P|)$ as this step is performed for P items. The complexity of step 6 will be $O(|N|)$ as this step is performed for only top-k similar users. Worst-case complexity for step 7 and 8 is $(\log(|N|))$. Overall algorithm complexity will lead to $O(|N|*|P| + \log|N|)$.

IV. EXPERIMENTAL SETUP

A. DATASETS USED FOR EVALUATION

The five datasets used in experiments are Filmtrust, CiaoDVD, Epinions, MovieLens-100K, and MoviLens-1M. The sixth dataset, which is MovieTweetings, is covered in a separate sub-section V(E). All the datasets are open source and available for research. The first data set is Filmtrust, and in this data set, the number of users and items is 1508 and 2071 respectively. The ratings available are 35,497, and the rating scale is 0.5 to 4.0. The second data set is CiaoDVD and in this dataset number of users and items is 17,615 and 16,121 respectively. The ratings available in this dataset are 72,665, and the rating scale is 1.0 to 5.0. The third data set is Epinions and in this data set number of users and items is 40,163 and 139,738 respectively. The ratings available in this dataset are 664,823 and the rating scale is 1 to 5. The fourth dataset is MovieLens 100K and in this dataset, several users and items are 943 and 1,682 respectively. The ratings available in this dataset are 100,000, and the rating scale is 1.0 to 5.0. MovieLens 1M (ML-1M) contain 6,040 users, 3952 movies and 1,000,209 ratings. We choose the above six datasets just because the researchers and developers mostly use these data sets in the field of CF.

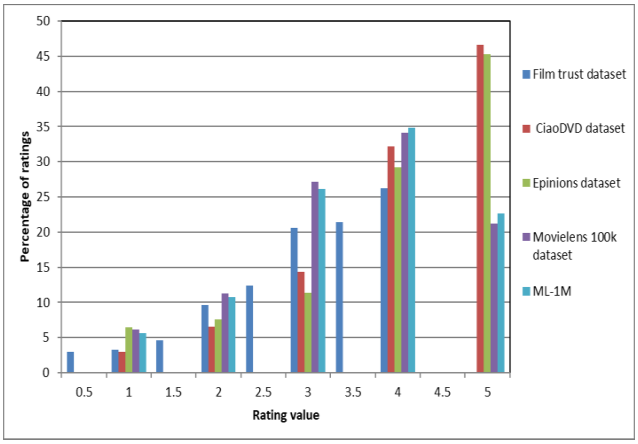


FIGURE 1. Rating distribution in five datasets excluding MovieTweetings dataset.

Fig. 1 shows the rating distribution of five datasets excluding the MovieTweetings dataset, whose distribution is shown

$$\hat{r}_{n,i_3} = \frac{sim^{ITR}_{(i_3,i_1)} * (r_{n,i_1} - \bar{i}_1) + sim^{ITR}_{(i_3,i_2)} * (r_{n,i_2} - \bar{i}_2) + sim^{ITR}_{(i_3,i_5)} * (r_{n,i_5} - \bar{i}_5)}{sim^{ITR}_{(i_3,i_1)} + sim^{ITR}_{(i_3,i_2)} + sim^{ITR}_{(i_3,i_5)}}$$

$$\hat{r}_{n,i_3} = \frac{0.164 * (5 - 3.8) + 0.22 * (1 - 1.8) + 0.209 * (4 - 2.2)}{0.164 + 0.22 + 0.209}$$

$$= \frac{0.748}{0.593} = 1.261 \simeq 2.0$$

separately in figure 5. Ratings value 4 has a major share of ratings in the Filmtrust dataset with an approximate percentage of 26%. For CiaoDVD and Epinions datasets, the most used rating value is 5 with an approximate percentage of 45%. The least used rating value in these datasets is 1. For both datasets of MovieLens, the most used rated value is 4 having an overall 35% weightage. The least used rating value for both datasets is also 1.

B. FIVE-FOLD CROSS VALIDATION

Cross-validation is a process that is used by many researchers in machine learning in predicting the results or performance of models. In the cross-validation or machine learning model, there are two types of data sets, which are used to measure the performance of a model. One of them is known as a test set, and the other is known as a training set. We need to train our model on the training set and test the accuracy of this model on a test set. To reduce the error, several iterations are performed on different training sets. After that, results from all these iterations will be taken for average to evaluate the accuracy of the model. The limitation of this K-fold is that the test and training sets are required to measure the performance of any machine learning model.

We performed 5-fold cross-validation on datasets, a variant of K-fold cross-validation. In K-fold cross-validation, the whole dataset is divided into K subsets. One subset is taken as a test set. The remaining K-1 subsets are used as training sets. Therefore, in our experiment, we divide each dataset into two parts. Twenty percent of users from each dataset is selected as a test user, and remaining are considered as training users.

C. EVALUATION METRICS

To predict the user ratings for an item, the metrics mostly used by the researcher are Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). Although MAE and RMSE both evaluate the average magnitude of the error, the best results of these two methods don't mean that they provide a satisfactory recommendation to the user.

MAE and RMSE can be calculated as:

$$\mathrm{MAE} = \frac{1}{N}\sum\nolimits_{i=1}^{N} |r_{m,i} - \hat{r}_{m,i}| \tag{21}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum\nolimits_{i=1}^{N} |r_{m,i} - \hat{r}_{m,i}|^2} \tag{22}$$

MAE and RMSE results are a numerical value in the range of 0 to max rating scale value. Thus, we need metrics like Precision, recall, and F-score whose output value is binary. This binary value is measured as relevant and non-relevant items. An item is considered as relevant if its actual value in the test set is above the user average value and non-relevant in the reverse case. An item declared as relevant for a specific user means the item is a good recommendation for the target user. These metrics are used to evaluate the satisfaction of users. The output of most recommender systems is not a numerical value instead output is a list of recommended items in a ranked fashion termed as a top N recommendation list. Recall and precision rely on the accuracy of the top N recommendation list. So to measure the performance of the suggested *ITR* method, the accuracy is measured in terms of precision and F-Score measure. The precision score is defined as the mean proportion of items in the top N list that are relevant. Recall score is defined as the mean proportion of relevant items appearing in the top N recommended list. The readings from the recall score must be high to achieve the best efficiency. The following formula is used to calculate recall:

$$\text{Precision} = \frac{n}{TopN} \tag{23}$$

$$\text{Recall} = \frac{n}{M_T} \tag{24}$$

In Eq (23), n is the number of items appearing in the recommended list and relevant to the testing user. In Eq (24), M_T is the total number of relevant items in the testing set. We used precision and F-Score metric to measure the accuracy of the top N recommendation list. Precision also needs to be as high as it can be for a resultant efficient performance.

