Knowledge-Based Systems 217 (2021) 106842

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Enhancing recommendation systems performance using highly-effective similarity measures

Ali A. Amer [a,*], Hassan I. Abdalla [b], Loc Nguyen [c]

[a] Computer Science Department, TAIZ University, TAIZ, Yemen
[b] College of Technological Innovation, Zayed University, P.O. Box 144534, Abu Dhabi, United Arab Emirates
[c] Loc Nguyen's Academic Network, Board of Advisors, Long Xuyen, Viet Nam

ARTICLE INFO

Article history:
Received 13 September 2020
Received in revised form 20 January 2021
Accepted 23 January 2021
Available online 10 February 2021

Dataset link: https://drive.google.com/drive/folders/1lz3-eVjAf-IZ5auIJSK4dX81Wt2_OFz3?fbclid=IwAR0fgDjrlUORMdhMg5TKVxd-tMHoFKooDOYH9g1rEXRFV7yJqV1L3_Q674U, https://github.com/aliamer/Enhancing-Recommendation-Systems-Performance-Using-Highly-Effective-Similarity-Measures

Keywords:
Collaborating filtering
Recommendation systems
Similarity
KNN algorithm
Cross validation
Empirical evaluation

ABSTRACT

In Recommendation Systems (RS) and Collaborative Filtering (CF), the similarity measures have been the operating component upon which CF performance is essentially reliant. A dozen of similarity measures have been proposed to reach the desired performance particularly under the circumstances of data sparsity (the cold-start problem). Nevertheless, these measures still suffer the cold-start problem, and have a complex design. Moreover, a comprehensive experimental work to study the impact of the cold-start problem on CF performance is still missing. To these ends, therefore, this paper introduces three simply-designed similarity measures, namely, difference-based similarity measure (SMD), hybrid difference-based similarity measure (HSMD), and, triangle-based cosine measure (TA). Along with proposing these measures, a comprehensive experimental guide for CF measures using the K-fold cross validation is also presented. In contrary to all previous CF studies, the evaluation process is split into two sub-processes: the estimation process and recommendation process to accurately obtain the desired appropriateness in the evaluation. In addition, a new formula to calculate the dynamic recommendation count is developed depending on both the dataset and rating vectors. To draw a comprehensive experimental analysis, a dozen state-of-the-art similarity measures (30 similarity measures) including the proposed and the most widely-used traditional measures are comparatively tested. The experimental study has critically been made on three datasets with five-fold cross-validation grounded on the K nearest neighbor algorithm (KNN). The obtained results on both estimation and recommendation processes prove unquestionably that SMD and TA are preeminent measures with the lowest computational complexity outperforming all state-of-the-art CF measures.

1. Introduction

One of the ultimate aims of the online companies is to offer a highly-effective personalized recommendations to enormous number of users based on their past preferences. The recommender system (RS) has a vital role in the industry of electronic commerce for helping and advising customers to select their favorite products [1,2] among millions of products. Moreover, RS has been popularly leveraged in diverse sectors, including the online-based business sectors like the sector of travel, online broadcasting, online articles books (either scientific or news, like LIBRA which is a book recommender system), online advertising, a movie sites like Netflix, and music [2–4], etc. All of these sectors combined would establish gigantic volumes of rating data in a daily basis. Therefore, to properly process such huge volumes of data, several works have been found in recommendation systems literature over the last twenty years. Most of the earlier works focused on introducing three types of filtering approaches to process data. These approaches are content-based filtering (CBF), collaborative filtering (CF) and hybrid filtering (HF) which is a combination of CBF and CF. Both CBF and CF work through navigating user/item profiles to discover the past preferences of user/item along with using the user/item similarity metrics.

Generally speaking, CBF and CF approaches are the most widely studied in recommendation systems literature [5,6]. The CF can further be split into two classes, namely, the model-based CF and memory-based CF. On one extreme, the memory-based CF utilizes the rating vectors to discover the similar users/items [7]. This class is the most commonly leveraged by online companies due to it being more efficient and easier to be implemented than the model-based CF. In the model-based CF, on the other hand,

The code (and data) in this article has been certified as Reproducible by Code Ocean:https://help.codeocean.com/en/articles/1120151-code-ocean-s-verification-process-for-computational-reproducibility. More information on the Reproducibility Badge Initiative is available at www.elsevier.com/locate/knosys.

* Corresponding author.
E-mail addresses: aliaaa2004@yahoo.com (A.A. Amer), Hassan.Abdalla@zu.ac.ae (H.I. Abdalla), ng_phloc@yahoo.com (L. Nguyen).

https://doi.org/10.1016/j.knosys.2021.106842
0950-7051/

Table 1
User-based rating matrix.

	Item 1	Item 2	Item 3	Item 4
User 1	$r_{11} = 1$	$r_{12} = 2$	$r_{13} = 1$	$r_{14} = 5$
User 2	$r_{21} = 2$	$r_{22} = 1$	$r_{23} = 2$	$r_{24} = 4$
User 3	$r_{31} = 4$	$r_{32} = 1$	$r_{33} = 5$	$r_{34} = 5$
User 4	$r_{41} = 1$	$r_{42} = 2$	$\boldsymbol{r_{43} = ?}$	$\boldsymbol{r_{44} = ?}$

the latent factors are searched and found to run the prediction like singular value decomposition (SVD) [8] and Bayesian networks [9]. In comparison, the model-based CF is faster than the memory-based CF to make the prediction. However, the memory-based CF produces results of higher accuracy than those results produced by model-based CF [10]. CF is the focus of our work as it has long been an effective approach for recommendation systems which can efficaciously predict the potential future of users' interests depending on their previous preferences [11]. Depending on the users' given ratings, the CF model would cluster the most similar users/items, across building and using the user–user or item–item similarity measures.

However, CF recommends an item to a user if his/her neighbors are interested in such an item [12,13]. In its turn, the item is anything that users consider, such as books, newspapers, etc. Some of the most famous instances of recommender systems are YouTube [14], Amazon [15], and Google news [16]. In recommender systems, there are two main classes to generate recommendations, namely user-based and item-based models. Both models have mostly been using the K-nearest neighbors (KNN) algorithm. The KNN is one of the most popular algorithms used in CF. The essence of the KNN algorithm is to find out the nearest neighbors of the targeted user (called active user), and then to recommend the active user's items that their neighbors may like. Let $U = \{U_1, U_2, \ldots, U_m\}$ be the set of m users and let $V = \{V_1, V_2, \ldots, V_n\}$ be the set of n items. On one hand, the user-based rating matrix is the matrix in which rows indicate users and columns indicate items, and each cell is a rating that a user U_i gave to an item I_j. On the other hand, the item-based rating matrix is the matrix in which rows indicate items and columns indicate users, and each cell is a rating that each item I_j has been given by the corresponding user U_i. In other words, each row in the user-based rating matrix is a rating vector of a specified user to several items, and each row in an item-based rating matrix is a rating vector of a specified item given by several users. The rating vector of the active user is called an active user vector [1,17].

Table 1 provides a simple example of the user-based rating matrix in which the missing values are denoted by question marks and the rating values range from 1 to 5 [18]. In Table 1, active vector is $U_4 = \{r_{41} = 1, r_{42} = 2, r_{43} = ?, r_{44} = ?\}$, which is shaded in a gray color.

In Table 1, there have been four rating vectors $u_1 = (1, 2, 1, 5)$, $u_2 = (2, 1, 2, 4)$, $u_3 = (4, 1, 5, 5)$, and $u_4 = (1, 2, r_{43} = ?, r_{44} = ?)$. Suppose the active rating vector is u_4, KNN algorithm will discover the nearest neighbors of u_4, and then compute the predictive values for r_{43} and r_{44} based on the similarities between these neighbors and u_4. The KNN algorithm which acts on the user-based rating matrix is called the user-based KNN algorithm. Even though both the user-based KNN algorithm and item-based KNN algorithm have the same ideology, their implementations are significantly different.

KNN algorithm, on the other hand, is still the most-widely used technique in CF literature [1,17] to make experimental studies and judge CF performance. However, while scanning literature, we found that a comprehensive experimentally-oriented similarity measure-based CF study is still missing. Most of earlier studies have either presented a new similarity measure with a limited comparative study with several measures, or investigated some similarity measures ranges from 6, 8 to 12 measures. Moreover, in the middle of the constant development to generate new measures every now and then, there has no single work sought to experimentally investigate the most-widely used similarity measures including those claimed to be top performers as well as the traditional ones at the same time. So, it is being extremely confusing to perceive which similarity measure would behave the best on RS chiefly under the data sparsity conditions.

In its turn, traditional similarity measures such as Cosine and PCC are used to determine the nearest neighbors of the targeted user. However, the computation costs for this kind of indices are relatively expensive chiefly in sparse data (cold-start problem), and thus it is not feasible to apply them effectively on the huge-sized sparse datasets. Albeit these problems have been addressed by proposing several measures to tackle the cold-start problem in particular. These newly-proposed measures, nevertheless, still suffer the cold-start problem as no single measure has been recorded yet to be an excellent measure for both processes (estimation and recommendation) of CF as shown in the drawn-below results. In addition, albeit their being effective, most of the new measures, including top performers, have a very complex design. These deficits have been key motivations that drove us to find highly competitive similarity measures of a very simplistic design, which is successfully accomplished through proposing these measures of this work. This research also focuses on presenting highly effective similarity measures for CF under the condition of data sparsity. Concisely, this paper comes to cover some of the drawn-above deficits through the following contributions:

1. Proposing three promising similarity measures to tackle the cold-start problem effectively. These measures are simply and professionally designed, and experimentally shown to be maximally effective in finding the accurate predictions and recommendations. They are also shown to be time-efficient.
2. To sustainably improve RS performance and quality, some important factors are implicitly considered in the design process of the proposed similarity measures without introducing any weighting factors. Among these factors are reducing or eliminating the full reliant on the co-rated items, making full use of all rated and non-rated items which lead to treat similarity as symmetric and asymmetry at the same time.
3. Unlike the earlier studies which either fixed the number of recommended items or changed the number over some values such as 10, 20, and 100, we propose a method to calculate the dynamic recommendation count C, based on the dataset with the purpose that N – the total number of the estimated items – will be more accurate and objective. The proposed method is dynamically formulated, and the formula takes advantage of the so-called sparse-relevant ratio.
4. Making an extensive empirically-driven comparative study for the commonly-used similarity measures in CF using the user-based model with 5-fold cross-validation. Roughly 30 similarity measures have been involved in the experimental study. This number of similarity measures makes this work unique as the first study that investigates and experiments such a dozen of state-of-art similarity measures. As a matter of fact, the aim is: to benchmark these measures on the targeted datasets, and establish a good experimental guide for CF scholars so they can perceive the impact of state-of-art similarity measures on the estimation and recommendation process under circumstances of data sparsity.

The rest of this paper is planned as follows. In Section 2, the closely-relevant works are covered. In Section 3, all similarity measures, that are sought to be evaluated in this work, are briefly mentioned. In Section 4, the proposed methodology, including the problem statement, motivations, and proposed similarity measures, is introduced. Section 5 draws the performance evaluation including the experimental setup of the proposed work, datasets, and the evaluation process. Section 6 provides the experimental results in details. Section 7 presents the discussion briefly. Finally, the conclusions and future work directions are given in Section 8.

2. Related work

It is commonly shown in CF literature that the traditional similarity measures like Pearson Correlation Coefficient (PCC), and their derivations are not robust enough under specific circumstances such as data sparsity. To overcome the limitation of these measures, several similarity measures in CF literature have been presented, and applied on RS, and their effects had been recorded.

For instance, Ayub et al. [3] proposed an improved jaccard measure which took the ratio between the values of absolute rating and the number of commonly-rated items into consideration. Authors also used threshold to enhance measure performance on the average rating value of a user. In the same page, Bobadilla et al. [19] proposed a combined measure through consolidating Jaccard and Mean Squared Difference (JMSD). Jaccard was set to catch the proportion of co-related items and MSD was used to grab the ratings information. Likewise, Bobadilla et al. [20] utilized the contextual information to present a similarity measure called "MJD" which is short for (Mean-Jaccard-Differences) to improve the traditional similarity measures using the singularity of user ratings. The ratings were categorized into two groups: positive and non-positive. A six similarity measures were combined to seize the global similarity, and the weight of every similarity measure was secured using the neural network learning. The users and items singularity were then combined with the actual ratings to find the weight of similarity between the targeted users. These measures, however, have not worked well under the circumstance of data sparsity (also known as cold-start problem).

