12-1-2023

Boosting the Item-Based Collaborative Filtering Model with Novel Similarity Measures

Hassan I. Abdalla
Zayed University, hassan.abdalla@zu.ac.ae

Ali A. Amer
Zayed University, amal.amer@zu.ac.ae

Yasmeen A. Amer
Thamar University

Loc Nguyen
Loc Nguyen's Academic Network

Basheer Al-Maqaleh
Thamar University

Recommended Citation

Abdalla, Hassan I.; Amer, Ali A.; Amer, Yasmeen A.; Nguyen, Loc; and Al-Maqaleh, Basheer, "Boosting the Item-Based Collaborative Filtering Model with Novel Similarity Measures" (2023). *All Works*. 5928.
https://zuscholars.zu.ac.ae/works/5928

International Journal of Computational Intelligence Systems (2023) 16:123
https://doi.org/10.1007/s44196-023-00299-2

RESEARCH ARTICLE

Boosting the Item-Based Collaborative Filtering Model with Novel Similarity Measures

Hassan I. Abdalla[1] · Ali A. Amer[1,2] · Yasmeen A. Amer[3] · Loc Nguyen[4] · Basheer Al-Maqaleh[3]

Received: 1 June 2023 / Accepted: 11 July 2023

Abstract

Collaborative filtering (CF), one of the most widely employed methodologies for recommender systems, has drawn undeniable attention due to its effectiveness and simplicity. Nevertheless, a few papers have been published on the CF-based item-based model using similarity measures than the user-based model due to the model's complexity and the time required to build it. Additionally, the substantial shortcomings in the user-based measurements when the item-based model is taken into account motivated us to create stronger models in this work. Not to mention that the common trickiest challenge is dealing with the cold-start problem, in which users' history of item-buying behavior is missing (i.e., new users) or items for which activity is not provided (i.e., new items). Therefore, our novel five similarity measures, which have the potential to solve sparse data, are developed to alleviate the impact of this important problem. Most importantly, a thorough empirical analysis of how the item-based model affects the CF-based recommendation system's performance has also been a critical part of this work, which presents a benchmarking study for thirty similarity metrics. The MAE, MSE, and accuracy metrics, together with fivefold cross-validation, are used to properly assess and examine the influence of all considered similarity measures using the Movie-lens 100 K and Film Trust datasets. The findings demonstrate how competitive the proposed similarity measures are in comparison to their alternatives. Surprisingly, some of the top "state-of-the-art" performers (such as SMD and NHSM) have been unable to fiercely compete with our proposed rivals when utilizing the item-based model.

Keywords Recommender systems · Collaborative filtering · Information retrieval · Similarity measure · Item-based model

1 Introduction

Currently, recommender systems are essential for speeding up internet users' searches for relevant content. In many various hubs, including e-commerce and e-libraries [1, 2], the recommender system (RS) is typically used as an information retrieval tool to address the problem of information overflow. Specifically speaking, the item recommendation process in RS uses three well-known techniques: content-based filtering (CBF), collaborative filtering (CF), and hybrid filtering. CF, on the one hand, is the most widely used method across many RS domains [3]. Locating the nearest neighbors using the kNN algorithm is the most important component of CF for enhancing recommendation quality. Thus, the similarity measure has undeniable impact on how well CF performs. To find the closest neighbors (i.e., comparable users/items), numerous different similarity measures have been proposed in the literature [3–6]. The target user's "preferences on different items" is predicted using the rating matrix and the similarity levels of the nearest neighbors. This idea serves as the foundation for the Top-N recommendation list [7].

On the other hand, CF is subdivided into memory-based and model-based models. In general, memory-based recommendations use similarity metrics and correlation to suggest items to a user for purchase based on past user-item ratings. Both user-based and item-based CF can be used to categorize memory-based literary works [8]. Since 1990, the significance of recommendations has increased, transforming

✉ Ali A. Amer
aliaaa2004@yahoo.com; aliaaa2004@gmail.com

[1] College of Technological Innovation, Zayed University, Dubai, UAE

[2] Computer Science Department, Taiz University, Taiz, Yemen

[3] Faculty of Computers and Informatics, Thamar University, Thamar, Yemen

[4] Loc Nguyen's Academic Network, Long Xuyen, Vietnam

Published online: 29 July 2023

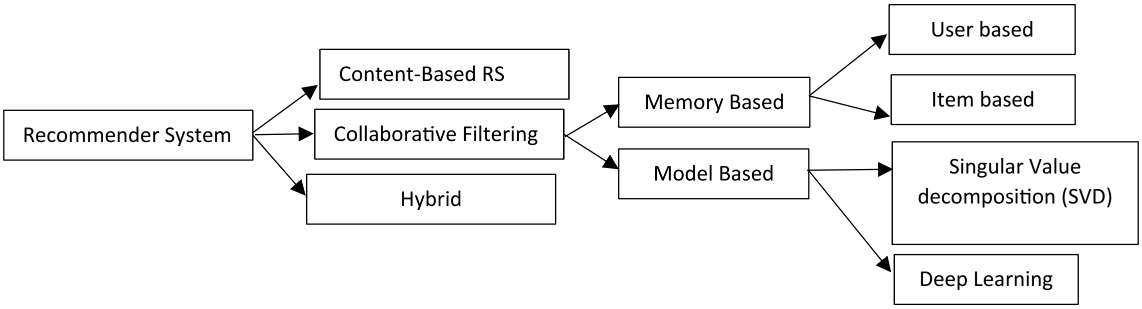

Fig. 1 The three basic categories of RS methods are content-based, collaborative filtering, and hybrid. The two types of collaborative filtering are memory-based and model-based. Deep learning models and SVD methods make up the two subcategories of model-based methods. User-based and item-based methods are two categories of memory-based techniques

RS into a separate field of study. The categories for the RS approaches are displayed in Fig. 1.

Our work in this paper will center on collaborative filtering (CF), which is based on the idea that we can forecast active users' ratings for new products by taking into account previous users' ratings of related products [4]. CF approaches are subdivided further into neighborhood or memory-based and model-based models. Memory-based models enable predictions of ratings for new goods based on ratings provided by the target user's closest friends (or neighbors). On the other hand, a linear model is used to train latent components that explain rating on a lower dimensional latent space in the model-based method applying SVD, the first subcategory of model-based CF [1]. Despite the RS effectiveness of SVD algorithms, deep learning models, the second subtype of model-based CF, have been taken into consideration by the research community and the industry to learn latent features from users and items [1, 7]. However, despite the advantages of deep learning, nearest neighbor and similarity measures are still widely used in commercial recommender systems [16, 22] because of their simplicity. This fact drives us to continue developing similarity measures for the improvement of item-based model performance.

While the item-based model (CF) has a great impact on CF's performance, the vast majority of CF efforts, however, have refrained from looking into the item-based model due to the model's complexity and the lengthy time needed to implement the model using similarity measure(s). As a result, a thorough analysis of the effect of the item-based model on CF performance is currently lacking in the CF literature. The item-based CF, originally introduced by GroupLens Research Group [9], is a more practical model for an item recommendation, although it is more complex than the user-based model. The item-based model's straightforward premise is that people frequently like analogs of products they previously loved. To forecast the target item [10] for the active user, the item-based model uses the degrees of similarity between things (s). Overall, the item-based model has the capacity to make more accurate recommendations than the user-based model, which is one of its main advantages [9–11].

In addition to the reasons mentioned above for choosing the item-based model, this work's primary objective is to introduce new similarity measures that have been empirically demonstrated to be promising and to investigate how effective the similarity measures are on the movie lens-100 K and Film Trust datasets using the item-based model. Due to other more significant factors, such as the item-based model's straightforward implementation, the ability to handle cold starts, the ability to scale well with the correlated items [8], as well as the glaring lack of a substantial investigation into the item-based model's impact on CF performance in the literature, we are also motivated to consider studying the impact of the item-based model on CF performance using a dozen similarity measures.

Simply put, the following contributions help this work bridge some of the gaps in the literature: (1) Develop five new item-based similarity measures to properly address the cold-start problem using the item-based model. These measurements are of simplistic and straightforward design, yet expertly drawn, and experimental evidence demonstrates that they are highly effective in identifying precise forecasts and recommendations. (2) Some important aspects are implicitly taken into account during the formulation of the proposed similarity measures without the use of weighting factors to consistently increase RS performance and quality. Decreasing or eliminating the reliance on co-rated items, complete utilization of all rated and non-rated items, and simultaneous treatment of similarity as symmetric and asymmetric are a few of these aspects.

Experimentally, the effectiveness of recommender systems based on item-based CF is improved using our proposed similarity measurements. (3) Leveraging the item-based model with fivefold cross-validation to conduct a comprehensive empirically driven comparative analysis on two commonly known datasets (Movielens-100 K and Film Trust) for 30 similarity measures. To demonstrate the effectiveness and precision of our proposed methods, we also provide comprehensive empirical results. So, one key goal is to use the item-based model to benchmark these measures on the selected datasets. According to experimental findings on benchmark data sets, our proposed measures—in particular, NPMS—have a competitive prediction and recommendation quality. Empirically, NPMS addresses the data sparseness successfully when compared to the existing representative measures.

The remaining work is scheduled to be completed as follows: the most pertinent works are covered in Sect. 2. All considered measures, including the new measures and their rivals, are introduced in Sect. 3. The experimental results are shown and concisely discussed in Sect. 4. Finally, our work is concluded by looking at future work in Sect. 5.

2 Related Work

Researchers have been attracted to the memory-based collaborative filtering (CF) whose ultimate goal is to improve the presentation of the recommender system by addressing its flaws. The general goal of the memory-based CF is to ascertain the user's preferences by taking into account their ratings of various goods. In reality, CF has become a popular strategy for the recommendation process due to the growing amounts of data on the internet. Recommendation uses the two-phase item-based CF process, which involves calculating the similarity between items and users and predicting ratings using similar items/users. However, in sparse datasets, the quality of the recommendations has been severely diminished [12]. The top-k most similar items/users are still found using a variety of similarity measures in both user-based and item-based CF, and prediction methodologies are used to rate the prediction. The top N of the recommended things is then generated. Numerous research studies that use new, combined, or modified similarity measurements have been published in the literature to enhance CF performance. In this work, On the one hand, we used item-based models in our empirical investigation to explore as many potential similarity measures as we could to gauge their actual influence on CF. Contrarily, we theoretically rely on the most recent researches that address the item-based CF.