To balance the tradeoffs of precision and recall, F-Score is used by most of the researchers. The F-Score is mathematically expressed in Eq. (25).

$$F - Score = 2.\frac{Precision.Recall}{Precision + Recall} \tag{25}$$

V. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, first we will analyze the impact of the similarity threshold on the proposed method. Then we will discuss the result of several experiments performed by using five datasets i.e. CiaoDVD, Epinions, Filmtrust, MovieLens-100K, and MovieLens-1M. This section also includes the comparison of the proposed similarity measure method, ITR, with other similarity measure methods, which include PCC, TMJ, Jaccard, Singularity measure, NHSM, CPCC, Cosine, and PIP. MAE, RMSE, Precision, and F-Score are used to measure the performance of the proposed model. The above accuracy measures are calculated on different values of K nearest neighbors on proposed and competitor similarity methods. In the experimental results, MAE, RMSE term of accuracy has reduced and precision term of accuracy has increased. This response is excellent and is required for an accurate recommender system.

A. IMPACT OF SIMILARITY THRESHOLD

AS CF systems use different measures to determine similarities of items or users. So determining a proper threshold for such measures is of particular importance. Our proposed method's results can vary in the range of 0 to 1.0. So we performed experiments to determine which threshold gives the best results in this range. Table 4 gives the impact of this threshold on MAE. Also for determining this threshold, the value of K nearest neighbors is set to 5. By careful analysis of table 4, we observe two trends on datasets. Trend1 applies to ML-1M, Epinions, Filmtrust, and MovieTweetings dataset. In this trend, an Increase in threshold either results in no

TABLE 4. Impact of varying similarity threshold on used datasets using MAE.

	Similarity Threshold									
Dataset	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
ML-1M	0.706	0.706	0.706	0.707	0.736	0.796	0.822	0.822	0.822	0.822
Epinions	0.86	0.861	0.865	0.912	0.925	0.926	0.927	0.927	0.926	0.926
CiaoDVD	0.762	0.762	0.764	0.796	0.804	0.805	0.805	0.803	0.803	0.803
MovieTweetings	1.123	1.123	1.142	1.184	1.142	1.25	1.25	1.253	1.253	1.253
ML-100K	0.738	0.738	0.738	0.739	0.734	0.704	0.6	0.556	0.542	0.53
FilmTrust	0.624	0.624	0.624	0.626	0.631	0.632	0.642	0.637	0.635	0.634

change or impact or increases MAE. If MAE is increased then we can say that accuracy of the system is decreased. The second trend, trend 2, is only visible on the ML-100K dataset. On this dataset, an increase in threshold from 0 to 1.0 decreases MAE thus increasing the accuracy of the system. Keeping in view both trends, we kept the value of similarity threshold to 0 for all other experimentation.

B. NUMBER OF K NEAREST NEIGHBOURS

In K nearest neighbors, K indicates the number of the nearest neighbors. (KNN). In KNN, changing the values of K will result in a different recommendation. Figs. 2(a-e) show the performance of different similarity measure methods with different values of K. This performance is measured in terms of MAE in Fig 1. Similarly, Figs. 3(a-e) show the results of RMSE on different similarity measures. Figs. 4(a-e) show the precision of different similarity measures with different values of K.

From all the figures mentioned above, we noticed that the values of the MAE and RMSE of the proposed method are reduced as compared to other methods. Therefore, we can say that the accuracy of this proposed method is better as compared to the traditional similarity measure. Similarly, we also noticed that the precision of the proposed method has got increased as compared to other methods on all the datasets which we used in this research.

C. MAE ON DIFFERENT DATASETS

The current section discusses the results of MAE on the proposed similarity measure method, ITR, and various other similarity methods are discussed using previously mentioned five datasets. The below Figs. 2(a-e) show the result of MAE on different similarity measures with different values of K.

1) MAE ON CIAODVD DATASET

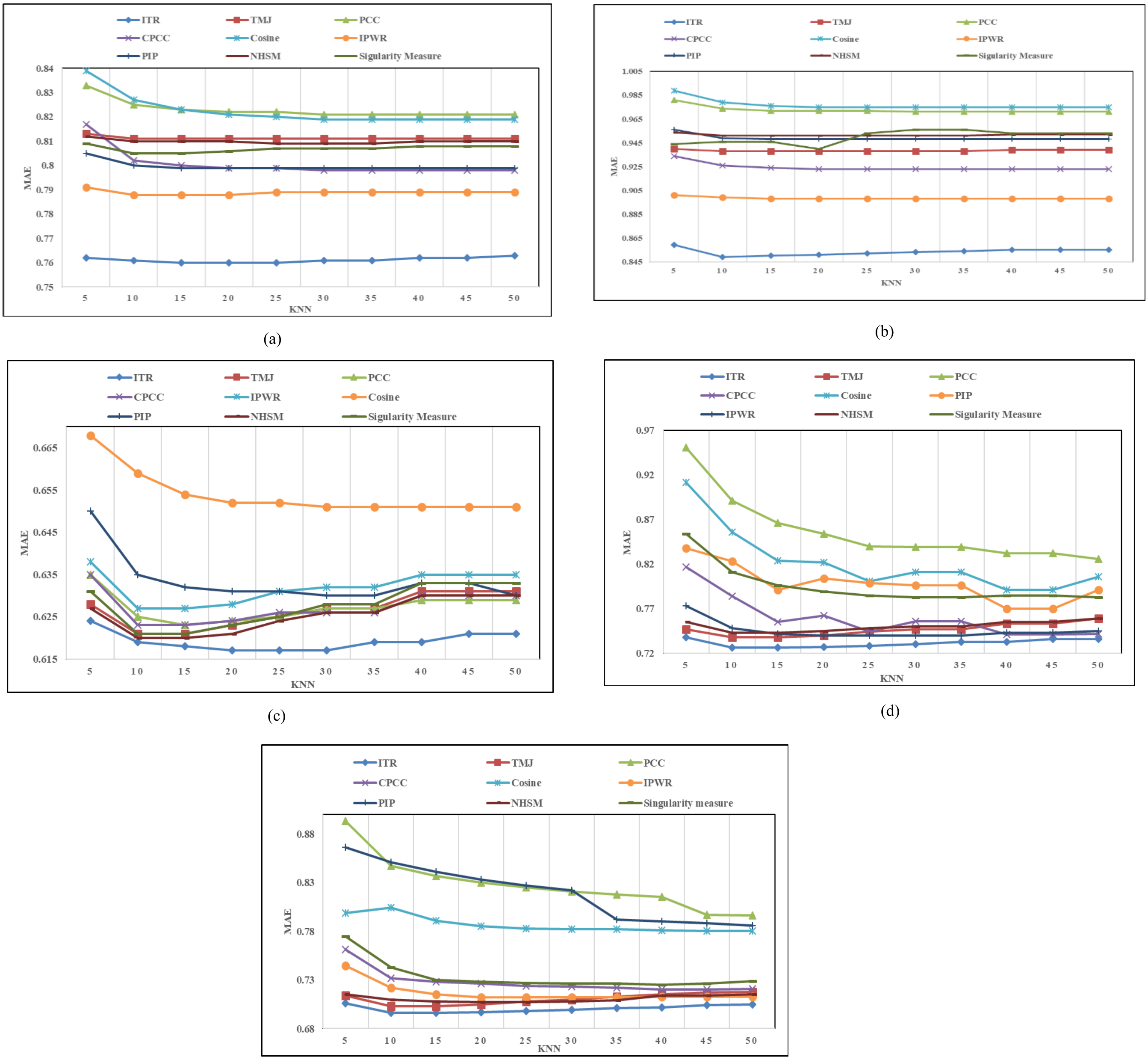
In this section, the MAE of the ITR method and various other similar methods are discussed by using the CiaoDVD dataset. The proposed ITR method's performance is tremendous for different values of K, which returns the least MAE as compared to other methods. After ITR, IPWR similarity performs better on CiaoDVD Dataset. The performance of the Cosine measure is not satisfactory on the CiaoDVD dataset. The performance of the other similarity methods lies between the proposed (ITR) and Cosine method as shown above. For all methods, a slight decrease in MAE occurs when K changes from 5 to 10. Subsequently, any increase in K does not affect performance much more and remains almost constant. Therefore, it can be concluded that an increase or decrease in the number of neighbors does not affect the MAE of the proposed ITR method.