On the other hand, Bobadilla et al. [20] was used in [21] to develop a new measure based on three kinds of significance: the item significance, the user significance in terms of users giving recommendations to other users, and the item significance for a user. The PCC and Cosine measures were then applied to find the intended likened clusters. Choi and Suh [22] combined some of the traditional measures (PCC, Cosine and some distance metrics) to introduce a new combined similarity measure. The correlation between the intended item and each co-rated item was taken into account to find similarity between users in the same neighborhood. Mykhaylo et al. [23] showed that Cosine and PCC measures yielded poor results for the recommendation prediction. The authors then used the inverse Euclidean distance (IED) to find the similarity between the ratings, and the results were seen to be slightly enhanced. El Alami et al. [24] evolved a new similarity measure to select neighbors based on both the neighborhood union and intersection. The neighbors were defined in two cases: those who shared the same items of user of interest, and those who shared at least one item of user of interest. Nevertheless, the proposed measures were still reliant on the shared items. As a result, the similarity between items would have a zero value when there is no shared items between the intended users, making these measures faulty in this case.

A new combined measure was given in [25] to improve RS accuracy under the data sparsity circumstance. This measure used mean measure of divergence (MMD) that considers the behavior of users' rating either high or low. Patra et al. [26] came up with a new similarity measure based on Bhattacharyya coefficient in the memory-based CF to tackle the data sparsity problem. On the other hand, Bhattacharyya Coefficient (BC) was used in [27] for the likeness enhancement as well as solving the data sparsity problem. It started by first locating the nearest neighbors of items of interest across BC computation between each two items and then taking the top N items to define the neighborhood of item (s) of interest. Then, second, the nearest neighbors of users of interest were located using the similarity measures proposed in [26].

Lately, a subspace clustering-driven measure was presented in [28] to tackle both problems of the high dimensionality and the data sparsity. The space of items was split into three subspaces: 1. Interested, 2. Neither Interested nor Uninterested, and 3. Uninterested. Then, using these subspaces, users correlations were computed. Saranya and Sadasivam [29] proposed a linear combination to address the data sparsity dilemma. Using the Proximity – Significance – Singularity measure (PSS) which was proposed in [30], Bhattacharya Coefficient, and Jaccard, the preferences and local context of users' behavior and the percentage of shared ratings between each user pair were taken into account.

Ahn [30] studied the deficits of traditional similarity measures in CF. Using the specific meanings of co-ratings and the explanation of user ratings, author then introduced a heuristic similarity measure. This measure was named PIP which stands for three semantic heuristics, namely, Proximity, Impact and Popularity. However, PIP was seen a faulty measure in the next aspects: (1) the absolute ratings were not considered, so the measure did not take the effects of non-related items into account, and the co-rated items were disregarded, (2) the user's global rating preference was not taken into account, and (3) PIP equation was not normalized. Driven by these deficits, Liu et al. [31] introduced a new heuristic similarity model called (NHSM). This measure was based on PIP, and indeed; NHMS tackled the limitation of PIP. The NHSM identifies each user pair through using the rating preference of each user.

Sun et al. [32] combined the Triangle and Jaccard similarities as a multiplication to form one single measure called the Triangle multiplying Jaccard (TMJ) similarity. While the angle and the length ratings (only the co-rating users/items) were considered by triangle, the non-related users were considered by Jaccard. The comparison was made under the leave one-out scenario on four datasets in terms of MAE and RSME. Jin et al. [33] proposed a new similarity measure that used a singularity factor to adjust non-linear equation. The measures were compared with traditional measures on Movielens 100-K. The proposed measure was seen enhancing the prediction accuracy better than the state-of-art measures. Finally, Chen et al. [34] proposed a vertex similarity index which was named CosRA. CosRA was a combination of both the cosine index and the resource-allocation (RA) index. The CosRA-based method was seen to have better performance in accuracy, diversity and novelty than its peers against which the evaluation was done. One significant advantage of the CosRA index was its being free of parameters making it a good measure in real applications.

3. The compared similarity measures

This section is fully dedicated to mention all similarity measures of CF that have been used to accomplish this study. Assuming we have 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, in which user 1 represents an active user and some r_{ij} can be missing (empty). The $|u_1|$ and $|u_2|$ are the lengths of u_1 and u_2, respectively whereas $u_1 * u_1$ is the dot product (scalar product) of u_1 and u_2, respectively.

Let I_1 and I_2 be the set of indices of items that user 1 and user 2 rated, respectively. Let $I = I_1 \cap I_2$ denotes the intersection set of I_1 and I_2 and let $I_1 \cup I_2$ denotes the union set of I_1 and I_2. All items whose indices belong to $I_1 \cap I_2$ are rated by both user 1 and user 2. In other words, all items whose indices belong to $I_1 \cap I_2$ co-exist in vectors u_1 and u_2. All items whose indices belong to $I_1 \cup I_2$ are rated by user 1 or user 2. Notation $|x|$ indicates an absolute value of the number, length of the vector, length of the geometric segment, or the cardinality of a set, which depends on the context. These denotations are going to be used along the paper. Let sim(u_1, u_2) denotes the similarity of u_1 and u_2. The compared similarity measures are listed as follows;

The **Cosine measure** of u_1 and u_2 is defined as follows [35]:

$$\text{sim}(u_1, u_2) = \cos(u_1, u_2) = \frac{u_1 * u_2}{|u_1|\,|u_2|} = \frac{\sum_{j \in I_1 \cap I_2} r_{1j} r_{2j}}{\sqrt{\sum_{j \in I_1 \cap I_2} (r_{1j})^2} \sqrt{\sum_{j \in I_1 \cap I_2} (r_{2j})^2}} \quad (1)$$

The Pearson correlation is another popular similarity measure, which is defined as follows [36]:

$$\text{Pearson}(u_1, u_2) = \frac{\sum_{j \in I_1 \cap I_2} (r_{1j} - \overline{u}_1)(r_{2j} - \overline{u}_2)}{\sqrt{\sum_{j \in I_1 \cap I_2} (r_{1j} - \overline{u}_1)^2} \sqrt{\sum_{j \in I_1 \cap I_2} (r_{2j} - \overline{u}_2)^2}} \quad (2)$$

where $\overline{u}_1$ and $\overline{u}_2$ are the mean values of u_1 and u_2, respectively.

$$\overline{u}_1 = \frac{1}{|I_1|} \sum_{j \in I_1} r_{1j}$$

$$\overline{u}_2 = \frac{1}{|I_2|} \sum_{j \in I_2} r_{2j}$$

The **Constrained Pearson correlation (CPC) measure** considers the impact of positive and negative ratings by using the median r_m instead of using the means. CPC measure is defined as follows [31]:

$$\text{CPC}(u_1, u_2) = \text{CON}(u_1, u_2) = \frac{\sum_{j \in I_1 \cap I_2} (r_{1j} - r_m)(r_{2j} - r_m)}{\sqrt{\sum_{j \in I_1 \cap I_2} (r_{1j} - r_m)^2} \sqrt{\sum_{j \in I_1 \cap I_2} (r_{2j} - r_m)^2}} \quad (3)$$

The **Weight Pearson correlation (WPC)** and **Sigmoid Pearson Correlation (SPC)** measures concern on how much common items are exist. WPC and SPC are defined as follows [31]:

$$\text{WPC}(u_1, u_2) = \begin{cases} \text{Pearson}(u_1, u_2) * \frac{|I|}{H}, \text{ if } |I| \le H \\ \text{Pearson}(u_1, u_2), \text{ otherwise} \end{cases} \quad (4)$$

$$\text{SPC}(u_1, u_2) = \text{Pearson}(u_1, u_2) * \frac{1}{1 + \exp\left(-\frac{|I|}{2}\right)} \quad (5)$$

where H is a threshold, and it is often set to be 50.

The Jaccard measure is defined as follows:

$$\text{Jaccard}(u_1, u_2) = \frac{|I_1 \cap I_2|}{|I_1 \cup I_2|} \quad (6)$$

Another version of Jaccard is the Jaccard2 which is defined as follows:

$$\text{Jaccard2}(u_1, u_2) = \frac{|I_1 \cap I_2|}{|I_1|\,|I_2|} \quad (7)$$

By following the ideology of Jaccard measure and cosine measure, the modified **COJ** is presented as follows:

$$\text{COJ}(u_1, u_2) = \frac{\sum_{j \in I_1 \cap I_2} r_{1j} r_{2j}}{\sqrt{\sum_{j \in I_1} (r_{1j})^2} \sqrt{\sum_{j \in I_2} (r_{2j})^2}} \quad (8)$$

On the other hand, t**he Normalized Cosine measure (CON)** is defined as follows:

$$\text{CON}(u_1, u_2) = \text{CPC}(u_1, u_2) = \frac{\sum_{j \in I_1 \cap I_2} (r_{1j} - r_m)(r_{2j} - r_m)}{\sqrt{\sum_{j \in I_1 \cap I_2} (r_{1j} - r_m)^2} \sqrt{\sum_{j \in I_1 \cap I_2} (r_{2j} - r_m)^2}}$$

The CON measure is a CPC measure (see Eq. (5)).

Let $v_j = (r_{1j}, r_{2j}, \ldots, r_{mj})$ be the vector of the rating values that item j receives from m users, for example. The mean of v_j is:

$$\overline{v}_j = \frac{1}{m} \sum_{i=1}^{m} r_{ij}$$

The adjusted cosine measure (COD) is defined as follows:

$$\text{COD}(u_1, u_2) = \frac{\sum_{j \in I_1 \cap I_2} (r_{1j} - \overline{v}_j)(r_{2j} - \overline{v}_j)}{\sqrt{\sum_{j \in I_1 \cap I_2} (r_{1j} - \overline{v}_j)^2} \sqrt{\sum_{j \in I_1 \cap I_2} (r_{2j} - \overline{v}_j)^2}} \quad (9)$$

On the other extreme, Jaccard can be combined with other measures to produce new combined measures. For instance, with cosine to produce the **CosineJ** which is defined as follows:

$$\text{CosineJ}(u_1, u_2) = \text{cosine}(u_1, u_2) * \text{Jaccard}(u_1, u_2) = \frac{\sum_{j \in I_1 \cap I_2} r_{1j} r_{2j}}{\sqrt{\sum_{j \in I_1} (r_{1j})^2} \sqrt{\sum_{j \in I_2} (r_{2j})^2}} * \frac{|I_1 \cap I_2|}{|I_1 \cup I_2|} \quad (10)$$

The PearsonJ is a combinations of Jaccard and Pearson, and is defined as follows:

$$\text{PearsonJ}(u_1, u_2) = \text{Pearson}(u_1, u_2) * \text{Jaccard}(u_1, u_2) = \frac{\sum_{j \in I_1 \cap I_2} (r_{1j} - \overline{u}_1)(r_{2j} - \overline{u}_2)}{\sqrt{\sum_{j \in I_1 \cap I_2} (r_{1j} - \overline{u}_1)^2} \sqrt{\sum_{j \in I_1 \cap I_2} (r_{2j} - \overline{u}_2)^2}} * \frac{|I_1 \cap I_2|}{|I_1 \cup I_2|} \quad (11)$$

Mean squared difference (MSD) is defined as an inverse of the distance between two vectors. Let MAX be the maximum value of ratings, MSD is calculated as follows:

$$\text{MSD}(u_1, u_2) = 1 - \frac{\sum_{j \in I} \left(\frac{r_{1j} - r_{2j}}{\text{MAX}}\right)^2}{|I|} \quad (12)$$

Another variant of MSD was specified by [19] as follows:

$$\text{MSD}(u_1, u_2) = \frac{1}{1 + \frac{1}{|I|} \sum_{j \in I} (r_{1j} - r_{2j})^2} \quad (13)$$

Meanwhile, MSD measure was combined with Jaccard measure to derive the **MSDJ measure** as follows:

$$\text{MSDJ}(u_1, u_2) = \text{MSD}(u_1, u_2) * \text{Jaccard}(u_1, u_2) \quad (14)$$

When the rating values are converted into ranks, **Spearman's Rank Correlation (SRC)** is defined as follows [32]:

$$\text{SRC}(u_1, u_2) = 1 - \frac{6 \sum_{j \in I} d_j^2}{|I|\left(|I|^2 - 1\right)} \quad (15)$$

where d_j is the difference between two ranks on item j given by user 1 and user 2.

$$d_j = \text{rank}_{1j} - \text{rank}_{2j}$$

On the other hand, to solve the data sparsity problem, some other studies were proposed in CF literature. In [32], PIP was proposed based on the concept of "agreement" in rating. If both

user 1 and user 2 like or dislike the same item, it is called that they have a rating "agreement" on such items. Let r_{1j} and r_{2j} be the ratings of user 1 and user 2 on item j, respectively, their agreement is defined as follows:

$$\text{agree}\left(r_{1j}, r_{2j}\right) = \begin{cases} \text{true if } \left(r_{1j} > r_m \text{and} r_{2j} > r_m\right) \\ \text{true if } \left(r_{1j} < r_m \text{and} r_{2j} < r_m\right) \\ \text{false otherwise} \end{cases}$$