For instance, a modified Bhattacharyya coefficient was proposed in [8] to lessen the impact of data sparsity. The user-user similarity, which served as a weight in the item-based CF, was determined using the proposed similarity measure. When user-user similarity was discovered by the proposed modified similarity measure, the experimental study on the MovieLens dataset indicated a modest improvement in item-based CF. The memory-based CF techniques, such as the KNNBasic, KNNBaseline, KNNWithMeans, SVD, and SVD++, were evaluated in [13] to run RS by (1) determining a user-item similarity matrix and prediction matrix. (2) Applying the model-based CF technique via Co-Clustering to find a user-item similarity matrix and prediction matrix. The results showed that when applied to MovieTrust datasets using cross-validation (CV = 5, 10, and 15), the CF run, utilizing the K-NNBaseline technique, there is reduced error rate.

On the same theme, a technique to incorporate item-variance weighting into item-based CF was proposed in [14] to improve CF performance. A time-related correlation degree was given and employed to create a time-aware similarity computation in the meantime to improve predicted accuracy. This action was performed to recognize the relationship between two items and reduce the weight of items graded over an extended period of time. According to their experimental findings, the proposed technique performed better than conventional item-based CFs. On the other hand, formal concept analysis (FCA), a mathematical model-based CF, was utilized in [15] to improve the rating prediction of the unknown consumers. In order to estimate the rating of the unidentified users, the FCA approach was applied using Boolean matrix factorization (i.e., the best formal notions). The proposed method was tested on the Movielens dataset using the item-based CF technique, and the results were good comparing with rivals.

The improvement of the conventional similarity measures that are frequently practiced on the item-based CF was the main goal of the work of [16]. The three used conventional measures—cosine-based similarity, Pearson correlation similarity, and modified cosine similarity—were considered to be balanced using an algorithm. The algorithm was also contrasted with the currently used conventional measures. The proposed approach provided a better item-based CF for the recommendation systems than the previous one, according to the results. In [17], an item-based CF algorithm was suggested. It comprised of an item-based prediction method with the new neighbor selection approach and an item-based similarity measure based on fuzzy sets [Kullback–Leibler (KL) divergence was employed as a similarity measure]. The proposed method offered improved prediction and suggestion quality, according to the results.

Meanwhile, the improvement of rating prediction was the focus of the effort in [18]. To obtain a more precise calculation of the weight between the active user and each of their neighbors, an objective function was created and improved to be employed in the rating prediction phase. The results demonstrated that the proposed strategy performed better than others, particularly when there were a small and medium number of carefully chosen neighbors. A user-internal similarity-based recommendation algorithm (UISRC) was subsequently proposed in [19]. The recommendation score was modified to produce score predictions and suggestions using the internal similarity of users. The findings demonstrated that UISRC might outperform conventional CF and MD algorithms in terms of recommendation accuracy and diversity.

In addition, to make an effective recommendation, Bag et al. [20] proposed a new structural similarity named RJaccard, which not only took into account co-rated cases but also emphasized the importance of the number of non-co-rated cases. They tested the measure over Movielens-100 K dataset, and results show the proposed measure promising. Gazdar et al. [21] proposed a new similarity measure OS, which further added the influence of the proportion of co-rated cases. Similarly, to calculate item similarity, Wang et al. [22] established a divergence-based similarity measure and further incorporated the effect of the fraction of co-rated cases. Finally, in [23], it was suggested to use a new item-based Kullback–Leibler (KL) divergence-based CF method in which the item similarity measure based on hazy sets and the item-based prediction method (ISP) make up the proposed algorithm. The results showed that the proposed method is promising.

3 Methodology

3.1 Motivation

In actuality of RS, the efficacy of both neighborhood-based and model-based techniques is significantly affected by the similarity measure (either user-based or item-based). More specifically, the similarity method can be considered as a key element of CF success given the impact that similarity has on the accuracy of its recommendations. In this regard, our experimental findings show that most of the existing similarity measures miss the effect of sparse data on prediction outcomes and fail to address the cold-start issue when the item-based model is taken into account. Thus, the ultimate goal of this research is to add new simple-yet-effective similarity measures to the item-based CF literature. These similarity measures are mathematically designed to significantly improve the CF's performance through implicit addressing the data sparsity challenge (i.e., cold-start problem). Moreover, the examination of the influence of the item-based model, in contrast to the user-based CF, has not been given convincing attention. Due to its intricate complexity and prolonged run time compared to the user-based model, only a small number of studies (which are already mentioned in related work section) have been undertaken to explore the item-based model using similarity measures for collaborative filtering. In the meantime, this paper aims to give a thorough empirical investigation on the similarity measures for collaborative filtering using the item-based model in addition to proposing new similarity measures.

Furthermore, the comparative evaluation findings reported in this article could be a greatly helpful guide for researchers in this field as they select the best similarity measure for the item-based model. Most crucially, our empirical study is the first to fully address the item-based model utilizing over 30 similarity metrics. The item-based empirical work, which addresses this research vacuum, is a more difficult challenge than the user-based model and has not yet gotten enough attention in the literature. In conclusion, even though a variety of similarity metrics have been offered from various angles to improve the accuracy of similarity results, more research is still required, according to our review of the literature and experimental study, to produce highly competitive similarity measures.

3.2 The Proposed Similarity Measures

The creation of an active user's neighborhood depends critically on calculation of similarity, which also influences how accurate the CF-based recommendation approach is. To improve prediction accuracy, this part describes five novel similarity measures that try to account for actual similarity between items. Technically speaking, the advancement of memory-based CF approaches' accuracy is the main goal of the proposed work. Memory-based CF approaches search for comparable users/items for an active user/item using a database of user evaluations of items. The ratings of items not viewed by the active user are then predicted using ratings provided by comparable users. The effectiveness of the similarity measure has a significant impact on the recommendations' accuracy.

In a trial to find the best-performing CF using the item-based model, we develop a new measure that is based on the ideologies of the TF and IDF. Given two rating vectors $u_1 = (r_{11}, r_{12}, \ldots, r_{1n})$ and $u_2 = (r_{21}, r_{22}, \ldots, r_{2n})$ of user 1 and user 2, respectively, in which some r_{ij} can be missing (empty), this measure can be seen as a quasi-IDF, which is defined in Eq. 1, as follows:

$$\mathrm{quasiIDF}(u_1, u_2) = 1 - \frac{\left(\sum_{j \in I_1 \backslash I_2} r_{1j}\right)\left(\sum_{j \in I_2 \backslash I_1} r_{2j}\right)}{\left(\sum_{j \in I_1} r_{1j}\right)\left(\sum_{j \in I_2} r_{2j}\right)} \tag{1}$$

The notation "\" denotes the complement operator. Similarly, following the ideology of the term "frequency (TF), we define the so-called "quasi-TF" in Eq. 2, as follows:

$$\mathrm{quasiTF}(u_1, u_2) = \frac{\left(\sum_{j \in I_1 \cap I_2} r_{1j}\right)\left(\sum_{j \in I_1 \cap I_2} r_{2j}\right)}{\left(\sum_{j \in I_1} r_{1j}\right)\left(\sum_{j \in I_2} r_{2j}\right)} \tag{2}$$

Note, the notation "∩" denotes the intersection operator in the set theory. This new measure, which is called a quasi-TF-IDF (QTI), is the product of the quasi-TF and the quasi-IDF, according to Eq. 3.

$$\begin{aligned}\mathrm{QTI}(u_1, u_2) &= \mathrm{quasiTF}(u_1, u_2) \times \mathrm{quasiIDF}(u_1, u_2)\\ &= \frac{\left(\sum_{j \in I_1 \cap I_2} r_{1j}\right)\left(\sum_{j \in I_1 \cap I_2} r_{2j}\right)}{\left(\sum_{j \in I_1} r_{1j}\right)\left(\sum_{j \in I_2} r_{2j}\right)} \times \left(1 - \frac{\left(\sum_{j \in I_1 \backslash I_2} r_{1j}\right)\left(\sum_{j \in I_2 \backslash I_1} r_{2j}\right)}{\left(\sum_{j \in I_1} r_{1j}\right)\left(\sum_{j \in I_2} r_{2j}\right)}\right)\end{aligned} \tag{3}$$

In the meantime, we also derive a Jaccard-combined QTI, which derives the QTIJ similarity measure according to Eq. 4.

$$\begin{aligned}\mathrm{QTIJ}(u_1, u_2) = &\frac{\left(\sum_{j \in I_1 \cap I_2} r_{1j}\right)\left(\sum_{j \in I_1 \cap I_2} r_{2j}\right) * \mathrm{Jaccard}(u_1, u_2)}{\left(\sum_{j \in I_1} r_{1j}\right)\left(\sum_{j \in I_2} r_{2j}\right)}\\ &\times \left(1 - \frac{\left(\sum_{j \in I_1 \backslash I_2} r_{1j}\right)\left(\sum_{j \in I_2 \backslash I_1} r_{2j}\right) \times (1 - \mathrm{Jaccard}(u_1, u_2))}{\left(\sum_{j \in I_1} r_{1j}\right)\left(\sum_{j \in I_2} r_{2j}\right)}\right)\end{aligned} \tag{4}$$

Based on the Jaccard measure, Eq. 4 is written simply in Eq. 5 as follows:

$$\mathrm{QTIJ}(u_1, u_2) = \mathrm{quasiTFJ}(u_1, u_2) \times \mathrm{quasiIDFJ}(u_1, u_2) \tag{5}$$

where,

$$\mathrm{quasiTFJ}(u_1, u_2) = \frac{\left(\sum_{j \in I_1 \cap I_2} r_{1j}\right)\left(\sum_{j \in I_1 \cap I_2} r_{2j}\right) \times \mathrm{Jaccard}(u_1, u_2)}{\left(\sum_{j \in I_1} r_{1j}\right)\left(\sum_{j \in I_2} r_{2j}\right)} \tag{5.1}$$

$$\begin{aligned}&\mathrm{quasiIDFJ}(u_1, u_2)\\ &= 1 - \frac{\left(\sum_{j \in I_1 \backslash I_2} r_{1j}\right)\left(\sum_{j \in I_2 \backslash I_1} r_{2j}\right) \times (1 - \mathrm{Jaccard}(u_1, u_2))}{\left(\sum_{j \in I_1} r_{1j}\right)\left(\sum_{j \in I_2} r_{2j}\right)}\end{aligned} \tag{5.2}$$