2) MAE ON EPINIONS DATASET

In this section, the MAE of the proposed ITR method and various other similarity measurement methods are discussed, by using the Epinions dataset. Our proposed method's performance is fantastic on this dataset with all values of K. Experiment returns least MAE for the proposed ITR method as compared to other methods. After ITR, IPWR similarity performs better on the Epinions dataset. Similarly, on the other side, the performance of Cosine is worst on Epinions Dataset. The performance of other similar methods lies between the proposed ITR and Cosine similarity method as shown in Fig. 2(b). The proposed ITR method set back all other methods with a great margin. Except for the singularity measure method, the performance of all methods is almost constant after K increases from ten.

3) MAE ON FILMTRUST DATASET

This section evaluates the MAE of the proposed method and various other methods using the Filmtrust dataset. FilmTrust is a smaller dataset. The proposed method performance is also better on this dataset for different values of K. The proposed ITR method gives the least MAE as compared to other methods. After ITR, NHSM similarity performs better on the Filmtrust dataset. Similarly, on the other hand, the performance of Cosine is worst on Filmtrust Dataset. The performance of the other methods lies between the proposed (ITR) and Cosine similarity method as shown in Fig 2(c). Varying values of K on this dataset has a different behavior than Epinions and CiaoDVD datasets. For values of K up to 15, MAE decreases and then start increasing slightly. For proposed ITR this increase starts from the value of $K = 35$. Unlikely to all other methods increase or decrease in MAE with increasing K is small.



(e)

FIGURE 2. (a) MAE results on the CiaoDVD dataset. (b) MAE results on the Epinions dataset. (c) MAE results on the FilmTrust dataset. (d) MAE results on the MovieLens-100K dataset. (e) MAE results on the MovieLens 1M.

4) MAE ON MOVIELENS-100K DATASET

In this section, the MAE of the proposed ITR method and various other similar methods are discussed using the MovieLens-100K dataset. Our method's performance is also better on this dataset for each value of K. Experiments returns least MAE for the ITR method as compared to other methods.

After ITR, TMJ similarity performs better on the MovieLens-100K dataset. On the other side, the performance of PCC is worst on the MovieLens-100K dataset. The performance of the other similarity methods lies between the proposed ITR and PCC similarity method as shown in Fig. 2(d). Good performance of ITR is of course due to its capability to utilize full ratings and incorporation of URP. Unlikely to all other methods effect of change in a number of neighbors for the proposed ITR method is small.

5) MAE ON MOVIELENS-1M DATASET

Here MAE of the proposed similarity method and various other similarity methods are discussed using the MovieLens-1M dataset. The proposed similarity method's performance is also better on this dataset with different values of K. Experiments on this dataset returns least MAE for our proposed ITR method as compared to other methods. On the contrary, TMJ similarity performs better on the MovieLens-1M dataset. On the other side, the performance of PCC is worst on MovieLens-1M Dataset. The performance of the