PIP measure is the sum of the products of triples Proximity, Impact, and Popularity, and is given as follows:

$$\text{PIP}\left(u_1, u_2\right) = \sum_{j \in I_1 \cap I_2} \text{Proximity}\left(r_{1j}, r_{2j}\right) * \text{Impact}\left(r_{1j}, r_{2j}\right) * \text{Popularity}\left(r_{1j}, r_{2j}\right) \quad (16)$$

where Proximity, Impact and Popularity are computed as follows

$$\text{Proximity}\left(r_{1j}, r_{2j}\right) = \begin{cases} \left(\left(2\left(r_{max} - r_{min}\right) + 1\right) - \left|r_{1j} - r_{2j}\right|\right)^2 \\ \quad \text{if agree}\left(r_{1j}, r_{2j}\right) = \text{true} \\ \left(\left(2\left(r_{max} - r_{min}\right) + 1\right) - 2\left|r_{1j} - r_{2j}\right|\right)^2 \\ \quad \text{if agree}\left(r_{1j}, r_{2j}\right) = \text{false} \end{cases}$$

where r_{min} and r_{max} are the minimum rating and maximum rating values, respectively

$$\text{Impact}\ \left(r_{1j}, r_{2j}\right) = \begin{cases} \left(\left|r_{1j} - r_m\right| + 1\right) * \left(\left|r_{2j} - r_m\right| + 1\right) \\ \quad \text{if agree}\ \left(r_{1j}, r_{2j}\right) = \text{true} \\ \frac{1}{\left(\left|r_{1j} - r_m\right| + 1\right) * \left(\left|r_{2j} - r_m\right| + 1\right)} \\ \quad \text{if agree}\left(r_{1j}, r_{2j}\right) = \text{false} \end{cases}$$

$$\text{Popularity}\left(r_{1j}, r_{2j}\right) = \begin{cases} 1 + \left(\frac{r_{1j} + r_{2j}}{2} - \mu_j\right)^2 \\ \text{if}\ \left(r_{1j} > \mu_j \text{ and } r_{2j} > \mu_j\right) \\ 1 + \left(\frac{r_{1j} + r_{2j}}{2} - \mu_j\right)^2 \\ \text{if}\ \left(r_{1j} < \mu_j \text{ and } r_{2j} < \mu_j\right) \\ 1 \\ \text{otherwise} \end{cases}$$

where μ_j is the average rating of item j, which is the same mean of rating values of item j.

The **PC measure** [22] – Pearson measure weighted by similarities of items – is defined as follows:

$$\text{PC}_k\left(u_1, u_2\right) = \frac{\sum_{j \in I}\left(\left(\text{sim}\left(v_k, v_j\right)\right)^2 \left(r_{1j} - \overline{u}_1\right)\left(r_{2j} - \overline{u}_2\right)\right)}{\sqrt{\sum_{j \in I_1}\left(\text{sim}\left(v_k, v_j\right)\left(r_{1j} - \overline{u}_1\right)\right)^2}\sqrt{\sum_{j \in I_2}\left(\text{sim}\left(v_k, v_j\right)\left(r_{2j} - \overline{u}_2\right)\right)^2}} \quad (17)$$

where k is the active item, $\text{sim}(v_k, v_j)$ is the similarity of the active item k and item j. The $\overline{u}_1$ and $\overline{u}_2$ are the mean values of u_1 and u_2, respectively.

$$\overline{u}_1 = \frac{1}{|I_1|}\sum_{j \in I_1} r_{1j}$$

$$\overline{u}_2 = \frac{1}{|I_2|}\sum_{j \in I_2} r_{2j}$$

On the other hand, the PSS measure (*Proximity – Significance – Singularity*) is calculated as follows:

$$\text{PSS}\left(u_1, u_2\right) = \sum_{j \in I} \text{Proximity}\left(r_{1j}, r_{2j}\right) * \text{Significance}\left(r_{1j}, r_{2j}\right) * \text{Singularity}\left(r_{1j}, r_{2j}\right) \quad (18)$$

$$\text{Proximity}\left(r_{1j}, r_{2j}\right) = 1 - \frac{1}{1 + \exp\left(-\left|r_{1j} - r_{2j}\right|\right)}$$

$$\text{Significance}\left(r_{1j}, r_{2j}\right) = \frac{1}{1 + \exp\left(-\left|r_{1j} - r_m\right|\left|r_{2j} - r_m\right|\right)}$$

$$\text{Singularity}\left(r_{1j}, r_{2j}\right) = 1 - \frac{1}{1 + \exp\left(-\left|\frac{r_{1j} + r_{2j}}{2} - \mu_j\right|\right)}$$

Whereas μ_j is the rating mean of item j. Liu et al. [31] also considered the similarity between two users via URP measure as follows:

$$\text{URP}\left(u_1, u_2\right) = 1 - \frac{1}{1 + \exp\left(-\left|\mu_1 - \mu_2\right|\left|\sigma_1 - \sigma_2\right|\right)} \quad (19)$$

where μ_1 and μ_2 are the rating means of user 1 and user 2, respectively and σ_1 and σ_2 are the rating standard deviations of user 1 and user 2, respectively.

$$\mu_1 = \frac{1}{|I_1|}\sum_{j \in I_1} r_{1j}$$

$$\mu_2 = \frac{1}{|I_2|}\sum_{j \in I_2} r_{2j}$$

$$\sigma_1 = \sqrt{\frac{1}{|I_1|}\sum_{j \in I_1}\left(r_{1j} - \mu_1\right)^2}$$

$$\sigma_2 = \sqrt{\frac{1}{|I_2|}\sum_{j \in I_2}\left(r_{2j} - \mu_2\right)^2}$$

Then, using jaccard_2 along with Eqs. (18)–(19), the new heuristic similarity model (NHSM) was proposed. It is defined as follows:

$$\text{NHSM}\left(u_1, u_2\right) = \text{PSS}\left(u_1, u_2\right) * \text{URP}\left(u_1, u_2\right) * \text{Jaccard2}\left(u_1, u_2\right) \quad (20)$$

The BCF [28] finds the importance of each pair of the rated items by exploiting Bhattacharyya (BC) similarity. Given items i and j item, BC coefficient for items is calculated as follows:

$$\text{bc}\left(i, j\right) = \sum_{h=1}^{m}\sqrt{\frac{\#h_i}{\#i}\frac{\#h_j}{\#j}} \quad (21)$$

where $\#i$ and $\#j$ are the numbers of users who rated items i and j, respectively, whereas $\#h_i$ and $\#h_j$ are the numbers of users who gave the rating value h on items i and j, respectively. The BC similarity [28] is defined as follows:

$$\text{BC}\left(u_1, u_2\right) = \sum_{i \in I_1}\sum_{j \in I_2} \text{bc}\left(i, j\right)\ \text{loc}\left(r_{1i}, r_{2j}\right) \quad (22)$$

The local similarity is calculated as a part of the constrained Pearson coefficient (CPC) and calculated as follows:

$$\text{loc}\left(r_{1i}, r_{2j}\right) = \frac{\left(r_{1i} - r_m\right)\left(r_{2j} - r_m\right)}{\sqrt{\sum_{k \in I_1}\left(r_{1k} - r_m\right)^2}\sqrt{\sum_{k \in I_2}\left(r_{2k} - r_m\right)^2}}$$

Using Jaccard and BC, The BCF is defined as follows:

$$\text{BCF}\left(u_1, u_2\right) = \text{Jaccard}\left(u_1, u_2\right) + \text{BC}\left(u_1, u_2\right) \quad (23)$$

The Cosine-Jaccard-Mean was proposed in [25] as a Measure of Divergence (CjacMD) based on the Mean Measure of Divergence (MMD) to solve the problem of the sparse rating matrix. The CjacMD is found by combining three measures, namely, cosine, Jaccard, and MMD.

$$\text{CjacMD} = \cos\left(u_1, u_2\right) + \text{Jaccard}\left(u_1, u_2\right) + \text{MMD}\left(u_1, u_2\right) \quad (24)$$

where the MMD measure is defined as follows:

$$\text{MMD}\left(u_1, u_2\right) = \frac{1}{1 + \frac{1}{b}\sum_{j=1}^{b}\left(\left(\theta_{1j} - \theta_{2j}\right)^2 - \frac{1}{0.5 + x_j} - \frac{1}{0.5 + y_j}\right)} \quad (25)$$

where θ_{1j} and θ_{2j} are Grewal's transformations of X and Y, respectively.

$$\theta_{1j} = \frac{1}{\sin\left(1-\frac{2x_j}{|l_1|}\right)}$$
$$\theta_{2j} = \frac{1}{\sin\left(1-\frac{2y_j}{|l_2|}\right)}$$

The Triangle similarity measure (TS) in [33] considered both angle and lengths of the rating vectors. It is defined as follows:

$$\text{Triangle}(u_1,u_2) = 1 - \frac{|\text{AB}|}{|\text{OA}|+|\text{OB}|} = 1 - \frac{|u_1-u_2|}{|u_1|+|u_2|} = 1 - \frac{\sqrt{\sum_{j\in I}\left(r_{1j}-r_{2j}\right)^2}}{\sqrt{\sum_{j\in I} r_{1j}^2}+\sqrt{\sum_{j\in I} r_{2j}^2}} \quad (26)$$

where u_1 and u_2 are considered as two vector OA $= u_1$ and OB $= u_2$ and hence, OAB forms a triangle. TS was combined with Jaccard measure to form Triangle multiplying Jaccard (TMJ) measure which is defined as follows:

$$\text{TMJ}(u_1,u_2) = \text{Triangle}(u_1,u_2) * \text{Jaccard}(u_1,u_2) \quad (27)$$

The Feng similarity measures was proposed in [37] and defined as follows:

$$\text{Feng}(u_1,u_2) = S_1(u_1,u_2) * S_2(u_1,u_2) * S_3(u_1,u_2) \quad (28)$$

where S_1 is a normal similarity, and they choose cosine as S_1.

$$S_1(u_1,u_2) = \begin{cases}\text{cosine}(u_1,u_2) \text{ ifsparsity} < \rho \\ \text{COJ}(u_1,u_2) \text{ otherwise}\end{cases}$$

where ρ is the sparsity threshold. The S_2 punishes the user pairs whose co-rated items are few, and is defined as follows:

$$S_2(u_1,u_2) = \frac{1}{1+\exp\left(-\frac{|I_1\cap I_2|^2}{|I_1||I_2|}\right)}$$

The S_3 is the aforementioned URP measure.

$$S_3(u_1,u_2) = \text{URP}(u_1,u_2) = 1 - \frac{1}{1+\exp\left(-|\mu_1-\mu_2|\,|\sigma_1-\sigma_2|\right)}$$

Mu et al. [38] combined the local measures (Pearson and Jaccard) with Hellinger (Hg) distance as the global measure called Mu, which is defined as follows:

$$\text{Mu}(u_1,u_2) = \alpha * \text{Pearson}(u_1,u_2) + (1-\alpha) * (\text{Hg}(u_1,u_2) + \text{Jaccard}(u_1,u_2)) \quad (29)$$

where Hellinger (Hg), as the inverse of BC coefficient in the discrete distributions, was defined as follows:

$$\text{Hg}(u_1,u_2) = 1 - \text{bc}(u_1,u_2) = 1 - \sum_{h=1}^{m}\sqrt{\frac{\#h_1}{\#1}\frac{\#h_2}{\#2}} \quad (30)$$

where #1 and #2 are the numbers of items that are rated by user 1 and user 2, respectively whereas $\#h_1$ and $\#h_2$ are the numbers of items that receive the rating value h from user 1 and user 2, respectively.

Last but not least, the Similarity Measure for Text Processing (SMTP) was also implemented to test its effectiveness regarding CF. SMTP was developed in [39], and originally used for computing the similarity between two documents in text processing. Here documents are considered as rating vectors. Given two rating vectors $u_1 = (r_{11}, r_{12}, \ldots, r_{1n})$ and $u_2 = (r_{21}, r_{22}, \ldots, r_{2n})$, the function F of u_1 and u_2 is defined as follows:

$$F(u_1,u_2) = \frac{\sum_{j=1}^{n} A\left(r_{1j},r_{2j}\right)}{\sum_{j=1}^{n} B\left(r_{1j},r_{2j}\right)} \quad (31)$$

where:

$$A\left(r_{1j},r_{2j}\right) = \begin{cases}0.5\left(1+\exp\left(-\left(\frac{r_{1j}-r_{2j}}{\sigma_j}\right)^2\right)\right) \\ \text{if both } r_{1j} \text{ and } r_{2j} \text{ non-missing} \\ 0 \text{ if both } r_{1j} \text{ and } r_{2j} \text{ missing} \\ -\lambda \text{ otherwise}\end{cases}$$

$$B\left(r_{1j},r_{2j}\right) = \begin{cases}0 \text{ if both } r_{1j} \text{ and } r_{2j} \text{ missing} \\ 1 \text{ if both } r_{1j} \text{ and } r_{2j} \text{ non-missing}\end{cases}$$

where λ is the pre-defined number and σ_j is the standard deviation of rating values belonging to field j (item j). After setting several values for λ parameter including (1) value as authors claimed its being the best setting, we experimentally found that the (0.5) was the best value of λ to derive the best results for this measure. SMTP measure is defined as follows:

$$\text{SMTP}(u_1,u_2) = \frac{F(u_1,u_2)+\lambda}{1+\lambda} \quad (32)$$

4. Methodology

4.1. Problem statement and motivations

As drawn in the above sections, the data sparsity problem has long been shown to have a great impact on the behavior of similarity measures, and then on RS accuracy as a result [37]. This problem emerges from the lowest number of rated items comparing with the required-to-be-predicted items. It is often that the rating overlap between a specific pair of users is extremely small or even completely absent. In consequence, under such conditions, CF has been unable to offer the desired recommendations. This problem is also named Multiple-interest and Multiple-content recommendations in e-commerce [1,10,40], though. In conclusion, it has been noted that CF techniques, including traditional and the recently-founded, suffer inadequacies to obtain the desired accuracy when it comes to handling the recommendations amid an extremely small rate of co-rated items [1,28,32]. Hence, a broad space for CF performance enhancement is still available chiefly under the umbrella of data sparsity problem.