The following example demonstrates how the QTIJ measure works. Given two rating vectors $u_1 = (r_{11} = 2,\ r_{12} = 5,\ r_{13} = 7,\ r_{14} = 8,\ r_{15} = ?,\ r_{16} = 9)$ and $u_2 = (r_{21} = 9,\ r_{22} = ?,\ r_{23} = ?,\ r_{24} = 6,\ r_{25} = 5,\ r_{26} = 1)$, we calculate QTIJ as an example. Of course, we have $I_1 = \{1, 2, 3, 4, 6\}$ and $I_2 = \{1, 4, 5, 6\}$. We also have the following:

$$\begin{aligned}\mathrm{quasiTFJ}(u_1, u_2) &= \frac{(2 + 8 + 9)(9 + 6 + 1)}{(2 + 5 + 7 + 8 + 9)(9 + 6 + 5 + 1)}\\ &\times \frac{|I_1 \cap I_2|}{|I_1 \cup I_2|} = \frac{19 \times 16}{31 \times 21} \times \frac{3}{6} \cong 0.23,\end{aligned}$$

$$\begin{aligned}\mathrm{quasiIDFJ}(u_1, u_2) &= 1 - \frac{(5 + 7) * 5}{(2 + 5 + 7 + 8 + 9)(9 + 6 + 5 + 1)}\\ &\times \left(1 - \frac{|I_1 \cap I_2|}{|I_1 \cup I_2|}\right) = 1 - \frac{12 \times 5}{31 \times 21}\left(1 - \frac{3}{6}\right) \cong 0.95.\end{aligned}$$

This implies:

$$\begin{aligned}\mathrm{QTIJ}(u_1, u_2) &= \mathrm{quasiTFJ}(u_1, u_2)\\ &\times \mathrm{quasiIDFJ}(u_1, u_2) = 0.23 \times 0.95 \cong 0.22.\end{aligned}$$

However, the Jaccard measure, which is a powerful similarity measure, concentrates on the rating status rather than the magnitude rating values like cosine, for instance. So, we have also tried to resolve this limitation by putting forth the so-called numerical proximity similarity measure (NPSM). NPSM maintains the benefit of the Jaccard measure while focusing on magnitude rating values. In other words, NPSM adds the cardinalities of item sets and the sums of rating values as well. Equation 6 provides the following NPSM formulation.

$$\mathrm{NPSM}(u_1, u_2) = \frac{|I_1 \cap I_2| \sum_{j \in I_1 \cap I_2} r_{1j} r_{2j}}{|I_1| \sum_{j \in I_1} r_{1j} + |I_2| \sum_{j \in I_2} r_{2j}}. \tag{6}$$

Observe that $|I_1 \cap I_2|$ represents the total number of products rated by users 1 and 2, $|I_1|$ represents the total number of items rated by user 1, and $|I_2|$ represents the total number of items rated by user 2. It is obvious that NPSM is an intriguing advanced cosine measure version with Jaccard support. As such, we are also motivated to prove that NPSM differs greatly from the proposed cosine-Jaccard combination, which is described in Eq. 7, as follows:

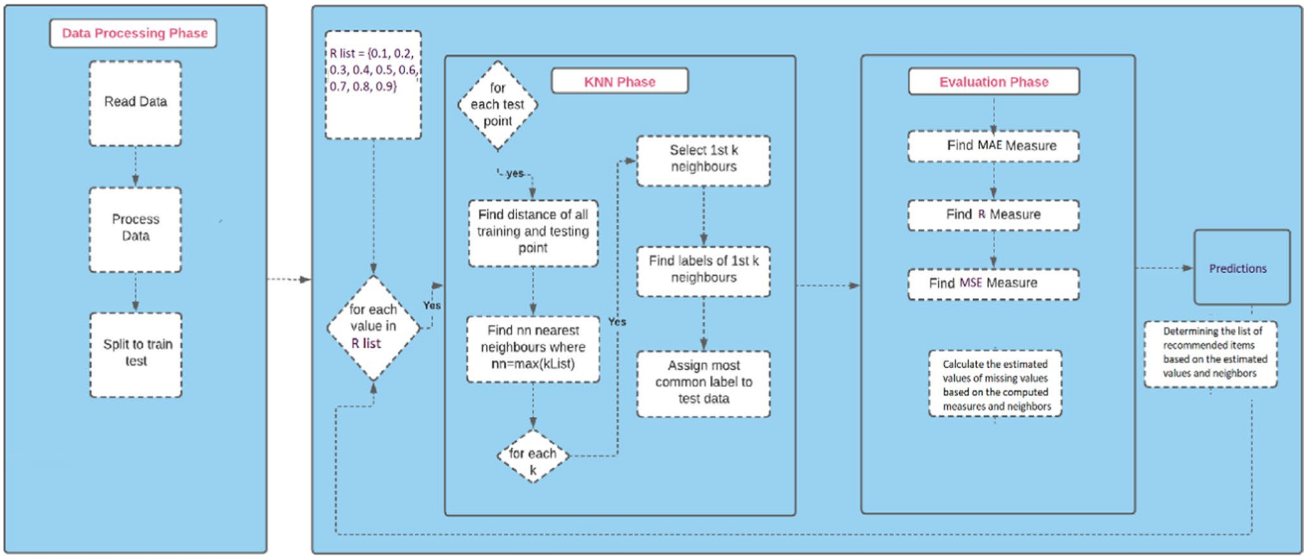
Data Processing Phase
Read Data
Process Data
Split to train test
R list = {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}
for each value in R list
Yes
KNN Phase
for each test point
yes
Find distance of all training and testing point
Find nn nearest neighbours where nn=max(kList)
for each k
Yes
Select 1st k neighbours
Find labels of 1st k neighbours
Assign most common label to test data
Evaluation Phase
Find MAE Measure
Find R Measure
Find MSE Measure
Calculate the estimated values of missing values based on the computed measures and neighbors
Predictions
Determining the list of recommended items based on the estimated values and neighbors

Fig. 2 Experimental design scheme

$$\mathrm{COSINEJ}\left(u_1,u_2\right) = \mathrm{cosine}\left(u_1,u_2\right) \times \mathrm{Jaccard}\left(u_1,u_2\right) = \frac{\sum_{j\in I_1\cap I_2} r_{1j}r_{2j}}{\sqrt{\sum_{j\in I_1}\left(r_{1j}\right)^2}\sqrt{\sum_{j\in I_2}\left(r_{2j}\right)^2}} \times \frac{\left|I_1 \cap I_2\right|}{\left|I_1 \cup I_2\right|} \tag{7}$$

Our experimental study has shown that our NPSM, in numerous cases, is significantly more straightforward and effective than Jaccard, and our proposed COSINEJ as well. To understand the work of NPSM, the next example is provided. Given two rating vectors $u_1=(r_{11}=2,\ r_{12}=5,\ r_{13}=7,\ r_{14}=8,\ r_{15}=?,\ r_{16}=9)$ and $u_2=(r_{21}=9,\ r_{22}=?,\ r_{23}=?,\ r_{24}=6,\ r_{25}=5,\ r_{26}=1)$, we calculate NPSM as an example. Of course, we have $I_1=\{1,\ 2,\ 3,\ 4,\ 6\}$ and $I_2=\{1,\ 4,\ 5,\ 6\}$, then NPSM is computed as follows:

$$\mathrm{NPSM}\left(u_1,u_2\right) = \frac{\left|I_1 \cap I_2\right|(2\times 9+8\times 6+9\times 1)}{\left|I_1\right|(2+5+7+8+9)+\left|I_2\right|(9+6+5+1)} = \frac{3\times 75}{5\times 31+4\times 21} \cong 0.94.$$

In conclusion, we propose the measure families QTI, QTIJ, and NPSM in this research. The accuracy of similarity is further enhanced by NPSM, which implicitly takes into account the impact of rating quantity. Moreover, The NPSM equation has been designed to adjust prediction outcomes by taking into account more rating data. Every vector in the user-based rating matrix is a user rating vector by default. Therefore, the equations for these measures are not altered in terms of their semantics when the user-based rating matrix is transformed into an item-based rating matrix, in which every vector is an item rating vector. According to experimental findings on benchmark data sets, our suggested measures—in particular, NPMS—have competitive prediction and recommendation quality, and successfully address the data sparseness issue when compared to existing representative measures.

3.3 The Evaluation Mechanism

Each dataset used in our experiments (using Java) was divided into five folders, each containing a training set and a testing set. The training set and testing set are independent sets located in the same folder. The proportion of the testing set to the overall dataset has been calculated using the specified testing parameter r, which has a range of 0.1–0.9. Figure 1 shows the proposed strategy that we adopted for our experimental study. In the case of Movielens-100 K, for example, the training set has 90,000 ratings assuming $r=0.1$, while the testing set has 10,000 ratings (10% × 100,000), indicating that the testing set covers 10% of the dataset. In our experimental setting, parameter r has nine different values that are 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9, respectively. We discovered that the smaller r is, the larger the training set is, and the more accurate measures are.

It is important to note that the "*r*" parameter is used to test how well all similarity measures, including the ones we present, perform while taking into account how much training data sparsity they can handle. Therefore, the more instances of r where the relative similarity measure X surpasses its rivals, the more likely it is that the X measure is robustly applicable to sparse datasets. This method of testing has allowed us to show that some of the similarity metrics we proposed to solve the cold start issue are indeed effective. (See Fig. 2).

We used both datasets—the Movielens-100 K and Film Trust datasets—to investigate how the item-based model affected CF performance. Amazingly, our proposed similarity measures in this work are promising. Assuming that the active item has k neighbors, Eq. 8 computes the missing value raj as follows:

$$r_{aj} = \overline{r}_a + \frac{\sum_{i=1}^{k} \left(r_{ij} - \overline{r}_i\right) sim\left(r_a, r_i\right)}{\sum_{i=1}^{k} \left|\mathrm{sim}\left(r_a, r_i\right)\right|} \tag{8}$$

where $\overline{r}_a$ and $\overline{r}_i$ are, respectively, the mean values of r_a and r_i. The accompanying Tables 1, 2, 3, 4, 5 and 6 show the performance averaged across 5 folders for the tested measures. The results of the associated assessment metric for each of the following values of r = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9 are displayed in each of the accompanying Tables 1, 2, 3, 4, 5, 6. The last column of each Table, which gives the averaged results of the pertinent measure over all values of r, displays the five best values, which are shown in grey-shaded columns.