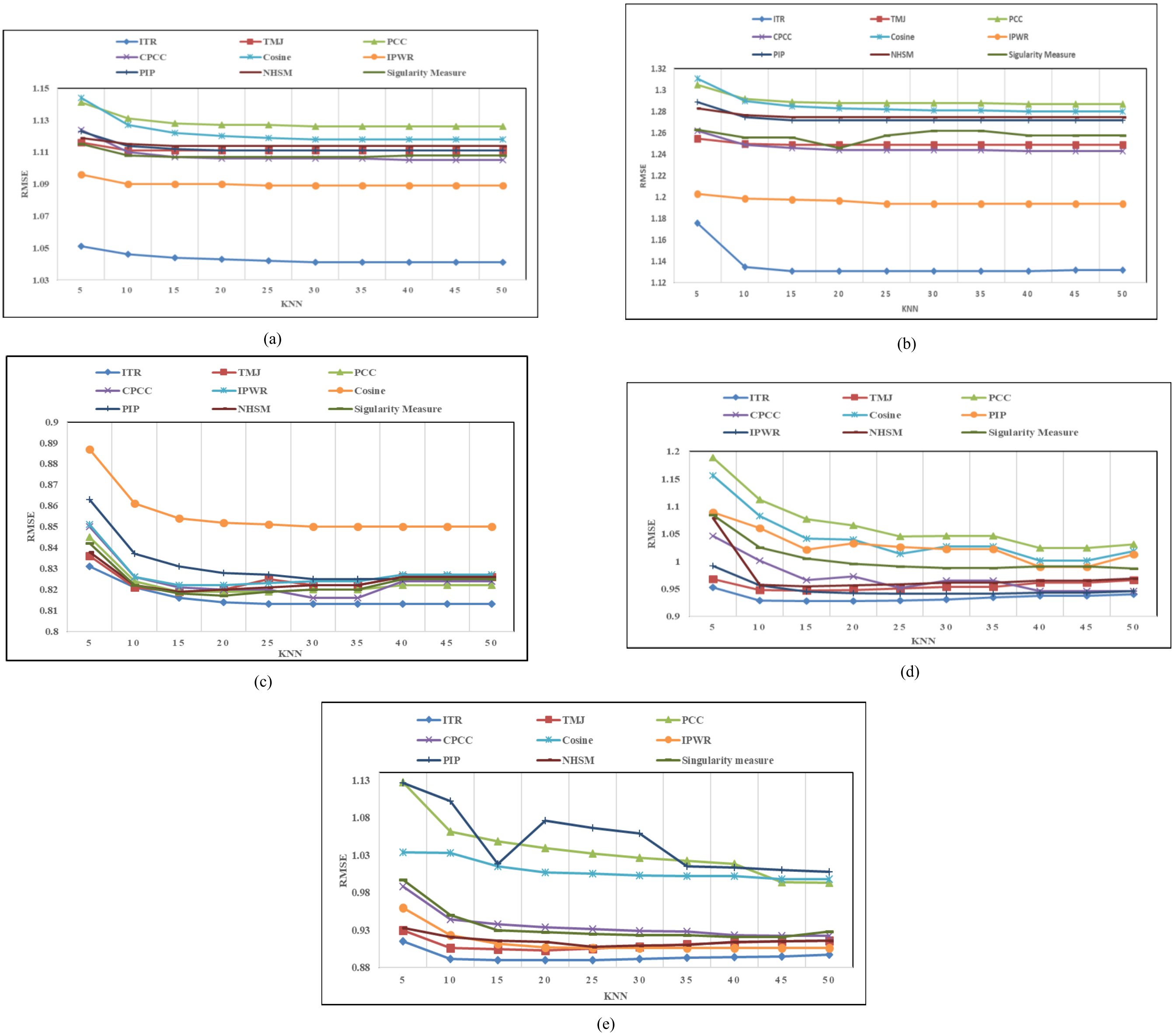


FIGURE 3. **(a) RMSE results on the CiaoDVD dataset. (b) RMSE results on the Epinions dataset. (c) RMSE results on the FilmTrust dataset. (d) RMSE results on the MovieLens-100K dataset. (e) RMSE results on MovieLens-1M.**

other similarity methods lies between the proposed (ITR) and Cosine similarity method as shown in Fig. 2(e).

D. RMSE ON DIFFERENT DATASETS

This section shows the results of RMSE on the proposed ITR method and various other similarity measure methods on the selected five datasets. Figs. 3(a-e) show the result of RMSE on various similarity measures with varying values of K nearest neighbors.

1) RMSE ON CIAODVD DATASET

This section discusses the RMSE of the proposed ITR method and various other similarity methods using the CiaoDVD dataset. The experiment on this dataset returns the least RMSE for our proposed ITR method as compared to other methods. After ITR, IPWR similarity performs better on the CiaoDVD dataset. On the other side, the performance of PCC is worst on the CiaoDVD dataset. The performance of other methods lies between the proposed ITR and PCC similarity method as shown in Fig. 3(a). An increase in KNN initially decreases RMSE and after KNN = 15, no change is observed.

2) RMSE ON EPINIONS DATASET

In this section, RMSE of the proposed method and various other similarity methods are discussed using the Epinions dataset. The experiment on this dataset returns the least RMSE for our proposed ITR method, as compared to other methods. After ITR, the IPWR method performs better on

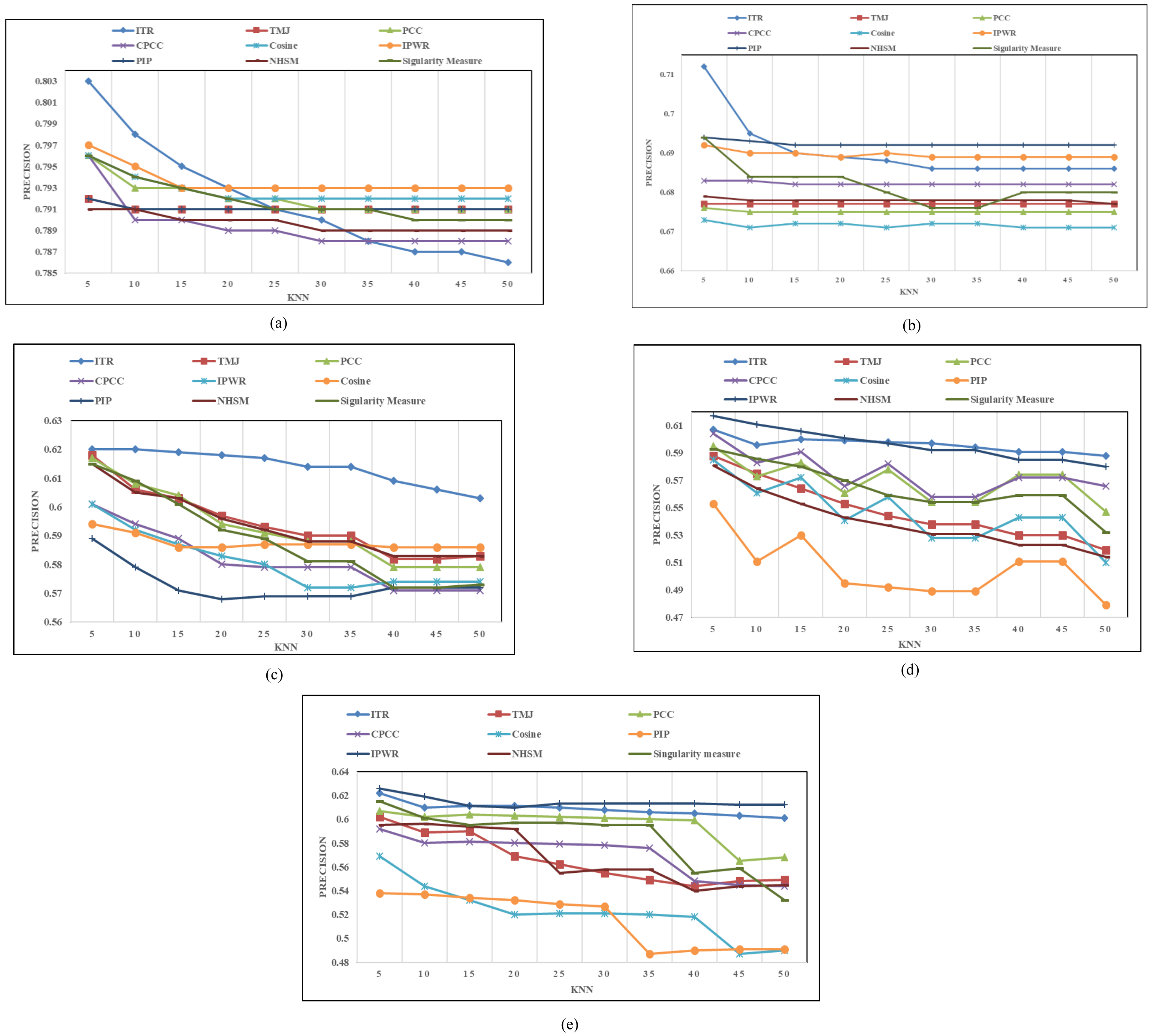


FIGURE 4. (a) Precision results on the CiaoDVD dataset. (b) Precision results on the Epinions dataset. (c) Precision results on the FilmTrust dataset. (d) Precision results on the MovieLens-100K. (e) Precision results on MovieLens-1M.

the Epinions dataset. On the other side, the performance of Cosine and PCC is worst on the Epinions dataset. The performance of the other similarity methods lies between the proposed ITR and Cosine similarity method as shown below. When K increases from 5 to 10 RMSE decreases and afterward becomes constant.