CF literature, in its turn, is full of similarity measures which have come to address CF problems including the cold-start dilemma. The shortcomings and limitations of these measures (including the traditional ones) have been thoroughly examined [10,11,31,41]. On the other extreme, it is also worth drawing the motivations that drive us toward establishing this work. As a result of our in-depth investigation for a dozens of earlier CF studies, we find that the data sparsity problem (also known cold-start problem [8]) has not yet dully and effectively tackled as no influential solutions have been recorded except for some studies like [28–33,35–38,41]. The lowest number of rated items versus the required-to-be-predicted items is the cause for data sparsity. As a result, under such condition, CF has been unable to offer the desired recommendations. Moreover, the proposed measures including the most effective ones like NHMS, PIP and MSDJ suffer the design complexity.

Therefore, to go in parallel with those studies that have been shown effective to handle the data sparsity problem, our work undertakes the challenges and introduces three new simple-yet-effective similarity measures. The pivotal point is to design and present effective similarity measures that can efficiently and effectively deal with the sparsity problem so the highly-accurate recommendation can be secured. These measures have treated co-rated items while taking non-co-rated items at the same time when making both estimation and recommendation. In other words, the proposed measures have used all the rating vectors so

that a better prediction is made, and then highly-accurate recommendations are generated. Based on the experimental results, our proposed measures have been shown to outperform almost all CF techniques which have been investigated in this study (almost 30 measures) in terms of effectiveness (including Accuracy, MAE, MSE, Precision, Recall, and F1) and efficiency that includes time complexity for top performers. From experimental results, the proposed measures have been seen maximally effective offering the desired recommendations with the lowest computation time. Finally and most importantly, contrary to all previous works, our work performs an extensive experimental analysis for the performance of all 30 similarity measures on user-based model so a comprehensive experimental guide is provided.

4.2. Proposed similarity measures

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). In the binary representation, r_{ij} is converted into 1 if it is non-missing (rated) and otherwise, r_{ij} is converted into 0. Let N_{12} be the number of common values "1" in both u_1 and u_2. Let N be the total number of all items under consideration; in this case, $N = n$. Let N_1 and N_2 be the numbers of values "1" of u_1 and u_2, respectively, F would be the sum of number of differences between u_1 and u_2. For example, the fact that $r_{11} = 0$ and $r_{21} = 1$, would contribute one difference to F. In next sub-sections, we define our proposed measures in a very simplistic manner as follows;

4.2.1. SMD

The *SMD measure* is defined in binary representation as follows:

$$\text{SMD} = \frac{\left(1 - \frac{F}{N}\right) + \left(\frac{2N_{12}}{N_1+N_2}\right)}{2} \tag{33}$$

While $\left(1 - \frac{F}{N}\right)$ seeks to carefully calculate the similarity between each two rating vectors based on discovering the latent differences between both vectors, $\left(\frac{2N_{12}}{N_1+N_2}\right)$ seeks to emphasize the similarity of vectors through walking through the shared features of both vectors. In doing so, every part of this measure is being an indispensable complement to the other part so the exact desired similarity is recorded, and so the RS performance is effectively promoted as drawn in the results and discussion sections. Both parts are simply designed without introducing any weighting factors. Among these factors, reducing or eliminating the full reliant on the co-rated items, making full use of all rated and non-rated items which led to similarities being treated as symmetric and asymmetry at the same time.

4.2.2. HSMD

Let I_1 or I_2 be sets of indices of items that user 1 or user 2 rates, respectively. HSMD is a developed variation of SMD, but the difference is that HSMD deals with the numerical representation of ratings directly without the need for the binary conversion as done with SMD. HSMD is formulated as follows;

$$\text{HSMD} = 1 - \frac{R_1 R_2 + 1}{G} \tag{34}$$

where, R_1 (R_2) is the sum of non-missing values r_{1j} (r_{2j}) of u_1 (u_2) such that the respective values r_{2j} (r_{1j}) are missing.

$$R_1 = \sum_{\substack{r_{1j}\text{non-missing} \\ r_{2j}\text{missing}}} r_{1j} = \sum_{j \in I_1 \setminus I_2} r_{1j}$$

$$R_2 = \sum_{\substack{r_{2j}\text{non-missing} \\ r_{1j}\text{missing}}} r_{2j} = \sum_{j \in I_2 \setminus I_1} r_{2j}$$

Note, the notation "\" denotes complement operator in the set theory. G is the product of two sums of the non-missing values for both r_1 and r_2.

$$G = \left(\sum_{r_{1j}\text{non-missing}} r_{1j}\right)\left(\sum_{r_{2j}\text{non-missing}} r_{2j}\right) = \left(\sum_{j \in I_1} r_{1j}\right)\left(\sum_{j \in I_2} r_{2j}\right)$$

The next examples draw a brief clarification into mechanism of these two measures. For example, given two rating vectors $u_1 = (r_{11} = 2, r_{12} = 5, r_{13} = 7, r_{14} = 8, r_{15} = ?, r_{16} = 9)$ and $u_2 = (r_{21} = 9, r_{22} = ?, r_{23} = ?, r_{24} = 6, r_{25} = 5, r_{26} = 1)$. Binary representations of these two vectors are (1, 1, 1, 1, 0, 1) and (1, 0, 0, 1, 1, 1). Hence, SMD measure is calculated according to Eq. (35), as follows:

$$\text{SMD} = \frac{\left(1 - \frac{3}{6}\right) + \left(\frac{2*3}{5+4}\right)}{2} = 0.58$$

According to HSMD measure, we have $R_1 = 5 + 7 = 12$, $R_2 = 5$, and $G = (2+5+7+8+9) * (9+6+5+1) = 651$. Hence, HSMD measure is calculated according to Eq. (36), as follows:

$$\text{HSMD} = 1 - \frac{12 * 5 + 1}{651} = 0.91$$

Finally, to show the applicability of integrating these measures with Jaccard to give a good measures, we proposed HSMDJ measure as a combination of both HSMD and Jaccard measures. HSMDJ is specified by Eq. (35).

$$\text{HSMDJ}\,(u_1, u_2) = \text{HSMD}\,(u_1, u_2) * \textit{Jaccard}\,(u_1, u_2) \tag{35}$$

4.2.3. Triangle based Cosine Similarity Measure (TA)

Cosine measure is an effective measure; however, it has a drawback represented in its value being high when there are two endpoints of two vectors that are far from each other according to Euclidean distance. This negative effect of Euclidean distance decreases the accuracy of cosine similarity [42]. Therefore, an advanced triangle-based cosine measure (TA) is proposed to cover the deficit and present an advanced version of cosine.

TA measure uses the ratio of basic triangle area to the whole triangle area as a reinforced factor for Euclidean distance so that it can alleviate the negative effect of Euclidean distance whereas it integrates and keeps both simplicity and effectiveness of both cosine measure and Euclidean distance in making the similarity of two vectors. TA is defined by Eq. (36):

$$\begin{aligned}
u_1 \cdot u_2 \geq 0\text{: TA}\,(u_1, u_2) &= \begin{cases} \frac{(u_1 \cdot u_2)^2}{|u_1|(|u_2|)^3} & \text{if } |u_1| \leq |u_2| \\ \frac{(u_1 \cdot u_2)^2}{(|u_1|)^3 |u_2|} & \text{if } |u_1| > |u_2| \end{cases} \\
u_1 \cdot u_2 < 0\text{: TA}\ (u_1, u_2) &= \begin{cases} \frac{u_1 \cdot u_2}{(|u_2|)^2} & \text{if } |u_1| \leq |u_2| \\ \frac{u_1 \cdot u_2}{(|u_1|)^2} & \text{if } |u_1| > |u_2| \end{cases} \\
u_1 \cdot u_2 &= \textstyle\sum_{j \in I_1 \cap I_2} r_{1j} r_{2j} \\
|u_1| &= \sqrt{\textstyle\sum_{j \in I_1 \cap I_2} (r_{1j})^2} \\
|u_2| &= \sqrt{\textstyle\sum_{j \in I_1 \cap I_2} (r_{2j})^2}
\end{aligned} \tag{36}$$

Let TAJ denotes the combined measure, which combines TA and Jaccard. TAJ measure is defined as follows:

$$\text{TAJ}\,(u_1, u_2) = \text{TA}\,(u_1, u_2) * \textit{Jaccard}\,(u_1, u_2) \tag{37}$$

Let r_m be the median of rating values, TA measure is normalized to produce TAN measure as follows:

$$\begin{aligned} &\mathrm{TAN}(u_1, u_2) = \mathrm{TA}(u_1, u_2) \\ &u_1 \cdot u_2 = \sum_{j \in I_1 \cap I_2} (r_{1j} - r_m)(r_{2j} - r_m) \\ &|u_1| = \sqrt{\sum_{j \in I_1 \cap I_2} (r_{1j} - r_m)^2} \\ &|u_2| = \sqrt{\sum_{j \in I_1 \cap I_2} (r_{2j} - r_m)^2} \end{aligned} \tag{38}$$

By combined TAN with Jaccard, TAN becomes TANJ measure as follows:

$$\mathrm{TANJ}(u_1, u_2) = TAN(u_1, u_2) * Jaccard(u_1, u_2) \tag{39}$$

By convention, based on Eq. (36)–(39), TA family includes TA, TAJ, TAN, and TANJ.

5. Experimental setup

This section covers all tools using which all similarity measures are extensively evaluated. These tools are represented in the machine and environment description as an experimental setup, dataset and the evaluation process including both processes of estimation and recommendation as well as evaluation metrics. Finally, all results (of all evaluation metrics) are drawn in detail. Table 2 displays the machine and environment description using which the experiments has been made.

5.1. Movielens dataset

The dataset Movielens-100 K [11,33,43,44] has 100,000 ratings from 943 users on 1682 movies (items), and every rating ranges from 1 to 5. In this data set, the range of rating is 1–5, and the sparseness is: 1 - 10000/ $(943 * 1682) = 0.936953$.

5.2. Estimation and recommendation processes

In our experiments, dataset Movielens is divided into 5 folders and each folder included a training set and testing set. Training set and testing set in the same folder are disjoint sets. The ratio of the testing set over the whole dataset has been set based on the proposed testing parameter r which ranges from 0.1 to 0.9. For instance, if $r = 0.1$, the testing set covers 10% of the dataset, which means that the testing set has $10{,}000 = 10\% * 100{,}000$ ratings, and the training set has 90,000 ratings. In our experimental design, parameter r has nine values 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9. We discovered that t**he smaller *r* is, the more accurate measures are because the training set gets large if *r* gets smaller.** Furthermore, in the KNN, **we set the neighborhood threshold to four values (5, 20, 50 and 100) for Film Trust and Movielens-100K datasets and (5 and 20) for Movielens- 1 M.** Although a higher threshold may or may not improve the RS accuracy, we tested four **neighborhood** threshold values. It is, however, possible that a higher threshold would decrease RS accuracy considering the fact that we evaluated all experiments (for all 30 similarity measures) on the KNN with 5-fold cross-validation.