3.4 The Rivalry Similarity Measures

Through our GitHub repository, the public can view most of the included rivalry similarity metrics.[1]

3.5 Evaluation Metrics

We have used the standard metrics for CF algorithms, namely the mean absolute error (MAE) (Eq. 9), the mean squared error (MSE) (Eq. 10), and the R metric (Eq. 10), to analyze the correlation between the prediction vector v' and the genuine vector v.

$$\mathrm{MAE} = \frac{1}{n} \sum_{j=1}^{n} \left| v_j{}' - v_j \right| \tag{9}$$

When the MAE is smaller, the approach operates more effectively. The number n represents the expected item count, and the letters v_j and $v_j{}'$ stand for the true and predicted ratings of item j, respectively. On the other hand, the MSE is determined as follows given the prediction vector v' and true vector (tested vector) v:

$$\mathrm{MSE} = \frac{1}{n} \sum_{j=1}^{n} \left(v_j{}' - v_j \right)^2 \tag{10}$$

When the MSE is lower, the algorithm performs better and the measurement is more accurate. Finally, the accuracy metric R is shown in Eq. 11. The measure is better if R is higher.

$$R = \frac{\sum_{j=1}^{n} \left(v_j{}' - \overline{v'} \right) \left(v_j - \overline{v} \right)}{\sqrt{\sum_{j=1}^{n} \left(v_j{}' - \overline{v'} \right)^2} \sqrt{\sum_{j=1}^{n} \left(v_j - \overline{v} \right)^2}} \tag{11}$$

where $\overline{v'}$ and $\overline{v}$ are the mean values of the tested and predicted items, respectively.

$$\overline{v'} = \frac{1}{n} \sum_{j=1}^{n} v_j{}'$$

$$\overline{v} = \frac{1}{n} \sum_{j=1}^{n} v_j$$

Overall, three metrics, namely the MAE, MSE, and R, enable a precise assessment of the algorithms. A similarity measure is preferred when its MAE and RMSE values are lower and when its R values are higher.

4 Results and Discussion

Over both datasets, the findings of each similarity measure for the MAE, MSE, and R metrics are displayed in the following Tables 1, 2, 3, 4, 5 and 6. According to experimental findings on benchmark data sets, our suggested measures—in particular, NPSM—have competitive prediction and recommendation quality, and successfully address the data sparseness issue when compared to existing representative measures.

According to Table 1, the top six similarity measures, in order of averaged MAE, are: NPSM, NHSM, CosineJ, HSMDJ, TMJ, and QTIJ. On the other hand, the first three similarity measures, in order of r parameter performance (Figs. 3, 4), are: NPSM, QTIJ, and NHSM. The r parameters specify the sparsity level of the dataset. We can note that the proposed measures are more stable and better than all considered measures when the sparsity level of the data is very high 10–90, 20–80, 30–70, 40–60 splits, and even 50–50. These three measures (QTIJ, NPSM, and NHSM) also showed competitive averaged performance against other rivals as seen in Fig. 4.

[1] https://github.com/aliamer/Study-the-Impact-of-an-Item-based-Model-on-CF-Performance-Using-a-Dozen-Similarity-Measures/blob/main/similarity%20measures%20list%20used%20in%20item%20based%20model.pdf.

Table 1 MAE results—Movielens-100 K

	$r=0.1$	$r=0.2$	$r=0.3$	$r=0.4$	$r=0.5$	$r=0.6$	$r=0.7$	$r=0.8$	$r=0.9$	MAE
Cosine	0.7427	0.7449	0.7456	0.7497	0.7541	0.7573	0.7653	0.7818	0.8177	0.7621
COJ	0.7317	0.7346	0.7354	0.7398	0.7450	0.7495	0.7600	0.7831	0.8597	0.7598
CON	0.7468	0.7505	0.7539	0.7604	0.7681	0.7772	0.7905	0.8188	0.8813	0.7830
COD	0.7894	0.7937	0.7980	0.8031	0.8099	0.8183	0.8290	0.8494	0.9098	0.8222
CosineJ	0.7293	0.7324	0.7333	0.7379	0.7434	0.7479	0.7580	0.7784	0.8421	0.7558
Pearson	0.7447	0.7531	0.7591	0.7676	0.7774	0.7897	0.8057	0.8314	0.8326	0.7845
WPC	0.7348	0.7393	0.7438	0.7514	0.7613	0.7744	0.7933	0.8245	0.9088	0.7812
SPC	0.7434	0.7511	0.7569	0.7649	0.7743	0.7862	0.8022	0.8287	0.8341	0.7824
PearsonJ	0.7263	0.7336	0.7398	0.7493	0.7605	0.7742	0.7943	0.8259	0.8966	0.7778
Jaccard	0.7300	0.7330	0.7339	0.7385	0.7440	0.7485	0.7592	0.7815	0.8566	0.7583
MSD	0.7420	0.7443	0.7450	0.7491	0.7535	0.7566	0.7646	0.7811	0.8388	0.7638
SRC	0.7220	0.7271	0.7286	0.7334	0.7481	0.7634	0.8054	0.9082	1.0260	0.7958
PSS	0.7353	0.7379	0.7387	0.7429	0.7481	0.7529	0.7619	0.7836	0.8489	0.7611
NHSM	0.7219	0.7251	0.7260	0.7309	0.7376	0.7436	0.7566	0.7843	0.8583	0.7538
BCF	0.7822	0.7832	0.7845	0.7884	0.7934	0.7972	0.8034	0.8193	0.8568	0.8009
SMTP	0.7465	0.7487	0.7488	0.7517	0.7562	0.7589	0.7667	0.7833	0.8425	0.7670
PC	0.7887	0.7931	0.7988	0.8035	0.8099	0.8191	0.8309	0.8518	0.9128	0.8231
MMD	0.7802	0.7879	0.7935	0.8058	0.8201	0.8334	0.8562	0.8931	0.9723	0.8380
CjacMD	0.7685	0.7753	0.7800	0.7915	0.8044	0.8172	0.8406	0.8803	0.9851	0.8269
TMJ	0.7298	0.7329	0.7337	0.7383	0.7438	0.7483	0.7589	0.7811	0.8562	0.7581
Feng	0.7303	0.7333	0.7339	0.7385	0.7439	0.7485	0.7590	0.7824	0.8597	0.7588
Mu	0.7371	0.7422	0.7455	0.7526	0.7611	0.7718	0.7880	0.8153	0.8836	0.7774
TA	0.7399	0.7421	0.7429	0.7471	0.7517	0.7551	0.7634	0.7809	0.8399	0.7625
TAN	0.7341	0.7365	0.7397	0.7454	0.7532	0.7601	0.7724	0.7959	0.8599	0.7663
SMD	0.7412	0.7438	0.7446	0.7489	0.7537	0.7569	0.7648	0.7792	0.8174	0.7611
HSMD	0.7376	0.7403	0.7408	0.7451	0.7500	0.7541	0.7642	0.7871	0.8702	0.7654
HSMDJ	0.7262	0.7293	0.7302	0.7349	0.7411	0.7467	0.7597	0.7876	0.8681	0.7582
QTI	0.7233	0.7265	0.7278	0.7326	0.7400	0.7473	0.7639	0.7994	0.8852	0.7606
QTIJ	0.7160	0.7200	0.7224	0.7284	0.7380	0.7486	0.7701	0.8121	0.8976	0.7614
NPSM	0.7203	0.7237	0.7251	0.7303	0.7374	0.7444	0.7220	0.7193	0.8617	0.7426

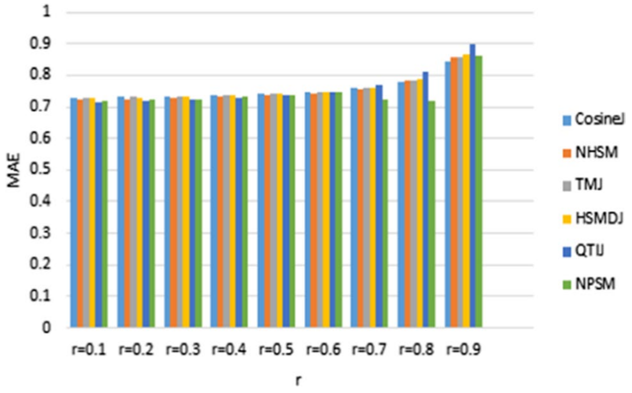

Fig. 3 Top six similarity measure based on MAE—Movielens

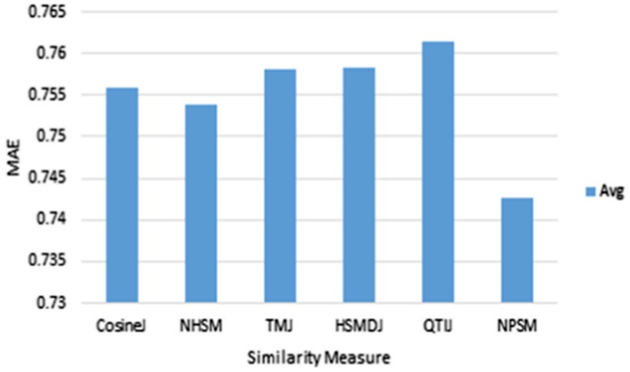

Fig. 4 Top six similarity measure in average—Movielens 100 K

According to Table 2, the top six similarity measures, in order of averaged MSE, are: NPSM, NHSM, CosineJ, HSMDJ, TMJ, and Jaccard. On the other hand, the first three similarity measures, in order of r parameter performance

(Figs. 5, 6), are: NPSM, CosineJ, and NHSM. We can note that the proposed measures are more stable and better than all considered measures when the sparsity level of the data is very high 10–90, 20–80, 30–70, 40–60 splits, and even 50–50. These measures also showed competitive averages against other rivals as seen in Fig. 6.