3) RMSE ON FILMTRUST DATASET

In this section, the RMSE of the proposed similarity method and other competitor methods are discussed by using the FilmTrust dataset. The experiment on this dataset returns the least RMSE as compared to other methods. After ITR, TMJ similarity performs better on the FilmTrust dataset. On the other side, the performance of Cosine is worst on the FilmTrust dataset. The performance of the other similarity methods lies between the proposed ITR and Cosine similarity method as shown in Fig. 3(c). For all methods increase in KNN decreases RMSE.

4) RMSE ON MOVIELENS-100K DATASET

In this section, the RMSE of the proposed method and other competitor methods are discussed by using the MovieLens 100K dataset. The experiment on this dataset returns the least RMSE as compared to other methods. After ITR, TMJ similarity performs better on MovieLens 100Kdataset. On the other side, the performance of PCC is worst on the MovieLens 100K dataset. The performance of other similar methods lies between the proposed ITR and Cosine similarity method as shown in Fig. 3(d). No considerable change in performance is visible for the proposed ITR method for the increased value of KNN.

5) RMSE ON MOVIELENS-1M DATASET

In this section, the RMSE of the proposed similarity method and various other competitor methods are discussed by using the MovieLens-1M dataset. The experiment on this dataset returns the least RMSE as compared to other methods. After ITR, TMJ similarity performs better on the MovieLens-1M dataset. On the other side, the performance of PCC is worst on the MovieLens-1M dataset. The performance of the other similarity methods lies between the proposed ITR and PCC similarity method as shown in Fig. 3(e).

RMSE of our proposed similarity method and various other competitor methods are discussed by using the MovieLens-1M dataset. The experiment on this dataset returns the least RMSE as compared to other methods. After ITR, TMJ similarity performs better on the MovieLens-1M dataset. Similarly, on the other side, the performance of Jaccard is not satisfactory on the MovieLens-1M dataset. The performance of the other similarity methods lies between the proposed ITR and PCC similarity method as shown in fig. 2(e).

E. PRECISION ON DIFFERENT DATASETS

In this section, the results of a precision metric on the proposed similarity method and other competitor methods are discussed using five datasets, which we used for evaluation purposes. The below Figs. 4(a-e) show the result of precision on different similarity measures with different values of K neighbors.

1) PRECISION ON CIAODVD DATASET

Now the precision of the proposed similarity method and various other competitor methods are discussed using the CiaoDVD dataset. The experiment returns the maximum precision for the proposed ITR method as compared to other methods with different values of K. It is noted that the small value of K gives high precision values, while large values of K gives low precision values. After ITR, IPWR similarity performs better and on the other side, the performance of CPCC is worst on the CiaoDVD dataset. The performance of the other similarity methods lies between the proposed ITR and CPCC similarity method as shown in Fig. 4(a). The proposed ITR method gives the lowest precision when the value of K is greater than 35, which shows that ITR performs better with smaller values of K.

2) PRECISION ON EPINIONS DATASET

Now the Precision of the proposed similarity method and various other competitor methods are discussed by using the Epinions dataset. The experiment gives maximum precision value for the proposed method, as compared to other methods with varying values of K. After ITR, PIP similarity performs better. On the other side, the performance of Cosine is worst on the Epinions dataset. The performance of the other similarity methods lies between the proposed ITR and Cosine similarity method as shown in Fig. 4(b).

3) PRECISION ON FILMTRUST DATASET

Now we will discuss the precision of the proposed similarity method and various other competitor methods using the FilmTrust dataset. The experiment on our proposed method gives the maximum precision as compared to other methods with different values of K. After ITR, TMJ similarity performs better. On the worse side, the performance of PIP is worst on the FilmTrust dataset. The performance of the other similarity methods lies between the proposed ITR and PIP similarity method as shown in Fig. 4(c).

4) PRECISION ON MOVIELENS-100K DATASET

Now we will discuss the results of a precision metric on the proposed similarity method and various other similar methods are discussed using the MovieLens-100K dataset. The experiment yields the maximum precision value for the IPWR method as compared to other methods until the value of K reaches to 25. At $K = 25$, curves of IPWR and ITR intersect each other and subsequently, the precision value of ITR becomes superior to IPWR.

After the ITR, CPCC similarity performs better. On the worse side, the performance of PIP is worst on the MovieLens-100K dataset. The performance of the other similarity methods lies between the proposed ITR and PIP similarity method as shown in Fig. 4(d). Curves of all measures show that the precision of all methods decreases with an increase in K. However, the behavior of ITR is different, and the curve increases slightly for each increase in the value of K.

5) PRECISION ON MOVIELENS-1M DATASET

Now we will discuss the results of the precision metric on the proposed similarity method and various other similarity methods are discussed using the MovieLens-1M dataset. The experiments yield the maximum precision value for the IPWR method, after which comes proposed ITR method. Although, at the start, the performance of IPWR is slightly better. After the ITR, CPCC similarity performs better. On the worse side, the performance of Cosine and PIP is worst on this dataset. The performance of other similarity measures lies between the proposed ITR and PIP similarity method as shown in Fig. 4(e).

F. MOVIETWEETINGS DATASET

The MovieTweetings dataset is crawled from twitter and we downloaded it from GitHub [40]. This dataset consists of movie ratings in the range of [1–10], with 1 being the worst rating and 10 being the best rating of a movie. This dataset contains a total of 7,59,746 ratings given to 32,810 movies by 56,304 users. MovieTweetings dataset is highly sparse having sparsity of 99.90%. Due to the large rating scale of MovieTweetings dataset, we are analyzing the performance