On the other hand, popular metrics to assess CF algorithms are the mean absolute error (**MAE**), **recall**, and **precision**. The Quality of CF algorithms like KNN algorithm depends on both **estimation and recommendation** as well. On one hand, the estimation ability is the ability to estimate or predict accurately the missing values. The recommendation, on the other hand, is the ability to provide the list of recommended items which is as suitable as possible for users. In our work, the threshold is calculated as the average of the minimum rating and maximum rating values. For example, if minimum value is 1 and maximum value is 5, the threshold is $(1 + 5)/2 = 3$. **Thus, in our work, to determine whether an item should be recommended, any item whose rating value is greater than the threshold – greater than 3 in this example – the item is relevant (favorite), and recommended as a result.** On the other extreme, as a novelty of our work, we do not follow the same pattern taken by the previous research that only focused on the recommendation tasks with metrics MAE, precision, and recall. Instead, we divide our tests into two processes such as estimation and recommendation as follows:

- In the estimation process, assuming given the tested vector $u_t = (v_1 = 1, v_2 = 2, v_3 = 3)$ that has three items, it is made empty as empty vector $u' = (v_1 = ?, v_2 = ?, v_3 = ?)$ with the missing values indicated by question marks. Later, the KNN algorithm was applied, and, the setting of the similarity and neighborhood thresholds was zero and five respectively, with the drawn above metrics for the task of predicting (estimating) the missing values. As a result, the predictive vector (estimated vector) was obtained, for example, $u_p = (v_1 = 2, v_2 = 3, v_3 = 4)$ that has three estimated items. By comparing u_t and u_p, the MAE metric was then used to evaluate the estimation process. For instance, MAE metric was (|2-1| + |3-2| + | 4-3|) / 3 = 1.
- In the recommendation process, assuming the given tested vector $u_t = (v_1 = 1, v_2 = 2, v_3 = 3)$, the KNN algorithm was asked for providing the recommended list (recommended vector) of items. Supposed the recommended vector was $u_s = (v_2 = 5, v_4 = 4, v_3 = 4, v_5 = 2)$ within the rating range {1, 2, 3, 4, 5}. By comparing u_t and u_s, the precision and recall metrics were calculated to evaluate the recommendation process. For instance, given u_t and u_s, the precision was 0.25 and recall was 1. The way to calculate precision and recall will be described in detail down below.

Hence, the different metrics (MAE, recall, precision) are commonly used for different evaluation processes (estimation and recommendation). This independent evaluation allowed us to test measures more objectively, in which the estimation process focused on CF accuracy and the recommendation process focused on CF quality. In general, MAE is used for the estimation whereas the recall and precision are used for the recommendation process. It is then necessary to describe the metrics MAE, precision, and recall. MAE [45] is calculated by next Eq. (40), in which n is the total number of the estimated items while v'_j and v_j are the predictive rating and the true rating of item j, respectively.

$$\mathrm{MAE} = \frac{1}{n} \sum_{j=1}^{n} \left| v'_j - v_j \right| \tag{40}$$

The smaller the MAE, the more accurate the measure, and so the better behavior the algorithm has. On the other hand, the Precision and recall are the quality metrics that measure quality of the recommended list — how much the recommendation list reflects the user's preferences. The larger the quality metric, the better the algorithm. An item is relevant if its rating is larger than the average rating. For example, within rating range {1, 2, 3, 4, 5}, the average rating is $3 = (1 + 5)/2$. An item is selective if it is recommended to users. Let N_r be the number of relevant items and let N_s be the number of selective items. Let N_{rs} be the number of items which are relevant and selective. According to Eq. (41), the precision is the ratio of N_{rs} to N_s and the recall is the ratio of N_{rs} to N_r [45]. In other words, the precision is the probability that

Table 2
Machine and environment description.

Tool	Specification
Language	Java: Java ™ SE Runtime Environment version 1.8.0_60-b27, Java HotSpot(TM) 64-Bit Server VM version 25.60-b23, Class version 52.0, Vendor "Oracle Corporation" at http://java.oracle.com/
OS	Windows 8.1, AMD 64, version 6.3
Memory	Memory (VM): Allocated memory = 1023.50 MB, Free memory = 335.83 MB, Max memory = 1023.50 MB
CPU	Intel64 Family 6 Model 76 Stepping 3, Genuine Intel, AMD64, the number of processors is 2
Dataset	Movielens-100K

selective item is relevant, and the recall is the probability that the relevant item is selective.

$$\text{Precision} = \frac{N_{rs}}{N_s} \qquad \text{Recall} = \frac{N_{rs}}{N_r} \tag{41}$$

For example, given the tested vector $u_t = (v_1 = 1, v_2 = 2, v_3 = 3)$, and the recommended vector $u_s = (v_2 = 5, v_4 = 4, v_3 = 4, v_5 = 2)$, we have $N_s = |u_s| = 4$. Because the vector u_t has only one relevant item 3 ($v_3 = 3$), we have $N_r = 1$. We also have $N_{rs} = 1$ because only one relevant item 3 exists in both u_t and u_s. Mathematically, we have Precision $= N_{rs}/N_s = 1/4 = 0.25$ and Recall $= N_{rs}/N_r = 1/1 = 1$.

The problem in the recommendation is how to determine the number of recommended items which is denoted C as the length of the recommended vector. By convention, C suggests the recommendation count. The count variable C cannot be too small or too large. If it is too small, the evaluation is inaccurate. Otherwise, if it is too large, the evaluation task will run slowly. Some researches fixed the number whereas other researches changed the number over some values such as 10, 20, and 100. In our work, however, we proposed a method to dynamically determine C based on the dataset with the purpose that N will be more accurate and objective. The proposed method is dynamic and takes advantage of the so-called sparse-relevant ratio. This ratio is the ratio of the count of relevant ratings to the count of cells considering that the count of cells is a product of user number and item number, which is the size of the rating matrix. Recall that a relevant rating is larger than the average rating and the count of cells is the sum of both the count of rating values and the count of missing values combined. Eq. (44) specifies the sparse-relevant ratio which is denoted by sr.

$$sr = \text{the} - \text{count} - \text{of} - \text{relevant} - \text{ratings} / (|\boldsymbol{U}| * |\boldsymbol{V}|) \tag{42}$$

where $|\boldsymbol{U}|$ is the number of users and $|\boldsymbol{V}|$ is the number of items. We calculated recommendation count C dynamically according to both the dataset and each rating vector u_i. Let $C(u_i)$ be the recommendation count for user i, which means that KNN algorithms will recommend at least $C(u_i)$ items to user i. Eq. (45) specifies $C(u_i)$.

$$C(u_i) = sr * (T - |I_i|) \tag{43}$$

where T is the number of items given that every item included in T is rated by at least one user. Of course, T is smaller than or equal to the number of users $|\boldsymbol{U}|$. In its turn, $|I_i|$ is the number of items rated by user i. The quantity $|I_i|$ is not redundant because the real recommendation systems always recommend the user's items that she/he does not either know or rate yet. If $|I_i|$ is smaller than $T(|I_i| < T)$, $C(u_i)$ can be calculated as follows:

$$C(u_i) = sr * T \tag{44}$$

Recall that in our experiments, dataset Movielens is divided into 5 folders and each folder has one training set and one testing set. In Eq. (43), for each folder, T and the sparse-relevant ratio sr are calculated on the training set but $|I_i|$ is determined on testing set. For example, suppose one among 5 folders, divided from Movielens, has a training set d_1 and testing set t_1. The number of users in d_1 is 943 and the number of items in d_1 is 1,584. Because of that every item in d_1 is rated by at least one user, we have $T = 1{,}584$. Training set d_1 has 50,000 rating values but only 27,712 rating values are relevant. So, the sparse-relevant ratio is $sr = 27{,}712 / (943 * 1584) \approx 1.86\%$. Suppose it is necessary to make the recommendation on user rating vector u_{12} (in testing set t_1) which has 23 rating values. Hence, the recommendation count for user 12 is $C(u_{12}) = 1.86\% * (1{,}584 - 23) \approx 29$.

6. Results

To perform experiments, we first recall that we already have the next similarity measures with their families as follows:

- SMD family includes measures: SMD, HSMD, and HSMDJ.
- Cosine family includes measures: Cosine, COJ, CON, COD, CosineJ.
- Pearson family includes measures: Pearson, WPC, SPC, PearsonJ.
- MSD family includes measures: MSD and MSDJ.
- TA family includes measures: TA, TAJ, TAN, and TANJ.
- Individual measures which are: Jaccard, SRC, NHSM, BCF, SMTP, PC, PIP, CjacMD, TMJ, Feng, and Mu.

We have tested all similarity measures with the user-based KNN algorithm on the given rating matrix. The KNN algorithm implies user-based KNN algorithm and the rating matrix implies user-based rating matrix. With setting the similarity threshold to zero (so all users have been involved fairly) and the neighborhood threshold to K = (5, 20, 50 and 100) for prediction, suppose that the KNN algorithm finds out k neighbors of the active item, the missing value r_{aj} of r_a is computed as follows:

$$r_{aj} = \bar{r}_a + \frac{\sum_{i=1}^{k} (r_{ij} - \bar{r}_i)\, sim(r_a, r_i)}{\sum_{i=1}^{k} |sim(r_a, r_i)|} \tag{45}$$

where $\bar{r}_a$ and $\bar{r}_i$ are the mean values of r_a and r_i, respectively. Movielens dataset was divided into 5 folders and each folder included a training set and testing set. Each folder had its own tested measures, and consequently, the tested measures, in the next tables, made an average over 5 folders. The next two subsections hold the results of initial evaluation from which the best nine similarity measures are included in further evaluation. Each Table, among next Tables 3–9, show results of the corresponding evaluation metric of all tested measures on all values of $r = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8$, and 0.9 **within the estimation/recommendation process**. The last column, in each Table, shows the averaged results of corresponding metric over all values of r and the shaded cells, in gray color, indicates the best values. By convention, we define that the pre-eminent measures (dominant measures) are those in the top-5 lists.

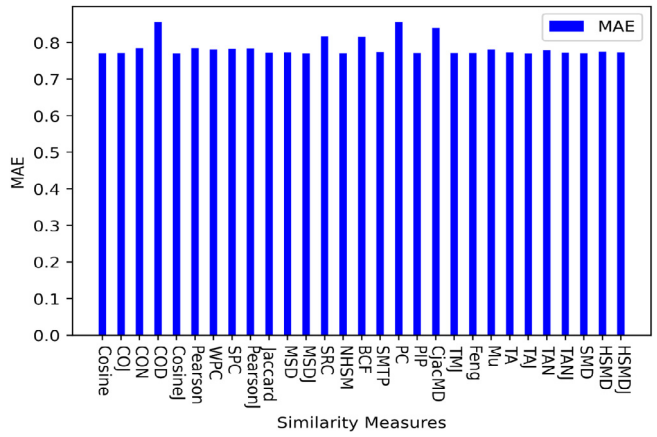

Fig. 1. Similarity measures behavior on MAE metric within the estimation process — Averaged results.

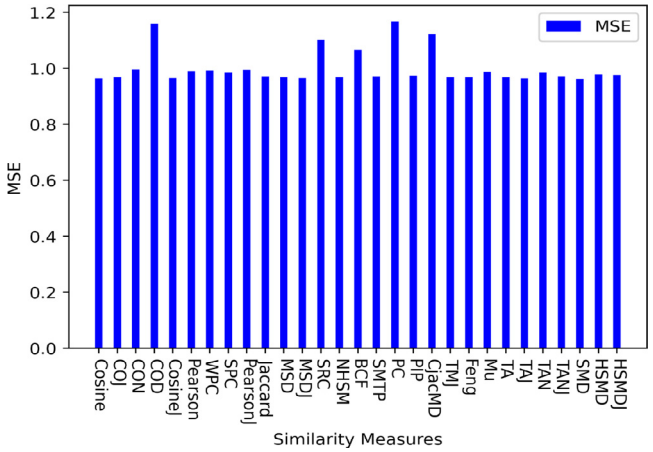

Fig. 2. Similarity measures behavior on MSE metric within the estimation process — Averaged results.

6.1. Estimation process

The Top-5 measures, according to MAE metric within the estimation process (Table 3, Fig. 1), are TAJ, MSDJ, CosineJ, SMD, and NHSM whose average MAE metrics are 0.7699, 0.7703, 0.7704, 0.7709, and 0.7712, respectively. Shortly, the dominant orders of our measures, TA, TAJ, TAN, TANJ, SMD, HSMD, and HSMDJ, are 13rd, 1st, 20th, 11st, 4th, 18th, and 14th among all measures, in Table 4, respectively. Fig. 1, on the other hand, draws a concise observation into the averaged results of MAE metric.