Table 2 MSE results—Movielens-100 K

	$r=0.1$	$r=0.2$	$r=0.3$	$r=0.4$	$r=0.5$	$r=0.6$	$r=0.7$	$r=0.8$	$r=0.9$	Avg
Cosine	0.8843	0.8936	0.8981	0.9057	0.9166	0.9266	0.9475	0.9942	1.0900	0.9396
COJ	0.8592	0.8698	0.8746	0.8830	0.8957	0.9094	0.9369	1.0012	1.2213	0.9390
CON	0.8887	0.9016	0.9128	0.9257	0.9441	0.9689	1.0055	1.0852	1.2773	0.9899
COD	0.9833	0.9972	1.0087	1.0202	1.0371	1.0618	1.0916	1.1513	1.3414	1.0769
CosineJ	0.8536	0.8650	0.8703	0.8790	0.8923	0.9064	0.9328	0.9890	1.1730	0.9290
Pearson	0.8835	0.9067	0.9223	0.9393	0.9609	0.9932	1.0340	1.1037	1.1207	0.9849
WPC	0.8661	0.8776	0.8904	0.9054	0.9273	0.9605	1.0076	1.0889	1.3465	0.9855
SPC	0.8807	0.9020	0.9171	0.9331	0.9538	0.9851	1.0257	1.0969	1.1240	0.9798
PearsonJ	0.8456	0.8651	0.8809	0.9001	0.9246	0.9597	1.0095	1.0931	1.3039	0.9758
Jaccard	0.8550	0.8664	0.8717	0.8804	0.8937	0.9083	0.9364	0.9987	1.2143	0.9361
MSD	0.8829	0.8923	0.8968	0.9042	0.9151	0.9250	0.9458	0.9925	1.1624	0.9463
SRC	0.8399	0.8540	0.8608	0.8697	0.9036	0.9459	1.0573	1.3442	1.6989	1.0415
PSS	0.8668	0.8771	0.8822	0.8904	0.9032	0.9181	0.9425	1.0023	1.1940	0.9418
NHSM	0.8377	0.8497	0.8552	0.8649	0.8812	0.9000	0.9342	1.0097	1.2228	0.9283
BCF	0.9846	0.9910	0.9979	1.0058	1.0187	1.0317	1.0492	1.0950	1.2090	1.0425
SMTP	0.8932	0.9027	0.9059	0.9108	0.9219	0.9306	0.9513	0.9984	1.1733	0.9542
PC	0.9933	1.0081	1.0228	1.0323	1.0476	1.0729	1.1025	1.1602	1.3501	1.0877
PIP	0.8567	0.8672	0.8740	0.8821	0.8970	0.9151	0.9469	1.0193	1.2345	0.9436
MMD	0.9835	1.0058	1.0227	1.0526	1.0878	1.1241	1.1852	1.2898	1.5165	1.1408
CjacMD	0.9534	0.9726	0.9883	1.0149	1.0473	1.0819	1.1454	1.2585	1.5677	1.1144
TMJ	0.8547	0.8660	0.8713	0.8800	0.8933	0.9077	0.9356	0.9977	1.2130	0.9354
Feng	0.8565	0.8672	0.8718	0.8805	0.8937	0.9079	0.9358	1.0007	1.2230	0.9374
Mu	0.8689	0.8832	0.8927	0.9061	0.9243	0.9516	0.9917	1.0637	1.2684	0.9722
TA	0.8780	0.8873	0.8920	0.8995	0.9109	0.9216	0.9432	0.9918	1.1659	0.9433
TAN	0.8682	0.8787	0.8883	0.9005	0.9182	0.9382	0.9703	1.0363	1.2269	0.9584
SMD	0.8808	0.8911	0.8958	0.9039	0.9151	0.9248	0.9445	0.9824	1.0895	0.9364
HSMD	0.8725	0.8829	0.8871	0.8949	0.9072	0.9201	0.9466	1.0111	1.2521	0.9527
HSMDJ	0.8465	0.8582	0.8637	0.8729	0.8880	0.9057	0.9399	1.0169	1.2468	0.9376
QTI	0.8409	0.8528	0.8590	0.8691	0.8871	0.9094	0.9533	1.0509	1.2959	0.9465
QTIJ	0.8267	0.8410	0.8499	0.8630	0.8869	0.9180	0.9752	1.0906	1.3345	0.9540
NPSM	0.8347	0.8472	0.8537	0.8643	0.8815	0.9029	0.9301	1.0047	1.1092	0.9142

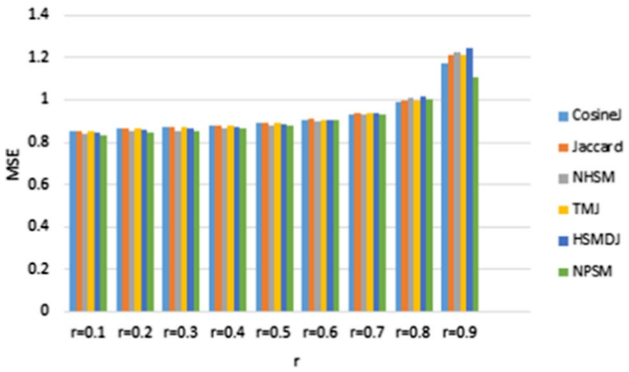

Fig. 5 Top six similarity measure based on MSE—Movielens 100 K

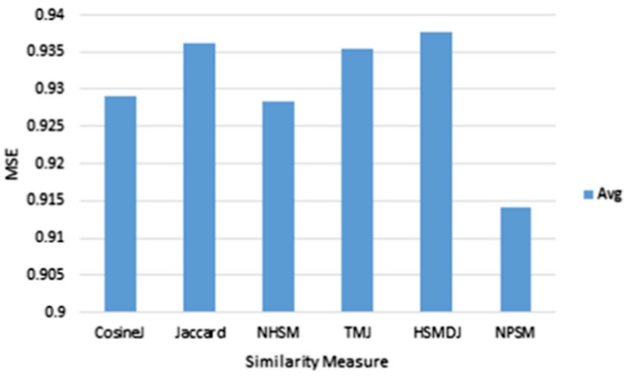

Fig. 6 Top six similarity measure in average—Movielens 100 K

According to Table 3, the top six similarity measures, in order of averaged R, are: NHSM, Jaccard, NPSM, HSMDJ, QTI, and QTIJ. On the other hand, the first three similarity measures, in order of *r* parameter performance (Figs. 7, 8), are: Jaccard, NPSM, and NHSM. These measures also showed competitive averaged against other rivals as seen in Fig. 8.

Table 3 R results—Movielens-100 K

	$r=0.1$	$r=0.2$	$r=0.3$	$r=0.4$	$r=0.5$	$r=0.6$	$r=0.7$	$r=0.8$	$r=0.9$	Avg
Cosine	0.4217	0.3790	0.3817	0.3808	0.3776	0.3763	0.3652	0.3415	0.3154	0.3710
COJ	0.4004	0.3983	0.3998	0.3993	0.3948	0.3885	0.3722	0.3392	0.2620	0.3727
CON	0.4342	0.3930	0.3932	0.3889	0.3773	0.3629	0.3400	0.2979	0.2273	0.3572
COD	0.4403	0.3918	0.3895	0.3832	0.3672	0.3541	0.3311	0.2968	0.2243	0.3531
CosineJ	0.4061	0.3996	0.4027	0.4019	0.3961	0.3893	0.3741	0.3429	0.2661	0.3754
Pearson	0.4429	0.3911	0.3942	0.3860	0.3738	0.3604	0.3364	0.3002	0.2785	0.3626
WPC	0.4549	0.4073	0.4105	0.4024	0.3893	0.3755	0.3488	0.3054	0.2412	0.3706
SPC	0.4459	0.3936	0.3966	0.3888	0.3770	0.3642	0.3405	0.3029	0.2774	0.3652
PearsonJ	0.4306	0.4196	0.4176	0.4072	0.3916	0.3762	0.3477	0.3036	0.2266	0.3690
Jaccard	0.4444	0.3992	0.4015	0.4008	0.3950	0.3877	0.3717	0.3397	0.2637	0.3782
MSD	0.4226	0.3787	0.3819	0.3811	0.3779	0.3767	0.3658	0.3419	0.2714	0.3664
SRC	0.3411	0.3483	0.3321	0.3266	0.3026	0.2769	0.2276	0.1673	0.1544	0.2752
PSS	0.4402	0.3980	0.3997	0.3991	0.3951	0.3882	0.3740	0.3406	0.2649	0.3778
NHSM	0.4579	0.4137	0.4136	0.4133	0.4040	0.3907	0.3699	0.3289	0.2547	0.3830
BCF	0.3978	0.3557	0.3546	0.3548	0.3457	0.3440	0.3333	0.3206	0.2815	0.3431
SMTP	0.4145	0.3683	0.3711	0.3738	0.3698	0.3707	0.3589	0.3370	0.2674	0.3591
PC	0.4067	0.3980	0.3960	0.3894	0.3712	0.3560	0.3314	0.2965	0.2238	0.3521
PIP	0.4071	0.4050	0.4050	0.4050	0.3982	0.3894	0.3686	0.3314	0.2540	0.3737
MMD	0.3547	0.3469	0.3444	0.3358	0.3249	0.3155	0.2986	0.2747	0.2440	0.3155
CjacMD	0.3615	0.3545	0.3518	0.3431	0.3329	0.3222	0.3020	0.2698	0.2080	0.3162
TMJ	0.4045	0.3994	0.4018	0.4010	0.3953	0.3881	0.3722	0.3402	0.2640	0.3741
Feng	0.4012	0.3988	0.4008	0.4002	0.3953	0.3886	0.3723	0.3393	0.2620	0.3732
Mu	0.4064	0.3958	0.3993	0.3913	0.3814	0.3662	0.3416	0.3031	0.2288	0.3571
TA	0.4249	0.3811	0.3841	0.3835	0.3800	0.3780	0.3671	0.3418	0.2702	0.3679
TAN	0.4321	0.3890	0.3865	0.3825	0.3707	0.3592	0.3393	0.3075	0.2466	0.3570
SMD	0.4246	0.3811	0.3836	0.3822	0.3791	0.3773	0.3668	0.3529	0.3158	0.3727
HSMD	0.4321	0.3895	0.3915	0.3916	0.3873	0.3835	0.3680	0.3362	0.2515	0.3701
HSMDJ	0.4132	0.4072	0.4082	0.4078	0.4003	0.3890	0.3680	0.3289	0.2553	0.3753
QTI	0.4179	0.4152	0.4128	0.4117	0.4015	0.3855	0.3595	0.3141	0.2459	0.3738
QTIJ	0.4305	0.4251	0.4164	0.4143	0.3990	0.3777	0.3481	0.3008	0.2400	0.3724
NPSM	Excel	0.4136	0.4128	0.4123	0.4023	0.3874	0.3665	0.3261	0.2546	0.3766