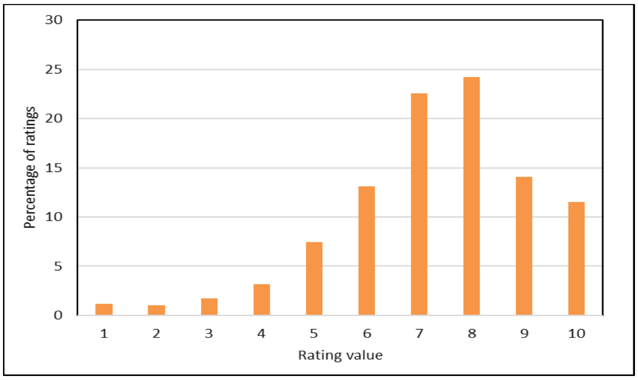
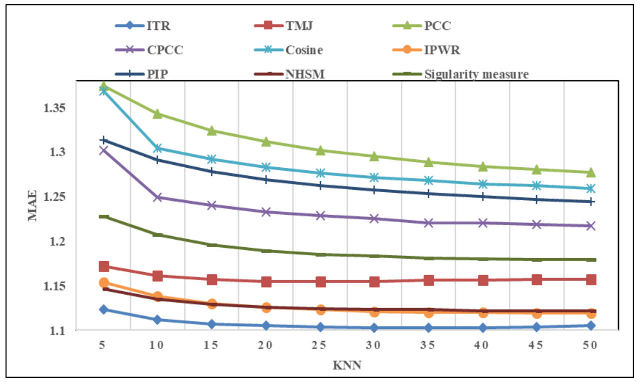
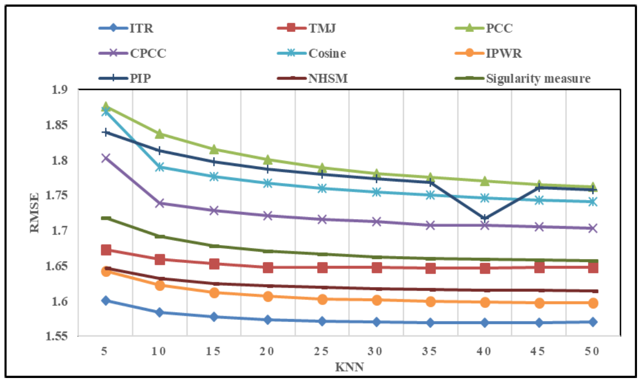


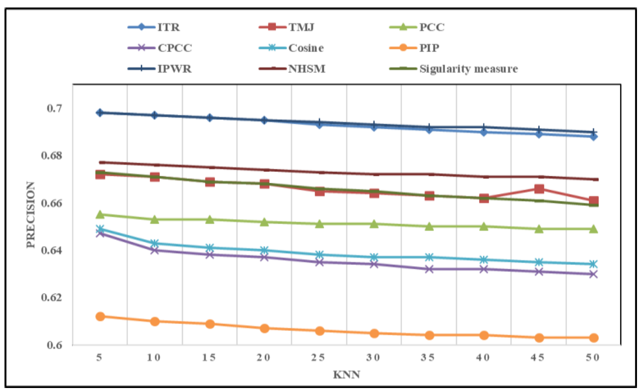
FIGURE 5. Percentage of rating distribution in the MovieTweetings dataset.



(a)



(b)



(c)

FIGURE 6. (a) MAE results on the MovieTweetings. (b) RMSE results on the MovieTweetings dataset. (c) Precision results on MovieTweetings.

of the proposed ITR method in this separate section. The rating distribution of MovieTweetings dataset is shown in Fig. 5. It shows that the most rated value in this dataset is 8 having an approximate percentage of 24%. Whereas, rating values 1 and 2 are the least rated value with an approximate percentage of 4%.

1) MAE ON MOVIETWEETINGS DATASET

Fig. 6(a) shows the results of MAE on the MovieTweetings dataset. It is visible that the proposed ITR method has the lowest MAE than all other methods. The performance of IPWR and NHSM comes after ITR. A small decrease in MAE occurs for all methods when KNN value lies in the interval 5 and 10. For all other values of KNN (i.e. from 15 to 50), MAE remains almost constant and no major or abrupt change is visible. PCC and Cosine give the worst performance.

2) RMSE ON MOVIETWEETINGS DATASET

Fig. 6(b) shows the results of RMSE on the MovieTweetings dataset. It is visible that the proposed ITR method has the lowest RMSE than all other methods. The performance of IPWR and NHSM comes after ITR. A small decrease in RMSE occurs for all methods when KNN value lies in the interval 5 and 10. For all other values of KNN (i.e. from 15 to 50), RMSE remains almost constant and no major or abrupt change is visible. Performance PCC and PIP is the worst.

3) PRECISION ON MOVIETWEETINGS DATASET

Fig. 6(c) shows the results of precision on the MovieTweetings dataset. It is visible that the proposed ITR method and IPWR have the highest Precision than all other methods. The results of both methods are very close to each other. Performance of NHSM, TMJ, and Singularity measure comes after ITR and IPWR. A small decrease in Precision occurs for all methods when KNN value increases in the interval 5 to 50. The performance of PIP is worst as compared to all other methods. Meanwhile, the curve formation of Cosine and CPCC is very similar.

4) F-SCORE OF REPORTED DATASETS

Keeping in view the experimental details presented in Section V and this section (Figs. 7(a-f)). We can conclude the following regarding the performance of the proposed ITR similarity on the reported datasets. On CiaoDVD, Epinions, and FilmTrust datasets, the performance of proposed ITR similarity is better as compared to its competitive methods in terms of MAE, RMSE, precision, and F-score metrics. However, its performance lagged in terms of F-score metric by CPCC and PIP methods on the CiaoDVD dataset. The sparsity of the CiaoDVD dataset is 99.90%. In the presence of this huge sparsity, finding co-ratings is a difficult matter.