Regarding the estimation process, some popular metrics which are different from MAE are the mean squared error (MSE) and the correlation coefficient (R). MSE, Table 4, is calculated by Eq. (48), in which n is the total number of the estimated items while v_j' and v_j are the predictive and true ratings of item j, respectively. Given the predictive vector v' and true vector (tested vector) v, MSE is computed as follows;

$$\mathrm{MSE} = \frac{1}{n}\sum_{j=1}^{n}\left(v'_j - v_j\right)^2 \tag{46}$$

The smaller MSE is, the more accurate the measure is, and so the better the algorithm is. The R metric [46], Table 5, is used to evaluate the correlation between the predictive vector v' and true vector v. The larger R is, the better the measure is. Eq. (49) specifies R metric as follows;

$$\mathrm{R} = \frac{\sum_{j=1}^{n}\left(v'_j - \overline{v'}\right)\left(v_j - \overline{v}\right)}{\sqrt{\sum_{j=1}^{n}\left(v'_j - \overline{v'}\right)^2}\sqrt{\sum_{j=1}^{n}\left(v_j - \overline{v}\right)^2}} \tag{47}$$

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

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

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

The Top-5 measures, according to MSE metric within the recommendation process (Table 4, Fig. 2), are SMD, Cosine, TAJ, MSDJ, and CosineJ whose average recall metrics are 0.9618, 0.9633, 0.9637, 0.9647, and 0.9649 respectively. Our SMD measure is in the top-5 list of MSE metric. Shortly, the dominant orders of our measures, TA, TAJ, TAN, TANJ, SMD, HSMD, and HSMDJ are 7th, 3rd, 20th, 14th, 1st, 17th and 16th among all measures of MSE metric, respectively. Fig. 2, on the other hand, draws a concise observation into the averaged results of MSE metric. It is abundantly obvious that the bigger r is, the bigger MSE value is, and vice versa. Interestingly, MSE is stable for almost all measures from $r = 0.1$ to $r = 0.7$. The best performance for almost all measures was observed when $r = 0.7$, $r = 0.8$, and $r = 0.9$ consecutively, though.

The Top-5 measures, according to R metric within the recommendation process (Table 5, Fig. 3), are SMD, Cosine, TA, TAJ, and MSDJ whose average R metrics are 0.3686, 0.3663, 0.3616, 0.3614, and 0.3598. Our SMD measure is in the top-5 list of R metric. Shortly, the dominant orders of our measures, TA, TAJ, TAN, TANJ, SMD, HSMD, and HSMDJ, are 3rd, 4th, 17th, 10th, 1st, 15th and 18th among all measures of R metric, respectively. Fig. 3, on the other hand, draws a concise observation into the averaged results of R metric.

6.2. Recommendation process

Tables 6–8 show the results of Precision, recall and F1 metrics respectively.

The Top-5 measures, according to the precision metric within the recommendation process (Table 6, Fig. 4), are PIP, TANJ, NHSM , CON, TAJ whose average precision metrics are 0.0320, 0.0319, 0.0319, 0.0319, and 0.0319, respectively. Our measures TANJ and TAJ are in the top-5 list. Shortly, the dominant orders of our measures, TA, TAJ, TAN, TANJ, SMD, HSMD, and HSMDJ, are 8th, 5th, 10th, 3rd, 19th, 29th, and 24th among all measures in Table 9, respectively. Fig. 4, on the other hand, draws a concise observation into the averaged results of the precision metric. It is clear that the bigger r is, the bigger the precision is, and vice versa. Interestingly, the best performance for almost all measures observed when $r = 0.8$ and $r = 0.9$ consecutively, though.

The Top-5 measures, according to the recall metric within the recommendation process (Table 7, Fig. 5), are SPC, Pearson, WPC, PearsonJ, and SMD whose average recall metrics are 0.9138, 0.9133, 0.9132, 0.9130, and 0.9105, respectively. Our SMD measure is in the top-5 list of recall metric. It is interesting that four of the top-5 list given recall metric are SPC, Pearson, WPC, and PearsonJ which belong to Pearson family. Hence, Pearson family is pre-eminent measures over the recall metric. Shortly, the dominant orders of our measures, TA, TAJ, TAN, TANJ, SMD, HSMD, and HSMDJ, are 23rd, 22nd, 28th, 29th, 5th, 12nd and

Table 3
MAE metric within the estimation process.

Measure	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	Average (MAE)
Cosine	0.7532	0.7551	0.7560	0.7593	0.7630	0.7654	0.7736	0.7905	0.8255	0.7713
COJ	0.7457	0.7483	0.7494	0.7533	0.7572	0.7610	0.7710	0.7937	0.8681	0.7720
CON	0.7469	0.7503	0.7532	0.7582	0.7657	0.7738	0.7889	0.8227	0.8985	0.7842
COD	0.8224	0.8271	0.8307	0.8357	0.8441	0.8526	0.8621	0.8821	0.9423	0.8555
CosineJ	0.7459	0.7485	0.7496	0.7537	0.7577	0.7615	0.7712	0.7921	0.8537	0.7704
Pearson	0.7395	0.7462	0.7519	0.7611	0.7734	0.7882	0.8091	0.8435	0.8473	0.7845
WPC	0.7312	0.7365	0.7405	0.7480	0.7581	0.7721	0.7947	0.8337	0.9202	0.7817
SPC	0.7388	0.7452	0.7506	0.7592	0.7708	0.7848	0.8051	0.8399	0.8490	0.7826
PearsonJ	0.7311	0.7375	0.7427	0.7510	0.7624	0.7766	0.7992	0.8379	0.9173	0.7840
Jaccard	0.7465	0.7491	0.7502	0.7543	0.7583	0.7620	0.7717	0.7939	0.8651	0.7723
MSD	0.7529	0.7549	0.7558	0.7591	0.7627	0.7651	0.7732	0.7901	0.8471	0.7734
MSDJ	0.7457	0.7484	0.7495	0.7536	0.7575	0.7613	0.7709	0.7919	0.8538	0.7703
SRC	0.7429	0.7434	0.7426	0.7453	0.7536	0.7682	0.8127	0.9394	1.1041	0.8169
NHSM	0.7410	0.7441	0.7452	0.7498	0.7545	0.7599	0.7728	0.8006	0.8729	0.7712
BCF	0.7984	0.8004	0.8011	0.8035	0.8061	0.8095	0.8159	0.8316	0.8658	0.8147
SMTP	0.7533	0.7551	0.7560	0.7592	0.7629	0.7652	0.7733	0.7903	0.8478	0.7737
PC	0.8229	0.8279	0.8317	0.8365	0.8446	0.8522	0.8612	0.8805	0.9443	0.8558
PIP	0.7424	0.7451	0.7466	0.7510	0.7556	0.7606	0.7726	0.7989	0.8723	0.7717
CjacMD	0.7804	0.7885	0.7946	0.8054	0.8193	0.8310	0.8554	0.8975	0.9795	0.8391
TMJ	0.7463	0.7489	0.7500	0.7541	0.7581	0.7618	0.7714	0.7935	0.8647	0.7721
Feng	0.7454	0.7479	0.7489	0.7528	0.7566	0.7605	0.7704	0.7928	0.8673	0.7714
Mu	0.7388	0.7425	0.7456	0.7517	0.7612	0.7725	0.7917	0.8262	0.9027	0.7814
TA	0.7518	0.7538	0.7547	0.7581	0.7618	0.7643	0.7726	0.7901	0.8487	0.7729
TAJ	0.7449	0.7475	0.7486	0.7527	0.7568	0.7606	0.7704	0.7920	0.8552	0.7699
TAN	0.7467	0.7503	0.7527	0.7586	0.7654	0.7713	0.7846	0.8084	0.8723	0.7789
TANJ	0.7379	0.7416	0.7439	0.7501	0.7568	0.7633	0.7779	0.8050	0.8734	0.7722
SMD	0.7524	0.7546	0.7557	0.7592	0.7631	0.7657	0.7737	0.7885	0.8253	0.7709
HSMD	0.7481	0.7505	0.7514	0.7550	0.7588	0.7622	0.7719	0.7944	0.8812	0.7748
HSMDJ	0.7427	0.7458	0.7470	0.7516	0.7562	0.7614	0.7739	0.8017	0.8791	0.7733

Table 4
MSE metric within the estimation process.

Measure	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	Average (MSE)
Cosine	0.9114	0.9193	0.9235	0.9304	0.9394	0.9476	0.9681	1.0162	1.1135	0.9633
COJ	0.8971	0.9060	0.9104	0.9184	0.9285	0.9390	0.9638	1.0255	1.2393	0.9698
CON	0.8947	0.9050	0.9124	0.9245	0.9427	0.9632	1.0019	1.0939	1.3155	0.9949
COD	1.0676	1.0800	1.0898	1.1031	1.1233	1.1459	1.1739	1.2349	1.4165	1.1594
CosineJ	0.8973	0.9064	0.9111	0.9193	0.9294	0.9400	0.9643	1.0209	1.1956	0.9649
Pearson	0.8810	0.8970	0.9107	0.9304	0.9590	0.9930	1.0438	1.1355	1.1640	0.9905
WPC	0.8650	0.8786	0.8892	0.9048	0.9289	0.9600	1.0137	1.1146	1.3721	0.9919
SPC	0.8792	0.8947	0.9076	0.9261	0.9531	0.9851	1.0344	1.1267	1.1678	0.9861
PearsonJ	0.8641	0.8798	0.8922	0.9097	0.9362	0.9680	1.0224	1.1235	1.3499	0.9940
Jaccard	0.8987	0.9078	0.9125	0.9208	0.9308	0.9415	0.9658	1.0262	1.2313	0.9706
MSD	0.9109	0.9190	0.9231	0.9300	0.9388	0.9468	0.9671	1.0150	1.1794	0.9700
MSDJ	0.8971	0.9062	0.9108	0.9190	0.9291	0.9396	0.9638	1.0206	1.1959	0.9647
SRC	0.8871	0.8872	0.8866	0.8933	0.9150	0.9572	1.0799	1.4585	1.9472	1.1013
NHSM	0.8878	0.8977	0.9025	0.9120	0.9239	0.9383	0.9709	1.0444	1.2484	0.9695
BCF	1.0126	1.0222	1.0260	1.0332	1.0405	1.0518	1.0709	1.1165	1.2249	1.0665
SMTP	0.9116	0.9195	0.9237	0.9304	0.9394	0.9473	0.9677	1.0158	1.1812	0.9707
PC	1.0813	1.0936	1.1030	1.1138	1.1324	1.1505	1.1741	1.2295	1.4196	1.1664
PIP	0.8940	0.9031	0.9085	0.9174	0.9300	0.9432	0.9740	1.0444	1.2526	0.9741
CjacMD	0.9645	0.9871	1.0048	1.0299	1.0660	1.0966	1.1603	1.2773	1.5198	1.1229
TMJ	0.8983	0.9073	0.9120	0.9202	0.9303	0.9408	0.9650	1.0250	1.2299	0.9699
Feng	0.8962	0.9048	0.9091	0.9171	0.9269	0.9376	0.9626	1.0237	1.2380	0.9684
Mu	0.8812	0.8920	0.8998	0.9121	0.9337	0.9596	1.0044	1.0949	1.3152	0.9881
TA	0.9085	0.9164	0.9205	0.9276	0.9367	0.9449	0.9657	1.0151	1.1833	0.9687
TAJ	0.8952	0.9041	0.9087	0.9170	0.9274	0.9380	0.9627	1.0206	1.1993	0.9637
TAN	0.8981	0.9086	0.9168	0.9305	0.9470	0.9634	0.9967	1.0643	1.2514	0.9863
TANJ	0.8811	0.8911	0.8985	0.9127	0.9288	0.9455	0.9819	1.0556	1.2523	0.9719
SMD	0.9102	0.9187	0.9231	0.9304	0.9396	0.9475	0.9673	1.0064	1.1131	0.9618
HSMD	0.9023	0.9107	0.9151	0.9225	0.9320	0.9417	0.9663	1.0291	1.2821	0.9780
HSMDJ	0.8922	0.9022	0.9072	0.9166	0.9282	0.9423	0.9735	1.0475	1.2688	0.9754

13rd among all measures in Table 7, respectively. Fig. 5 draws a concise observation into the averaged results of the recall metric.

From the drawn above results, of all metrics MAE, MSE, R, Precision, and Recall, it seems to be a challenging task to decide which measures are the best. Recalling that the estimation

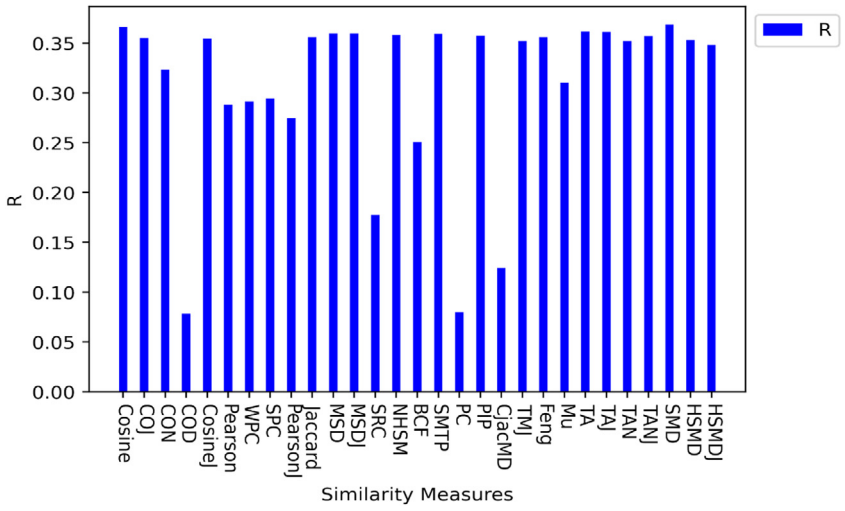

Fig. 3. Similarity measures behavior on R metric within estimation process — Averaged results.