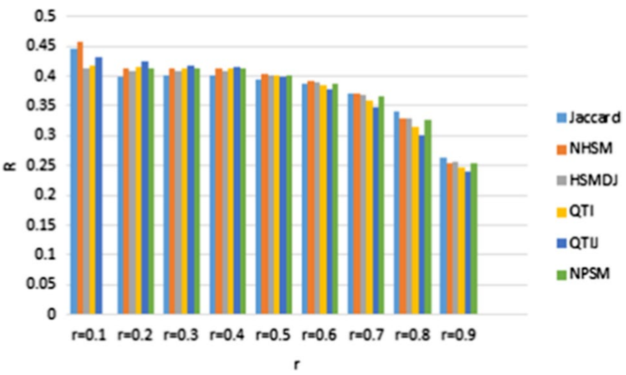

Fig. 7 Top six similarity measure based on R—Movielens 100 K

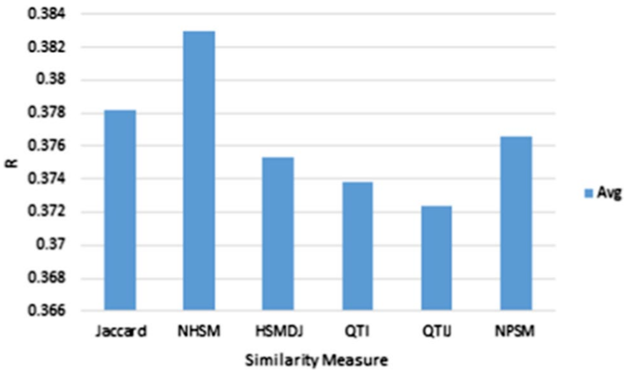

Fig. 8 Top six similarity measure in average—Movielens 100 K

According to Table 4, the top five similarity measures, in order of averaged MAE, are: Mu, NPSM, NHSM, TMJ, CosineJ, and CjacMD. On the other hand, the first three similarity measures, in order of *r* parameter performance (Figs. 9, 10), are: NPSM, Mu, and CosineJ.

Table 4 MAE results—Film Trust

	$r=0.1$	$r=0.3$	$r=0.5$	$r=0.7$	$r=0.9$	MAE
Cosine	0.6281	0.6274	0.6300	0.6367	0.6678	0.6380
COJ	0.6245	0.6261	0.6316	0.6456	0.6877	0.6431
CON						
COD	0.7479	0.7623	0.7828	0.8128	0.8596	0.7931
CosineJ	0.6207	0.6229	0.6274	0.6359	0.6728	0.6359
Pearson	0.6140	0.6109	0.6157	0.6313	0.7458	0.6435
WPC	0.6084	0.6061	0.6119	0.6292	0.7415	0.6394
SPC	0.6124	0.6093	0.6142	0.6302	0.7432	0.6419
PearsonJ	0.6077	0.6056	0.6111	0.6284	0.7405	0.6387
Jaccard	0.6224	0.6251	0.6311	0.6458	0.6867	0.6422
MSD	0.6268	0.6262	0.6291	0.6362	0.6679	0.6372
PIP	0.6223	0.6236	0.6277	0.6370	0.6805	0.6382
PSS	0.6259	0.6263	0.6294	0.6376	0.6767	0.6392
NHSM	0.6183	0.6210	0.6255	0.6355	0.6841	0.6369
BCF	0.6437	0.6416	0.6459	0.6576	0.6974	0.6572
SMTP	0.7947	0.8033	0.8003	0.7809	0.7071	0.7773
PC	0.8266	0.8393	0.8363	0.8438	0.8635	0.8419
MMD	0.9297	0.9259	0.9260	0.9333	0.9572	0.9344
CjacMD	0.6249	0.6248	0.6280	0.6358	0.6677	0.6362
TMJ	0.6201	0.6225	0.6271	0.6358	0.6728	0.6357
Feng	0.6245	0.6259	0.6312	0.6455	0.6876	0.6429
Mu	0.6116	0.6088	0.6137	0.6248	0.6789	0.6276
TA	0.6261	0.6257	0.6287	0.6360	0.6684	0.637
TAN	0.6241	0.6239	0.6273	0.6378	0.7077	0.6442
SMD	0.6230	0.6253	0.6312	0.6458	0.6865	0.6424
HSMD	0.6292	0.6290	0.6331	0.6472	0.6905	0.6458
HSMDJ	0.6221	0.6252	0.6310	0.6462	0.6965	0.6442
QTI	0.6225	0.6260	0.6318	0.6477	0.7072	0.647
QTIJ	0.6203	0.6249	0.6313	0.6520	0.7232	0.6503
QuasiIDFJ	0.6287	0.6285	0.6325	0.6457	0.6865	0.6444
QuasiTFJ	0.6207	0.6251	0.6311	0.6479	0.7073	0.6464
NPSM	0.61	0.619	0.622	0.632	0.677	0.632

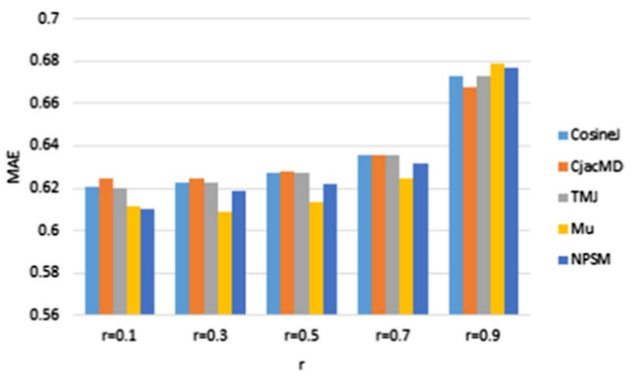

Fig. 9 Top five similarity measure based on MAE—Film Trust

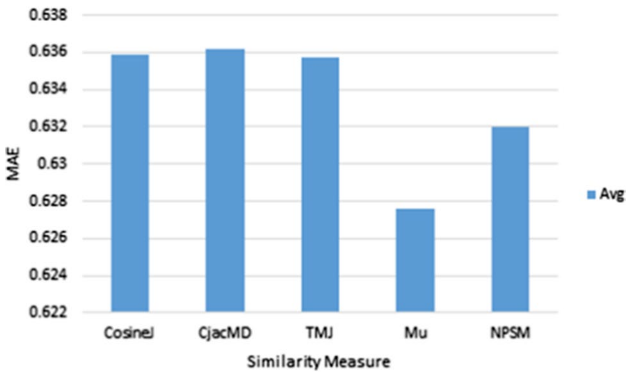

Fig. 10 Top five similarity measure in average—Film Trust

On the other hand, according to Table 5, the top five similarity measures, in order of averaged MSE, are: Mu, NPSM, CjacMD, WPC, and TMJ. On the other hand, the first three similarity measures, in order of *r* parameter performance (Figs. 11, 12), are: Mu, NPSM, and CjacMD.

Table 5 MSE results—Film Trust

	$r=0.1$	$r=0.3$	$r=0.5$	$r=0.7$	$r=0.9$	Avg
Cosine	0.6829	0.6895	0.6965	0.7140	0.7880	0.7142
COJ	0.6788	0.6897	0.7032	0.7427	0.8483	0.7325
CON						
COD	0.9124	0.9600	1.0050	1.0815	1.2015	1.0321
CosineJ	0.6714	0.6835	0.6945	0.7147	0.8019	0.7132
Pearson	0.6464	0.6450	0.6567	0.6938	0.9431	0.717
WPC	0.6381	0.6380	0.6513	0.6903	0.9335	0.7102
SPC	0.6435	0.6422	0.6541	0.6917	0.9362	0.7135
PearsonJ	0.6409	0.6405	0.6521	0.6915	0.9316	0.7113
Jaccard	0.6767	0.6895	0.7044	0.7441	0.8459	0.7321
MSD	0.6789	0.6859	0.6937	0.7126	0.7880	0.7118
PIP	0.6698	0.6797	0.6913	0.7146	0.8210	0.7153
PSS	0.6779	0.6856	0.6950	0.7157	0.8106	0.717
NHSM	0.6677	0.6805	0.6914	0.7152	0.8342	0.7178
BCF	0.7165	0.7178	0.7302	0.7611	0.8677	0.7587
SMTP	1.0999	1.1309	1.1320	1.0963	0.8927	1.0704
PC	1.1559	1.1859	1.1627	1.1762	1.2192	1.18
MMD	1.3710	1.3774	1.3751	1.3948	1.4590	1.3955
CjacMD	0.6772	0.6853	0.6926	0.7122	0.7875	0.711
TMJ	0.6691	0.6817	0.6933	0.7142	0.8017	0.712
Feng	0.6806	0.6908	0.7034	0.7429	0.8482	0.7332
Mu	0.6407	0.6405	0.6530	0.6818	0.8107	0.6853
TA	0.6772	0.6847	0.6928	0.7121	0.7892	0.7112
TAN	0.6707	0.6789	0.6882	0.7162	0.8727	0.7253
SMD	0.6775	0.6898	0.7046	0.7440	0.8453	0.7322
HSMD	0.6870	0.6941	0.7055	0.7435	0.8521	0.7364
HSMDJ	0.6758	0.6901	0.7046	0.7459	0.8735	0.738
QTI	0.6767	0.6920	0.7065	0.7501	0.9043	0.7459
QTIJ	0.6787	0.6959	0.7104	0.7639	0.9514	0.7601
QuasiIDFJ	0.6863	0.6932	0.7043	0.7422	0.8448	0.7342
QuasiTFJ	0.6766	0.6930	0.7072	0.7516	0.9049	0.7467
NPSM	0.6724	0.6869	0.6968	0.7184	0.8350	0.7056