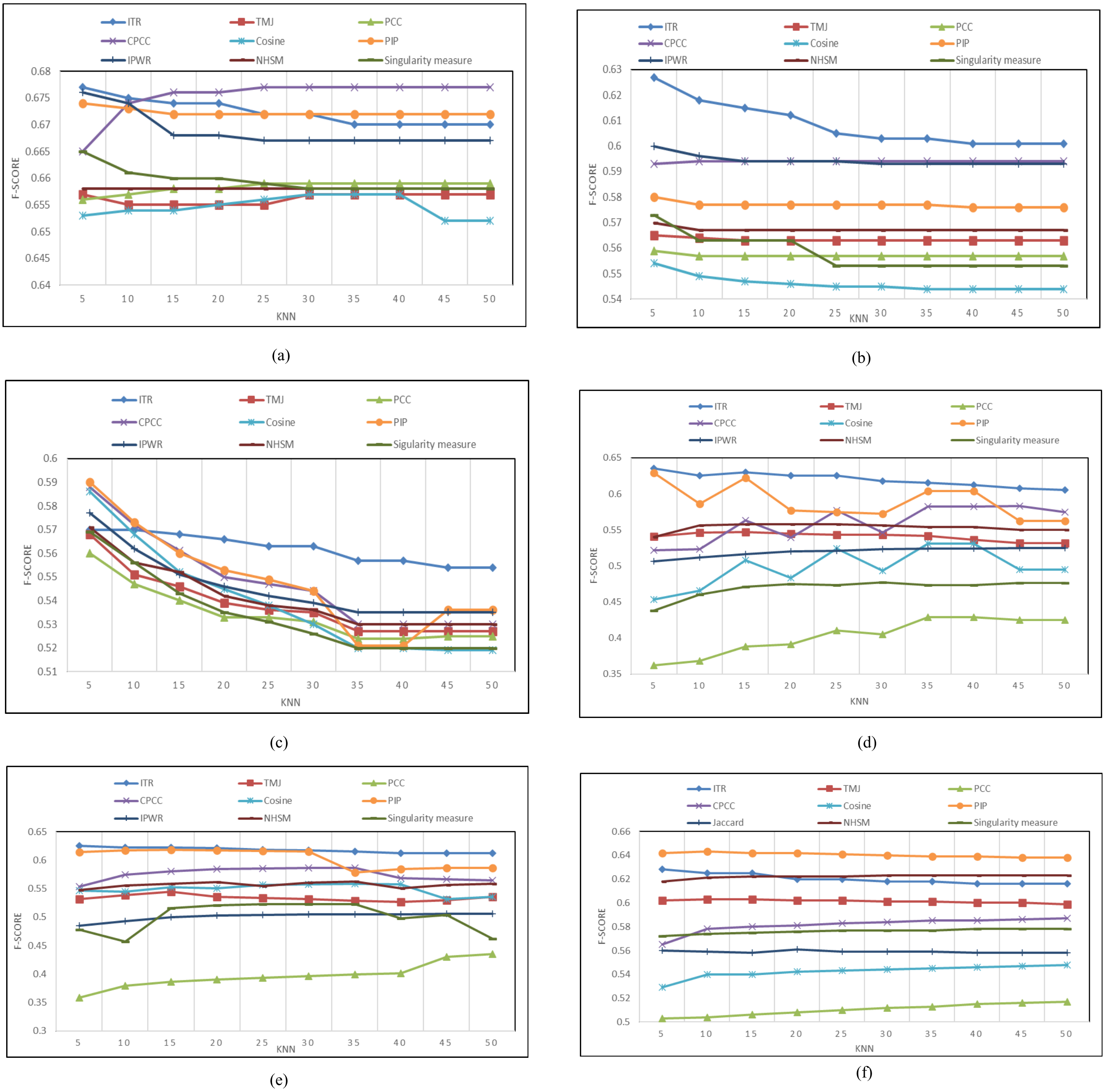


FIGURE 7. (a) F-score results on the CiaoDVD. (b) F-score results on the Epinions. (c) F-score results on the Filmtrust. (d) F-score results on the MovieLens-100K. (e) F-score results on the MovieLens-1M. (f) F-score results on the MovieTweetings.

Therefore, one possible reason for the good performance of the proposed ITR similarity on these datasets is the use of full ratings information and user rating preference (URP) behavior. Similarly, for the MovieLens-100K, MovieLens-1M, and MovieTweetings datasets, the proposed ITR similarity measure gives superior performance in terms of evaluation metrics as compared to its competitive methods. The reason for getting the superior performance of the proposed ITR similarity measure on these datasets is that it uses both rating and non-co-rating information of the users for a recommendation as compared to its competitive methods. However, the performance of the PIP method is very close to the proposed ITR similarity measure because it uses multiple information of the users called PIP factors for the same co-ratings of the users for recommendation purposes.

VI. CONCLUSION

In this paper, firstly, we reviewed the issues of the existing similarity methods of collaborative filtering. After that, we proposed an improved similarity measurement method based on the triangle similarity method to overcome the existing problems of similarity measure methods of collaborative filtering. In existing methods, like PCC, Cosine, PIP, and triangle similarity, recommendations are generated by utilizing only common ratings of users and not a complete set of ratings provided by the user. Therefore, we proposed a new similarity measurement method that overcomes the shortcomings of existing methods. The proposed method focuses on both common and non-common rating values of the user while producing recommendations. To improve accuracy, the proposed *ITR* similarity measure is complemented with the rating preference behavior of users. To evaluate its accuracy, we have performed extensive experiments on the six commonly used datasets in the field of CF. We noticed that the proposed ITR similarity measure performs better as compared to the existing similarity measures. In the future, we aim to apply this method in trust-based systems. Such systems employ social connection information and can help to mitigate more cold start and sparsity problem of CF systems. We also intend to use the proposed *ITR* similarity measure to measure distance in a clustering approach of the recommender systems.

DISCLOSURE OF POTENTIAL CONFLICTS OF INTEREST

All the authors declare no conflict of interest.

ACKNOWLEDGMENT

(Arta Iftikhar, Mustansar Ali Ghazanfar, Mubbashir Ayub, Zahid Mehmood, and Muazzam Maqsood contributed equally to this work.)

REFERENCES

[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl.-Based Syst.*, vol. 46, pp. 109–132, Jul. 2013.

[2] P. Moradi and S. Ahmadian, "A reliability-based recommendation method to improve trust-aware recommender systems," *Expert Syst. Appl.*, vol. 42, no. 21, pp. 7386–7398, Nov. 2015.

[3] N. Polatidis and C. K. Georgiadis, "Recommender systems: The importance of personalization in E-business environments," *Int. J. E-Entrepreneurship Innov.*, vol. 4, no. 4, pp. 32–46, Oct. 2013.

[4] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative filtering recommender systems," *Found. Trends Hum.-Comput. Interact.*, vol. 4, no. 2, pp. 81–173, Feb. 2010.

[5] J. A. Konstan and J. Riedl, "Recommender systems: From algorithms to user experience," *User Model. User-Adapted Interact.*, vol. 22, nos. 1–2, pp. 101–123, Apr. 2012.

[6] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Comput. Surv.*, vol. 47, no. 1, pp. 1–45, Jul. 2014.

[7] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, pp. 1–19, Oct. 2009.

[8] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, 1983.

[9] K. Lang, "NewsWeeder: Learning to filter netnews," in *Proc. 12th Int. Conf. Mach. Learn.*, Tahoe City, CA, USA, Jul. 1995, pp. 331–339.

[10] R. J. Mooney and L. Roy, "Content-based book recommending using learning for text categorization," in *Proc. ACM SIGIR Workshop Rec. Syst., Algorithms Eval.*, 1999, pp. 195–204.

[11] D. Anand and K. K. Bharadwaj, "Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5101–5109, May 2011.

[12] P. Jaccard, "Nouvelles recherches sur la distribution florale," *Bull. de la Societe Vaudoise des Sci. Naturelles*, vol. 44, no. 163, pp. 223-270, 1908.

[13] P. R. Rider, "On the distribution of the correlation coefficient in small samples," *Biometrika*, vol. 24, nos. 3–4, pp. 382–403, 1932.

[14] G. Qian, S. Sural, Y. Gu, and S. Pramanik, "Similarity between Euclidean and cosine angle distance for nearest neighbor queries," in *Proc. ACM Symp. Appl. Comput. (SAC)*, 2004, pp. 1232–1237.

[15] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Inf. Sci.*, vol. 178, no. 1, pp. 37–51, Jan. 2008.

[16] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 1999, pp. 230–237.

[17] M. Jamali and M. Ester, "TrustWalker: A random walk model for combining trust-based and item-based recommendation," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2009, pp. 397–406.

[18] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, "A collaborative filtering approach to mitigate the new user cold start problem," *Knowl.-Based Syst.*, vol. 48, pp. 204–217, Feb. 2012.