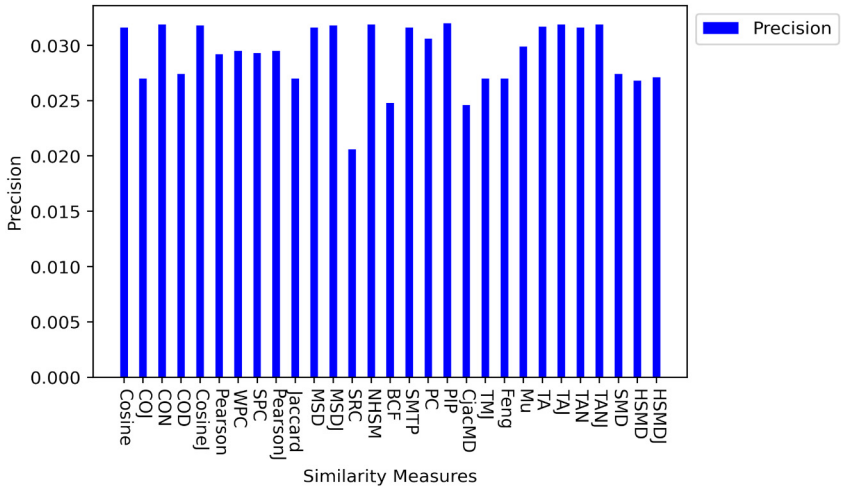

Fig. 4. Similarity measures behavior on precision metric within the recommendation process — Averaged results.

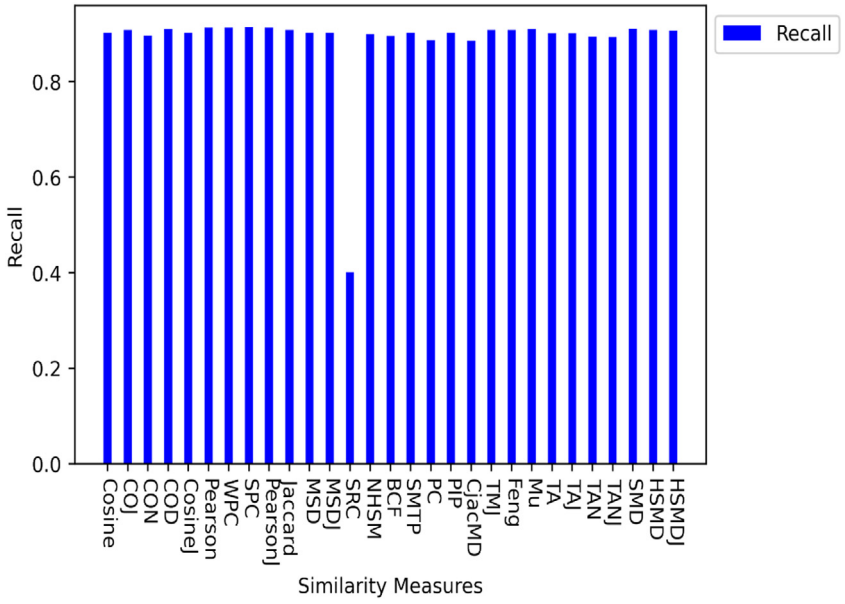

Fig. 5. Similarity measures behavior on recall metric within the recommendation process — Averaged results.

Table 5
R metric within the estimation process.

Measure	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	Average (R)
Cosine	0.4185	0.3780	0.3829	0.3806	0.3784	0.3736	0.3580	0.3261	0.3004	0.3663
COJ	0.3881	0.3880	0.3915	0.3883	0.3855	0.3771	0.3571	0.3169	0.2051	0.3553
CON	0.4339	0.3865	0.3851	0.3741	0.3525	0.3265	0.2879	0.2204	0.1403	0.3230
COD	0.1810	0.1147	0.1087	0.0911	0.0678	0.0537	0.0418	0.0288	0.0178	0.0784
CosineJ	0.3871	0.3856	0.3885	0.3843	0.3817	0.3733	0.3531	0.3145	0.2228	0.3545
Pearson	0.4355	0.3699	0.3557	0.3260	0.2905	0.2460	0.1931	0.1250	0.2530	0.2883
WPC	0.4519	0.3939	0.3821	0.3597	0.3265	0.2828	0.2236	0.1448	0.0569	0.2914
SPC	0.4378	0.3749	0.3620	0.3352	0.3013	0.2574	0.2031	0.1314	0.2476	0.2945
PearsonJ	0.4109	0.3823	0.3668	0.3424	0.3068	0.2648	0.2070	0.1342	0.0578	0.2748
Jaccard	0.4273	0.3841	0.3870	0.3828	0.3803	0.3719	0.3516	0.3120	0.2058	0.3559
MSD	0.4192	0.3789	0.3837	0.3813	0.3793	0.3750	0.3596	0.3274	0.2335	0.3598
MSDJ	0.4288	0.3859	0.3891	0.3847	0.3822	0.3741	0.3539	0.3150	0.2248	0.3598
SRC	0.3119	0.2985	0.2915	0.2621	0.2143	0.1676	0.0842	0.0078	-0.0425	0.1773
NHSM	0.4343	0.3928	0.3951	0.3905	0.3841	0.3711	0.3449	0.2981	0.2121	0.3581
BCF	0.3073	0.2636	0.2607	0.2638	0.2530	0.2553	0.2400	0.2224	0.1897	0.2506
SMTP	0.4185	0.3782	0.3833	0.3810	0.3789	0.3745	0.3589	0.3268	0.2328	0.3592
PC	0.1422	0.1300	0.1192	0.1004	0.0765	0.0609	0.0460	0.0309	0.0128	0.0799
PIP	0.3939	0.3913	0.3980	0.3940	0.3870	0.3768	0.3529	0.3089	0.2128	0.3573
CjacMD	0.2796	0.2561	0.2302	0.2007	0.1547	0.1157	0.0492	-0.0346	-0.1310	0.1245
TMJ	0.3859	0.3849	0.3881	0.3837	0.3812	0.3730	0.3527	0.3131	0.2063	0.3521
Feng	0.3872	0.3887	0.3925	0.3889	0.3862	0.3779	0.3578	0.3169	0.2053	0.3557
Mu	0.4062	0.3960	0.3941	0.3780	0.3525	0.3158	0.2667	0.1905	0.0932	0.3103
TA	0.4221	0.3815	0.3867	0.3839	0.3818	0.3771	0.3613	0.3279	0.2325	0.3616
TAJ	0.4311	0.3879	0.3910	0.3871	0.3843	0.3761	0.3554	0.3157	0.2243	0.3614
TAN	0.4338	0.3869	0.3901	0.3793	0.3712	0.3583	0.3366	0.2951	0.2148	0.3518
TANJ	0.4413	0.3964	0.4000	0.3898	0.3810	0.3650	0.3381	0.2913	0.2110	0.3571
SMD	0.4180	0.3779	0.3818	0.3792	0.3777	0.3739	0.3619	0.3459	0.3008	0.3686
HSMD	0.4257	0.3846	0.3892	0.3867	0.3835	0.3763	0.3563	0.3138	0.1617	0.3531
HSMDJ	0.3880	0.3876	0.3897	0.3855	0.3799	0.3674	0.3414	0.2971	0.1955	0.3480

Table 6
Precision metric within the recommendation process.

Measure	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	Average (Precision)
Cosine	0.0055	0.0104	0.0154	0.0207	0.0262	0.0324	0.0396	0.0508	0.0836	0.0316
COJ	0.0056	0.0105	0.0156	0.0207	0.0262	0.0318	0.0377	0.0439	0.0510	0.0270
CON	0.0054	0.0100	0.0148	0.0197	0.0250	0.0310	0.0384	0.0512	0.0912	0.0319
COD	0.0046	0.0086	0.0128	0.0172	0.0220	0.0273	0.0339	0.0443	0.0755	0.0274
CosineJ	0.0056	0.0105	0.0156	0.0209	0.0265	0.0327	0.0399	0.0510	0.0835	0.0318
Pearson	0.0051	0.0095	0.0141	0.0187	0.0237	0.0291	0.0359	0.0467	0.0803	0.0292
WPC	0.0052	0.0097	0.0144	0.0190	0.0241	0.0295	0.0363	0.0471	0.0804	0.0295
SPC	0.0051	0.0096	0.0141	0.0187	0.0238	0.0292	0.0359	0.0468	0.0804	0.0293
PearsonJ	0.0052	0.0097	0.0143	0.0190	0.0240	0.0295	0.0362	0.0471	0.0804	0.0295
Jaccard	0.0056	0.0105	0.0155	0.0207	0.0261	0.0317	0.0377	0.0438	0.0511	0.0270
MSD	0.0055	0.0104	0.0155	0.0207	0.0262	0.0324	0.0396	0.0508	0.0836	0.0316
MSDJ	0.0056	0.0105	0.0156	0.0209	0.0265	0.0327	0.0399	0.0510	0.0835	0.0318
SRC	0.0061	0.0116	0.0168	0.0213	0.0243	0.0255	0.0234	0.0199	0.0362	0.0206
NHSM	0.0057	0.0107	0.0158	0.0211	0.0268	0.0331	0.0402	0.0512	0.0829	0.0319
BCF	0.0047	0.0089	0.0133	0.0179	0.0230	0.0282	0.0341	0.0412	0.0518	0.0248
SMTP	0.0055	0.0104	0.0154	0.0207	0.0262	0.0324	0.0396	0.0508	0.0836	0.0316
PC	0.0050	0.0094	0.0140	0.0189	0.0244	0.0304	0.0379	0.0498	0.0854	0.0306
PIP	0.0057	0.0107	0.0158	0.0211	0.0268	0.0331	0.0402	0.0513	0.0833	0.0320
CjacMD	0.0054	0.0100	0.0147	0.0194	0.0243	0.0291	0.0339	0.0389	0.0453	0.0246
TMJ	0.0056	0.0105	0.0155	0.0207	0.0261	0.0317	0.0377	0.0438	0.0511	0.0270
Feng	0.0056	0.0105	0.0156	0.0207	0.0262	0.0318	0.0377	0.0439	0.0510	0.0270
Mu	0.0053	0.0099	0.0146	0.0194	0.0245	0.0300	0.0367	0.0476	0.0814	0.0299
TA	0.0055	0.0104	0.0155	0.0207	0.0263	0.0325	0.0397	0.0509	0.0836	0.0317
TAJ	0.0056	0.0106	0.0157	0.0209	0.0265	0.0328	0.0400	0.0511	0.0835	0.0319
TAN	0.0053	0.0099	0.0146	0.0194	0.0248	0.0307	0.0381	0.0510	0.0909	0.0316
TANJ	0.0054	0.0101	0.0148	0.0197	0.0251	0.0311	0.0385	0.0512	0.0909	0.0319
SMD	0.0055	0.0104	0.0154	0.0205	0.0259	0.0316	0.0377	0.0448	0.0546	0.0274
HSMD	0.0056	0.0105	0.0155	0.0207	0.0261	0.0318	0.0376	0.0437	0.0499	0.0268
HSMDJ	0.0057	0.0106	0.0157	0.0208	0.0263	0.0320	0.0379	0.0439	0.0507	0.0271

process is commonly evaluated by MAE metric, and the recommendation process is widely evaluated by both precision and recall metrics.

On the other extreme, F1 metric is the technique to assemble the precision and recall together. Eq. (48) specifies F1 metric. The

Table 7
Recall metric within the recommendation process.