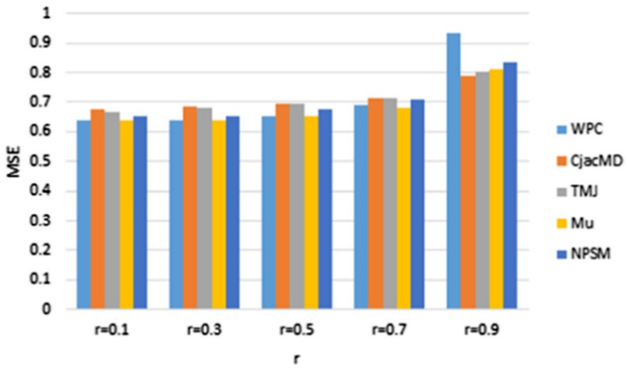

Fig. 11 Top Five Similarity Measure based on MSE– Film Trust

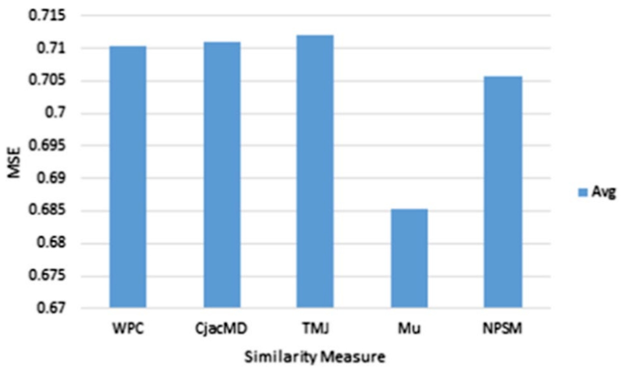

Fig. 12 Top Five Similarity Measure in Average – Film Trust

Finally, according to Table 6, the top five similarity measures, in order of averaged R, are: CosineJ, QTIJ, QuasiTFJ, NPSM, and NHSM. On the other hand, the first three similarity measures, in order of *r* parameter performance (Figs. 13, 14), are: QTIJ, QuasiTFJ, and CosineJ.

Table 6 R results—Film Trust

	r = 0.1	*r* = 0.3	*r* = 0.5	*r* = 0.7	*r* = 0.9	Avg
Cosine	0.2002	0.2201	0.2205	0.2083	0.1814	0.2061
COJ	0.2181	0.2326	0.2322	0.2094	0.1665	0.2118
CON						
COD	0.2564	0.2404	0.2139	0.1654	0.1017	0.1956
CosineJ	0.2262	0.2343	0.2299	0.2135	0.1729	0.2125
Pearson	0.2242	0.2433	0.2313	0.1977	0.0925	0.1978
WPC	0.2352	0.2527	0.2432	0.2047	0.0962	0.2064
SPC	0.2258	0.2459	0.2341	0.1990	0.0938	0.1997
PearsonJ	0.2503	0.2567	0.2498	0.2051	0.0962	0.2116
Jaccard	0.2234	0.2335	0.2316	0.2100	0.1667	0.213
MSD	0.2005	0.2207	0.2213	0.2087	0.1815	0.2065
PIP	0.2269	0.2396	0.2395	0.2178	0.1711	0.219
PSS	0.2146	0.2337	0.2317	0.2136	0.1732	0.2134
NHSM	0.2309	0.2416	0.2365	0.2131	0.1613	0.2167
BCF	0.1953	0.2242	0.2250	0.2119	0.1835	0.208
SMTP	0.1418	0.1378	0.1434	0.1012	0.1304	0.1309
PC	0.1438	0.1427	0.1359	0.1314	0.0929	0.1293
MMD	0.1962	0.2213	0.2222	0.2067	0.1797	0.2052
CjacMD	0.2053	0.2253	0.2263	0.2103	0.1806	0.2096
TMJ	0.2249	0.2342	0.2293	0.2133	0.1728	0.2149
Feng	0.2171	0.2331	0.2324	0.2094	0.1666	0.2117
Mu	0.2108	0.2347	0.2335	0.2126	0.1448	0.2073
TA	0.1991	0.2219	0.2217	0.2105	0.1815	0.2069
TAN	0.2233	0.2428	0.2374	0.2074	0.1328	0.2087
SMD	0.2219	0.2331	0.2312	0.2103	0.1670	0.2127
HSMD	0.2007	0.2241	0.2254	0.2039	0.1627	0.2034
HSMDJ	0.2267	0.2364	0.2353	0.2078	0.1560	0.2124
QTI	0.2305	0.2392	0.2389	0.2043	0.1485	0.2123
QTIJ	0.2515	0.2478	0.2406	0.1948	0.1412	0.2152
QuasiIDFJ	0.2010	0.2241	0.2277	0.2074	0.1673	0.2055
QuasiTFJ	0.2429	0.2422	0.2395	0.2045	0.1490	0.2156
NPSM	0.2275	0.2391	0.2343	0.2125	0.1623	0.2161

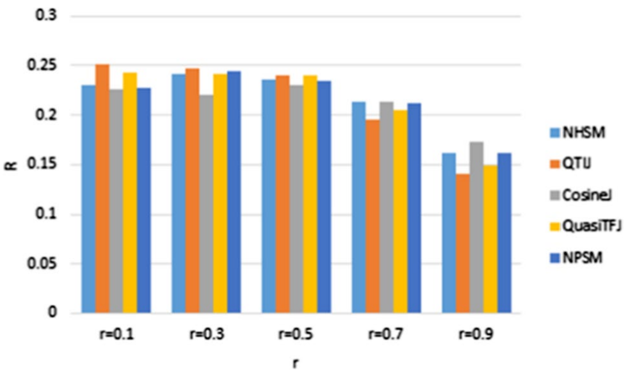

Fig. 13 Top five similarity measure based on R—Film Trust

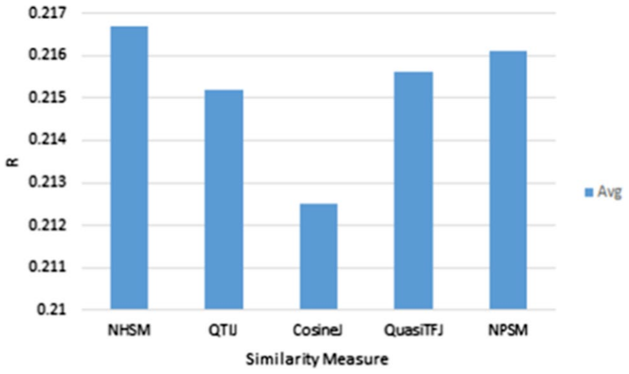

Fig. 14 Top five similarity measure in average—Film Trust

Table 7 Similarity measure ranking

Similarity measure (SM)	Top-rank 1	Top-rank 2	Top-rank 3	Averaged rank	Rank order
NPSM	0.34	0.33	0.17	0.28	**1**
NHSM	0.17	0.34	0.0	0.18	**2**
CosineJ	0.17	0.0	0.34	0.18	**2**
QTIJ	0.0	0.17	0.0	0.057	4
Jaccard	0.0	0.16	0.0	0.056	5
Mu	0.34	0.0	0.0	0.11	3
CjacMD	0.0	0.0	0.17	0.057	4
QuasiTFJ	0.0	0.0	0.17	0.057	4

The bold values indicate the best performing models in order

To sum up, it is worth stressing that the CF literature has been creating similarity metrics that are built on several notions to adequately represent the concept of similarity between users/items. A pair of users' preferences for a given item have been taken into consideration when computing similarity for some of these measures, just like our suggested measures in this article. To do this, we looked at how strongly a pair of users liked or disapproved of a certain item in relation to the majority. To verify the effectiveness of all similarity measures at varied rates of training and testing data, several scenarios/cases (r values) are addressed. According to experimental findings, currently available similarity measurements don't adequately account for the true similarity of items. Additionally, results showed that the computation of similarity over the majority of these instances has significantly improved with the proposed similarity measures.

4.1 Similarity Measure Ranking

The following formula is proposed to rank the similarity measures (SM) depending on how well it performed across both datasets (the formula can be expanded to be over N datasets):

$$\text{Rank(SM)} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}(P(SM)++)}{M \times N} \tag{12}$$

When the SM satisfies the desired performance regarding the target evaluation metric, a temporary counter, denoted by P(SM), is continually increased by one; otherwise, P is set to zero. N and M, respectively, stand for the number of datasets and the number of evaluation metrics (see Table 7).

The Top-Rank I (where I falls in 1, 2, 3) suggests the number of evaluation metrics on which the similarity measure has achieved the first (including the second and third) order concerning the performance (MAE, MSE, and R) over both datasets. The statistics in Table 7 can lead us to the conclusion that NPSM, NHSM, and CosineJ all performed best on average across both datasets, making these similarity measures quite promising, with NPSM being the top one.

4.2 Work Limitations

While we admit that some recent research on model-based recommender systems has focused on stochastic gradient descent [24, 25] and deep learning methods [26], it is worth stressing that the nearest neighbor and similarity measures are still highly preferred in the industry [2, 6–8], and, due to their simplicity, they are also still widely employed in commercial recommender systems [16, 22]—a fact that motivates us to keep introducing these much-appreciated measures. In other words, it is also important to note that online businesses most frequently use memory-based RS because it is the most effective and straightforward to install. So, the scope of this paper is restricted to demonstrate the vital impact of the similarity measures, including the proposed and combined ones, on memory-based collaborative filtering performance using the item-based model with kNN. The results do imply that further research into nearest neighbor-driven memory-based collaborative filtering approaches is worthwhile. It is also worth indicating that the proposed similarity measures can still be effectively leveraged, with high performance, in all similarity measure-based fields (i.e., information retrieval and natural language processing).

5 Conclusions and Future Work

In this study, we employed an item-based model to experimentally examine the effectiveness of collaborative filtering. Dozens of similarity measures have been taken into account in order to conduct this investigation completely. In addition, we have developed five new similarity measures of simplistic design comparing to their rivals. Our proposed "simple" similarity measures have proved to be promising in comparison to their contenders, including the most recent ones. Additionally, it is clear from the outcomes of the MAE, MSE, and R metrics that Jaccard was able to play a significant role in the discovery of highly competitive measures like the Jaccard-fused ones (QTIJ and CosineJ). Surprisingly, when an item-based model is accounted, some of the top SOTA performers (i.e., SMD and NHSM) have not been able to significantly compete against their rivals. In other words, we observe that the top-performing SMD and NHSM [2] are not superior when the item-based model is used. We have confirmed that the proposed measures of this study can effectively compete against SMD and NHSM using the user-based model, as all experiments of our current work were only done using the item-based model. We believe that one important factor in the effective performance of our proposed measures is due to their straightforward design, which constructively takes into account both the magnitude and presence of rating vectors. Nevertheless, we think the performance of CF using these measures still needs to be further enhanced to achieve the best rendition using both item-based and user-based models. In conclusion, according to experimental findings on benchmark data sets, our suggested measures—in particular, NPMS—had competitive prediction and recommendation quality, and successfully addressed the data sparseness issue when compared to existing representative measures.