[19] J. Bobadilla, F. Ortega, and A. Hernando, "A collaborative filtering similarity measure based on singularities," *Inf. Process. Manage.*, vol. 48, no. 2, pp. 204–217, Mar. 2012.

[20] J. Bobadilla, A. Hernando, F. Ortega, and A. Gutiérrez, "Collaborative filtering based on significances," *Inf. Sci.*, vol. 185, no. 1, pp. 1–17, Feb. 2012.

[21] H. Ma, I. King, and M. R. Lyu, "Effective missing data prediction for collaborative filtering," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2007, pp. 39–46.

[22] A. Abdelwahab, H. Sekiya, I. Matsuba, Y. Horiuchi, and S. Kuroiwa, "Collaborative filtering based on an iterative prediction method to alleviate the sparsity problem," in *Proc. 11th Int. Conf. Inf. Integr. Web-Based Appl. Services (iiWAS)*, 2009, pp. 375–379.

[23] A. Szwabe, M. Ciesielczyk, and T. Janasiewicz, "Semantically enhanced collaborative filtering based on RSVD," in *Proc. Int. Conf. Comput. Collective Intell.*, 2011, pp. 10–19.

[24] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social recommendation using probabilistic matrix factorization," in *Proc. 17th ACM Conf. Inf. Knowl. Mining (CIKM)*, 2008, pp. 931–940.

[25] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *Inf. Retr.*, vol. 4, no. 2, pp. 133–151, 2001.

[26] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl, "Application of dimensionality reduction in recommender system—A case study," in *Proc. ACM WebKDD*, 2000, pp. 4–15.

[27] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, and Z. Chen, "Scalable collaborative filtering using cluster-based smoothing," in *Proc. 28th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2005, pp. 114–121.

[28] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web (WWW)*, 2001, pp. 285–295.

[29] A. B. Barragáns-Martínez, E. Costa-Montenegro, J. C. Burguillo, M. Rey-López, F. A. Mikic-Fonte, and A. Peleteiro, "A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition," *Inf. Sci.*, vol. 180, no. 22, pp. 4290–4311, Nov. 2010.

[30] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 143–177, Jan. 2004.

[31] J. B. Schafer, J. Konstan, and J. Riedl, "Recommender systems in e-commerce," in *Proc. 1st ACM Conf. Electron. Commerce*, 1999, pp. 58–166.

[32] S.-B. Sun, Z.-H. Zhang, X.-L. Dong, H.-R. Zhang, T.-J. Li, L. Zhang, and F. Min, "Integrating triangle and Jaccard similarities for recommendation," *PLoS ONE*, vol. 12, no. 8, Aug. 2017, Art. no. e0183570.

[33] W. Wang, G. Zhang, and J. Lu, "Collaborative filtering with entropy-driven user similarity in recommender systems," *Int. J. Intell. Syst.*, vol. 30, no. 8, pp. 854–870, Aug. 2015.

[34] N. Polatidis and C. K. Georgiadis, "A dynamic multi-level collaborative filtering method for improved recommendations," *Expert Syst.*, vol. 48, no. 16, pp. 100–110, Apr. 2016.

[35] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering," *Knowl.-Based Syst.*, vol. 56, pp. 156–166, Jan. 2014.

[36] K. G. Saranya and G. S. Sadasivam, ''Modified heuristic similarity measure for personalization using collaborative filtering technique,'' *Appl. Math. Inf. Sci.*, vol. 11, no. 1, pp. 307–315, Jan. 2017.

[37] S. Lee, ''Improving Jaccard index for measuring similarity in collaborative filtering,'' in *Proc. Int. Conf. Inf. Sci. Appl.*, 2017, pp. 799–806.

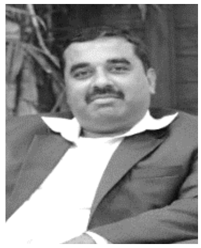
[38] M. Ayub, M. A. Ghazanfar, M. Maqsood, and A. Saleem, ''A Jaccard base similarity measure to improve performance of CF based recommender systems,'' in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2018, pp. 1–6.

[39] M. Ayub, M. A. Ghazanfar, Z. Mehmood, T. Saba, R. Alharbey, A. M. Munshi, and M. A. Alrige, ''Modeling user rating preference behavior to improve the performance of the collaborative filtering based recommender systems,'' *PLoS ONE*, vol. 14, no. 8, Aug. 2019, Art. no. e0220129.

[40] S. Dooms, T. De Pessemier, and L. Martens, ''Movietweetings: A movie rating dataset collected from Twitter,'' in *Proc. Workshop Crowdsourcing Hum. Comput. Rec. Syst. (CrowdRec RecSys)*, Oct. 2013, p. 43.

ARTA IFTIKHAR received the M.Sc. degree in software engineering from the University of Engineering and Technology (UET) Taxila, Pakistan, in 2013, where she is currently pursuing the Ph.D. degree in software engineering with the Department of Software Engineering. Her research interests include recommender systems, artificial intelligence, machine learning, digital image processing, computer vision, software engineering, and deep learning.

MUSTANSAR ALI GHAZANFAR received the Ph.D. degree from the University of Southampton, U.K. He is the author of more than 100 research articles. His research interests include machine learning and pattern recognition. He also served in the planning commission of Pakistan.



MUBBASHIR AYUB received the M.Sc. degree in software engineering from the University of Engineering and Technology (UET) Taxila, Pakistan, in 2012, where he is currently pursuing the Ph.D. degree in software engineering. He is also serving as an Assistant Professor at the Department of Software Engineering, UET Taxila, Pakistan. His research interests include recommender systems, machine learning, and semantic web.

ZAHID MEHMOOD received the B.S. degree (Hons.) in computer engineering from the COMSATS University of Sciences and Technology, Wah Campus, Pakistan, in 2009, the M.S. degree in electronic engineering with a specialization in signal and image processing from International Islamic University (IIU), Islamabad, Pakistan, in 2012, and the Ph.D. degree in computer engineering from the University of Engineering and Technology (UET) Taxila, Pakistan, in 2016. He published more than 70 publications in impact factor journals (ISI indexed) and international conferences. He is a Team Leader of the *FAMLIR (Forensic Analysis, Machine Learning, and Information Retrieval)* research group. His research interests include content-based image retrieval (CBIR), medical imaging, deep learning, image forensic, computer vision, and machine learning. He is also a reviewer for international journals and conferences, such as IEEE Access, *Pattern Recognition*, *Neural Computing and Applications*, *Neurocomputing*, *Journal of Electronic Imaging*, *Journal of Information Science*, *Computers & Electrical Engineering*, PAMI, and CVPR.

MUAZZAM MAQSOOD received the bachelor's and master's degrees from UET Taxila, and the Ph.D. degree from UET Taxila, in 2017. He is currently serving as an Assistant Professor at COMSATS University Islamabad, Attock Campus, Pakistan. He is the author of more than 20 publications. His research interests include medical imaging, machine learning, recommender systems, and image processing.