Therefore, future research will focus on adopting more novel similarity measures and carefully examining how each measure, including the Jaccard's combination, affects the performance of CF. This will be done using both user-based and item-based models. In particular, future work seeks to introduce an improved version of SMD to address the NHSM and SMD's shortcomings when the item-based model is taken into account. The main objective of this work was to efficiently use item and user-based models to reduce the impact of the sparsity problem on CF performance. The inclusion of as many similarity measurements, including the most current ones [27–29] in the comparison study, is another objective of the next effort. Moreover, we plan to integrate our proposed measures with multi-criteria ratings [3] and deep learning [30] for preference learning, and apply these models for several CF-related applications such as aware graph neural network [31] deep learning [32], and fuzzy-based recommender systems [33].

Acknowledgements The authors would like to thank and appreciate the support they received from the Research Office of Zayed University for providing the necessary facilities to accomplish this work. This research has been supported by the Research Incentive Fund (RIF) Grant Activity Code: R22083—Zayed University, UAE.

Author Contributions All authors have been key contributors in conception, design, implementing the approach and analyzing results of all experiments, and the preparation, writing and revising the manuscript.

Funding This research has been supported by Research Incentive Fund (RIF) Grant Activity Code: R22083—Zayed University, UAE.

Code and Data Availability The datasets used in this work are publicly available. The code exist on our Github repository at: https://github.com/aliamer/Study-the-Impact-of-an-Item-based-Model-on-CF-Performance-Using-a-Dozen-Similarity-Measures/tree/main.

Declarations

Conflict of interest The authors declare that they have no competing interests.

References

1. Lu, J., Wu, D., Mao, M., Wang, W., Zhang, G.: Recommender system application developments: a survey. Decis. Support Syst. **74**, 12–32 (2015). https://doi.org/10.1016/j.dss.2015.03.008
2. Pramod, D., Bafna, P.: Conversational recommender systems techniques, tools, acceptance, and adoption: a state of the art review. Expert Syst. Appl. (2022). https://doi.org/10.1016/j.eswa.2022.117539
3. Kaya, T., Kaleli, C.: A novel top-n recommendation method for multi-criteria collaborative filtering. Expert Syst. Appl. **198**, 116695 (2022). https://doi.org/10.1016/j.eswa.2022.116695
4. Amer, A.A., Abdalla, H.I., Nguyen, L.: Enhancing recommendation systems performance using highly-effective similarity measures. Knowl. Based Syst. (2021). https://doi.org/10.1016/j.knosys.2021.106842
5. Khojamli, H., Razmara, J.: Survey of similarity functions on neighborhood-based collaborative filtering. Expert Syst. Appl. **185**, 115482 (2021). https://doi.org/10.1016/j.eswa.2021.115482.10.1016/j.eswa.2021.115482
6. Singh, P.K., Pramanik, P.K.D., Choudhury P.: A comparative study of different similarity metrics in highly sparse rating dataset. In: Balas, V., Sharma, N., Chakrabarti, A. (eds.) Data Management, Analytics and Innovation (vol. 2), vol. 839. https://doi.org/10.1007/978-981-13-1274-8_4 of Advances in Intelligent Systems and Computing, pp. 45–60. Springer (2018). https://doi.org/10.1007/978-981-13-1274-8_4
7. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A.: Recommender systems survey. Knowl. Based Syst. **46**, 109–132 (2013). https://doi.org/10.1016/j.knosys.2013.03.012
8. Singh, P.K., Sinha, S., Choudhury, P.: An improved item-based collaborative filtering using a modified Bhattacharyya coefficient and user–user similarity as weight. Knowl. Inf. Syst. **64**(3), 665–701 (2022). https://doi.org/10.1007/s10115-021-01651-8

9. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, Item-based collaborative filtering recommendation algorithms, pp. 285–295. ACM (2001)
10. Kant, S., Mahara, T.: Merging user and item based collaborative filtering to alleviate data sparsity. Int. J. Syst. Assur. Eng. Manag. **9**(1), 173–179 (2018). https://doi.org/10.1007/s13198-016-0500-9
11. Singh, P.K., Sinha, M., Das, S., Choudhury, P.: Enhancing recommendation accuracy of item-based collaborative filtering using Bhattacharyya coefficient and most similar item. Appl. Intell. (2020). https://doi.org/10.1007/s10489-020-01775-4
12. Singh, P. K., Ahmed, R., Rajput, I. S., & Choudhury, P.: A comparative study on prediction approaches of item-based collaborative filtering in neighborhood-based recommendations. In: Wireless Personal Communications. Springer (2021). https://doi.org/10.1007/s11277-021-08662-2
13. Anwar, T., Uma, V., Hussain, M.I., Pantula, M.: Collaborative filtering and kNN based recommendation to overcome cold start and sparsity issues: a comparative analysis. Multimed. Tools Appl. (2022). https://doi.org/10.1007/s11042-021-11883-z
14. Zhang, Z.P., Kudo, Y., Murai, T., Ren, Y.G.: Enhancing recommendation accuracy of item-based collaborative filtering via item-variance weighting. Appl. Sci. (Switzerland). (2019). https://doi.org/10.3390/app9091928
15. ChemmalarSelvi, G., Lakshmi Priya, G.G.: Rating prediction method for item-based collaborative filtering recommender systems using formal concept analysis. EAI Endorsed Trans. Energy Web **8**(33), 1–9 (2021). https://doi.org/10.4108/eai.2-12-2020.167291
16. Ajaegbu, C.: An optimized item-based collaborative filtering algorithm. J. Ambient. Intell. Humaniz. Comput. **12**(12), 10629–10636 (2021). https://doi.org/10.1007/s12652-020-02876-1
17. Zhao, W., Tian, H., Wu, Y., Cui, Z., Feng, T.: A new item-based collaborative filtering algorithm to improve the accuracy of prediction in sparse data. Int. J. Comput. Intell. Syst. (2022). https://doi.org/10.1007/s44196-022-00068-7
18. Nam, L.N.H.: Towards comprehensive approaches for the rating prediction phase in memory-based collaborative filtering recommender systems. Inf. Sci. **589**, 878–910 (2022). https://doi.org/10.1016/j.ins.2021.12.123
19. Dhanabal, S., Baskar, K., Premkumar, R.: Fusion of internal similarity to improve the accuracy of recommendation algorithm. J. Univ. Shanghai Sci. Technol. **24**(1), 139–140 (2022). https://doi.org/10.51201/jusst/21/121044
20. Bag, S., Kumar, S.K., Tiwari, M.K.: An efficient recommendation generation using relevant Jaccard similarity. Inform. Sci. **483**, 53–64 (2019). https://doi.org/10.1016/j.ins.2019.01.023
21. Gazdar, A., Hidri, L.: A new similarity measure for collaborative filtering based recommender systems. Knowl. Based Syst. **188**, 105058 (2020). https://doi.org/10.1016/j.knosys.2019.105058
22. Wang, Y., Wang, P., Liu, Z., Zhang, L.Y.: A new item similarity based on α-divergence for collaborative filtering in sparse data. Expert Syst. Appl. **166**, 114074 (2021). https://doi.org/10.1016/j.eswa.2020.114074
23. Wang, Y., Deng, J., Gao, J., Zhang, P.: A hybrid user similarity model for collaborative filtering. Inform. Sci. **418–419**, 102–118 (2017). https://doi.org/10.1016/j.ins.2017.08.008
24. Luo, X., Wang, D., Zhou, M., Yuan, H.: Latent factor-based recommenders relying on extended stochastic gradient descent algorithms. IEEE Trans. Syst. Man Cybern. Syst. **51**(2), 916–926 (2019). https://doi.org/10.1109/TSMC.2018.2884191
25. Wang, D., Chen, Y., Guo, J., Shi, X., He, C., Luo, X., Yuan, H.: Elastic-net regularized latent factor analysis-based models for recommender systems. Neurocomputing **329**, 66–74 (2019). https://doi.org/10.1016/j.neucom.2018.10.046
26. Shrivastava, R., Sisodia, D.S., Nagwani, N.K.: Deep neural network-based multi-stakeholder recommendation system exploiting multi-criteria ratings for preference learning. Expert Syst. Appl. **213**, 119071 (2023). https://doi.org/10.1016/j.eswa.2022.119071
27. Amer, A.A., Nguyen, L.: Combinations of Jaccard with numerical measures for collaborative filtering enhancement: current work and future proposal (2021). https://doi.org/10.48550/arXiv.2111.12202
28. Hayashi, K.: Rethinking correlation-based item-item similarities for recommender systems. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2287–2291 (2022). https://doi.org/10.1145/3477495.3532055
29. Nguyen, L., Amer, A.A.: Advanced cosine measures for collaborative filtering. Adapt Personal. (ADP) **1**, 21–41 (2019). https://doi.org/10.31058/j.adp.2019.11002
30. Tao, H., Niu, X., Fu, L., Yuan, S., Wang, X., Zhang, J., Hu, Y.: DeepRS: a library of recommendation algorithms based on deep learning. Int. J. Comput. Intell. Syst. **15**(1), 45 (2022). https://doi.org/10.1007/s44196-022-00102-
31. Zhang, Y., Li, C., Cai, J., Liu, Y., Wang, H.: BKGNN-TI: a bilinear knowledge-aware graph neural network fusing text information for recommendation. Int. J. Comput. Intell. Syst. **15**(1), 95 (2022). https://doi.org/10.1007/s44196-022-00154-w
32. Li, W.: Intelligent recommendation system based on the infusion algorithms with deep learning, attention network and clustering. Int. J. Comput. Intell. Syst. **16**(1), 83 (2023). https://doi.org/10.1007/s44196-023-00264-z
33. Yera, R., Martinez, L.: Fuzzy tools in recommender systems: a survey. Int. J. Comput. Intell. Syst. **10**(1), 776 (2017). https://doi.org/10.2991/ijcis.2017.10.1.52

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