Measure	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	Average (Recall)
Cosine	0.9241	0.9208	0.9211	0.9177	0.9150	0.9147	0.9066	0.8937	0.8021	0.9018
COJ	0.9265	0.9235	0.9230	0.9199	0.9163	0.9157	0.9067	0.8942	0.8462	0.9080
CON	0.9313	0.9269	0.9269	0.9233	0.9201	0.9184	0.9079	0.8814	0.7327	0.8965
COD	0.9452	0.9416	0.9394	0.9340	0.9293	0.9269	0.9133	0.8872	0.7690	0.9095
CosineJ	0.9266	0.9232	0.9223	0.9193	0.9159	0.9153	0.9066	0.8914	0.7970	0.9020
Pearson	0.9439	0.9402	0.9388	0.9359	0.9331	0.9309	0.9190	0.8948	0.7834	0.9133
WPC	0.9430	0.9387	0.9373	0.9349	0.9331	0.9315	0.9197	0.8967	0.7836	0.9132
SPC	0.9443	0.9402	0.9390	0.9362	0.9335	0.9316	0.9196	0.8960	0.7840	0.9138
PearsonJ	0.9429	0.9440	0.9373	0.9351	0.9323	0.9309	0.9186	0.8948	0.7814	0.9130
Jaccard	0.9266	0.9230	0.9221	0.9191	0.9158	0.9155	0.9073	0.8947	0.8496	0.9082
MSD	0.9242	0.9209	0.9210	0.9178	0.9151	0.9147	0.9067	0.8938	0.8020	0.9018
MSDJ	0.9267	0.9231	0.9223	0.9194	0.9159	0.9154	0.9067	0.8914	0.7969	0.9020
SRC	0.7401	0.6827	0.6167	0.5303	0.4208	0.3010	0.1803	0.0825	0.0489	0.4004
NHSM	0.9283	0.9238	0.9234	0.9191	0.9159	0.9142	0.9037	0.8845	0.7842	0.8997
BCF	0.9124	0.9106	0.9098	0.9027	0.9032	0.9007	0.8927	0.8790	0.8495	0.8956
SMTP	0.9242	0.9209	0.9209	0.9177	0.9151	0.9148	0.9065	0.8936	0.8016	0.9017
PC	0.9132	0.9115	0.9117	0.9086	0.9080	0.9096	0.8987	0.8753	0.7456	0.8869
PIP	0.9284	0.9259	0.9257	0.9211	0.9183	0.9171	0.9068	0.8887	0.7882	0.9022
CjacMD	0.9187	0.9130	0.9102	0.9029	0.8971	0.8940	0.8779	0.8551	0.8009	0.8855
TMJ	0.9264	0.9228	0.9221	0.9191	0.9158	0.9155	0.9073	0.8948	0.8498	0.9082
Feng	0.9266	0.9237	0.9232	0.9199	0.9164	0.9156	0.9070	0.8944	0.8469	0.9082
Mu	0.9354	0.9320	0.9326	0.9298	0.9282	0.9278	0.9184	0.8981	0.7899	0.9102
ATC	0.9242	0.9211	0.9211	0.9177	0.9149	0.9145	0.9060	0.8928	0.8005	0.9014
ATCJ	0.9265	0.9229	0.9224	0.9191	0.9154	0.9149	0.9060	0.8907	0.7957	0.9015
ATCN	0.9303	0.9272	0.9262	0.9216	0.9178	0.9156	0.9025	0.8747	0.7277	0.8937
ATCNJ	0.9300	0.9267	0.9256	0.9210	0.9176	0.9154	0.9026	0.8744	0.7254	0.8932
SMD	0.9244	0.9212	0.9210	0.9180	0.9148	0.9152	0.9078	0.9000	0.8718	0.9105
HSMD	0.9262	0.9228	0.9224	0.9190	0.9161	0.9158	0.9070	0.8948	0.8479	0.9080
HSMDJ	0.9284	0.9244	0.9238	0.9199	0.9162	0.9152	0.9049	0.8888	0.8391	0.9067

larger F1 is, the better measure is.

$$\mathrm{F1} = \frac{\mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{48}$$

Shortly, MAE is used to evaluate the estimation process and F1 is used to evaluate the recommendation process. Table 8 shows the average MAE and F1 values of all measures. The shaded cells, in gray color, indicates the best values.

The Top-5 measures, according to F1 metric within the recommendation process (Table 8, Fig. 6), are: PIP, NHSM, TANJ, TAJ, and CON whose average F1 are 0.030904, 0.030849, 0.030769, 0.030768, and 0.030763. Shortly, the dominant orders of our measures, TA, TAJ, TAN, TANJ, SMD, HSMD, and HSMDJ, are 8th, 4th, 10th, 3rd, 19th, 29th, and 24th among all measures, respectively.

From the drawn-above results, it sounds easy to recognize that Table 8, and both Figs. 1 and 6 represent the general evaluation of all measures regarding both the estimation and recommendation processes. However, we cannot unify MAE and F1 as the precision and recall are unified because of that the estimation and recommendation processes are not always proportional. Therefore, let A be a set of top-5 measures regarding MAE and let B be set of top-5 measures regarding F1. Then, the intersection of A and B contains the best measures. From Tables 3 and 8, we have $A = \{\text{TAJ, MSDJ, CosineJ, SMD, NHSM}\}$, and $B = \{\text{PIP, NHSM, TANJ, TAJ, CON}\}$. Obviously, the best measures in general comparison of the initial evaluation are TAJ and NHSM.

Finally, it is worth indicating that we have made further evaluation using the best nine similarity measures (concluded from Sections 6.1 and 6.2 on three more datasets with several K values (5, 20, 50 and 100) for neighborhood using KNN under two r values (0.7 and 0.9) in which the sparsity of each datasets is stressed.[1] Experiments have been conducted on: (1) the Film Trust dataset, as one of the most sparse datasets, with K [5, 20, 50 and 100], (2) the recent version of Movielens-100k with K [5, 20 and 50], and (3) Movielens-1M with k [5, 20]. Results attested that our measures are still shown in the top particularly on MAE, MSE and R metrics. In conclusion of the obtained results, the final rank for the best sixth measures were: TA, SMD, MSDJ, PIP, NHSM and Cosine. Final Rank gives a priority for those measures whose values are the highest with both values of r in general, and $r = 0.9$ in particular. That is because of that this value of r ($r = 0.9$) reflects the highest level of sparsity of each datasets.

7. Discussion

While experimenting the similarity measures in Section 6., we found that some measures have been capable of being at the uppermost list of Top-N measure with more than one metric. For example, SMD, TAJ and Cosine are proven effective staying in the lists of top-5 measures with regard to MAE, MSE and R (see Tables 3–6). This implies the same semantics of MAE, MSE, and R within the estimation process. It is possible to conclude that the important matter is to split the evaluation process into two sub-processes such as the estimation and recommendation. For each sub-process, we only need to choose one representative metric. In this research, we choose MAE and F1 as representative metrics for the estimation and recommendation processes, respectively. Although it is totally feasible to evaluate the similarity measures with MAE and F1, we showed that it is better to go further with other metrics like MSE and R.

[1] https://github.com/aliamer/Enhancing-Recommendation-Systems-Performance-Using-Highly-Effective-Similarity-Measures/blob/main/Further%20evaluation.pdf .

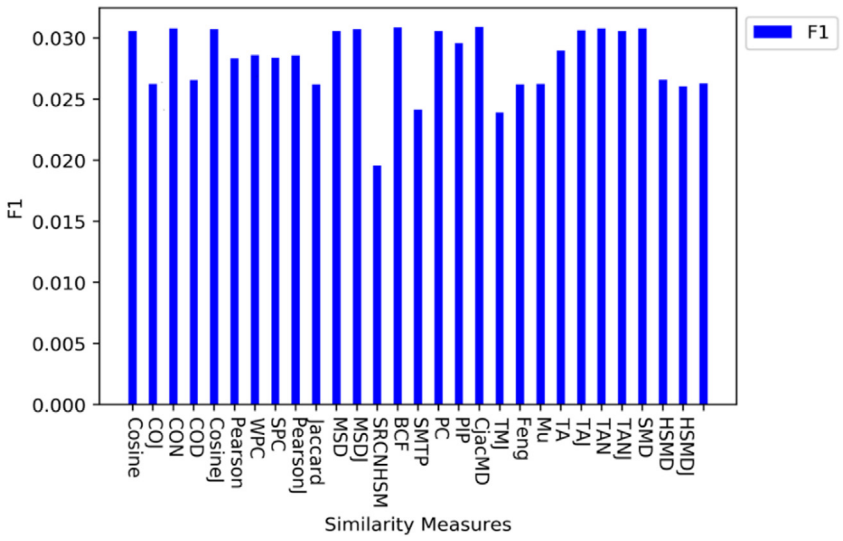

Fig. 6. Measure comparison with F1 metric.

Table 8
General MAE and F1 over all measures.

Measure	MAE	F1
Cosine	0.7713	0.030551
COJ	0.7720	0.026220
CON	0.7842	0.030763
COD	0.8555	0.026557
CosineJ	0.7704	0.030717
Pearson	0.7845	0.028327
WPC	0.7817	0.028598
SPC	0.7826	0.028379
PearsonJ	0.7840	0.028566
Jaccard	0.7723	0.026189
MSD	0.7734	0.030561
MSDJ	0.7703	0.030717
SRC	0.8169	0.019562
NHSM	0.7712	0.030849
BCF	0.8147	0.024121
SMTP	0.7737	0.030551
PC	0.8558	0.029559
PIP	0.7717	0.030904
CjacMD	0.8391	0.023893
TMJ	0.7721	0.026189
Feng	0.7714	0.026220
Mu	0.7814	0.028980
TA	0.7729	0.030602
TAJ	0.7699	0.030768
TAN	0.7789	0.030552
TANJ	0.7722	0.030769
SMD	0.7709	0.026579
HSMD	0.7748	0.026053
HSMDJ	0.7733	0.026282

Although SMD, TAJ and NHSM are the best measures, in general, with the representative metrics MAE and F1, TAJ and SMD are also shown pre-eminent measures concerning all other metrics. TAJ is a dominant measure over metrics MAE, precision, F1, MSE, and R whereas SMD is dominant measure over metrics

Table 9
Comparison of NHSM, SMD, and TAJ - Movielens — 100 K.

	MAE	MSE	I-R	I-Precision	I-Recall
NHSM	0.7712	0.9695	0.6419	**0.9681**	0.1003
TAJ	**0.7699**	0.9637	0.6386	**0.9681**	0.0985
SMD	0.7709	**0.9618**	**0.6314**	0.9726	**0.0895**

Table 10
Comparison of TA, SMD, Cosine, and TAJ with MAE, MSE, I-R, I-Precision, and I-Recall.

	MAE	MSE	I-R	I-Precision	I-Recall
Cosine	0.6608	0.7639	**0.8513**	0.8939	0.2045
TA	**0.6605**	**0.7604**	0.8482	0.8939	0.2046
TAJ	0.6610	0.7614	0.8483	0.8939	0.2051
SMD	0.6627	0.7736	0.8147	**0.9730**	**0.1605**

MAE, recall, MSE, and R. interestingly, NHSM has not been a pre-eminent measure with metrics MSE and R. As usual, we define the pre-eminent measures (dominant measures) as the ones in top-5 lists. It is useful to compare NHSM, SMD, and TAJ, but it is impossible to unify metrics MAE, MSE, and R together. However, to make the general comparison, some necessary transformations have been done. Let I-R be the inverse of R metric. Let I-Precision be the inverse of precision metric and let I-Recall be the inverse of recall metric. The smaller I-R, I-Precision, and I-Recall are, the better the measures are. Eq. (49) specifies I-R, I-Precision, and I-Recall. Hence, I-R, I-Precision, and I-Recall are replacers of R, Precision, and Recall, respectively.

$$\begin{aligned} \text{I} - \text{R} &= 1 - \text{R} \\ \text{I} - \text{Precision} &= 1 - \text{Precision} \\ \text{I} - \text{Recall} &= 1 - \text{Recall} \end{aligned} \tag{49}$$

Table 9 lists the metrics MAE, MSE, I-R, I-Precision, and I-Recall of the pre-eminent measures: NHSM, SMD, and TAJ.

From Table 9 (and Fig. 7), TAJ is the best measure with MAE, SMD is the best measure with MSE, I-R, and I-Recall. Both NHSM and TAJ are the best measures with I-Precision. Considering the statistics of Table 9, SMD has been the top performer followed by TAJ and NHSM respectively.

On the other hand, regarding Film trust datasets, Table 10 (and Fig. 8) lists metrics MAE, MSE, I-R, I-Precision, and I-Recall of preeminent measures TA, SMD, Cosine, and TAJ.

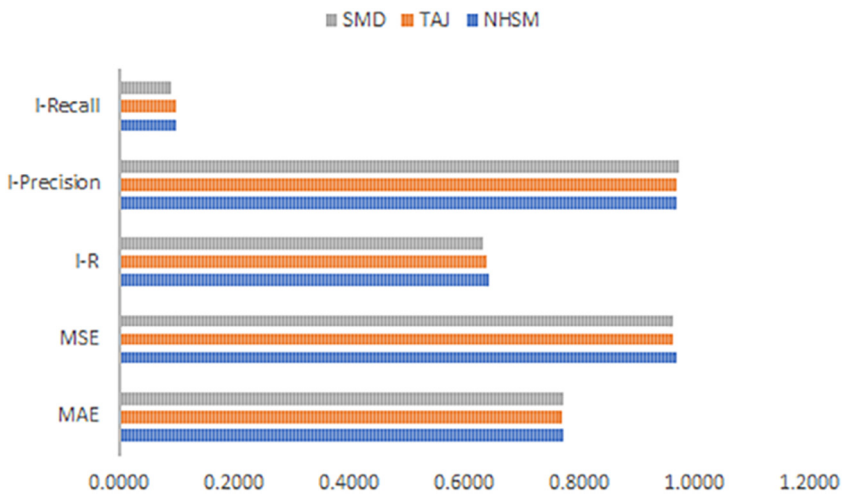

Fig. 7. Comparison of NHSM, SMD, and TAJ with MAE, MSE, I-R, I-Precision, and I-Recall.

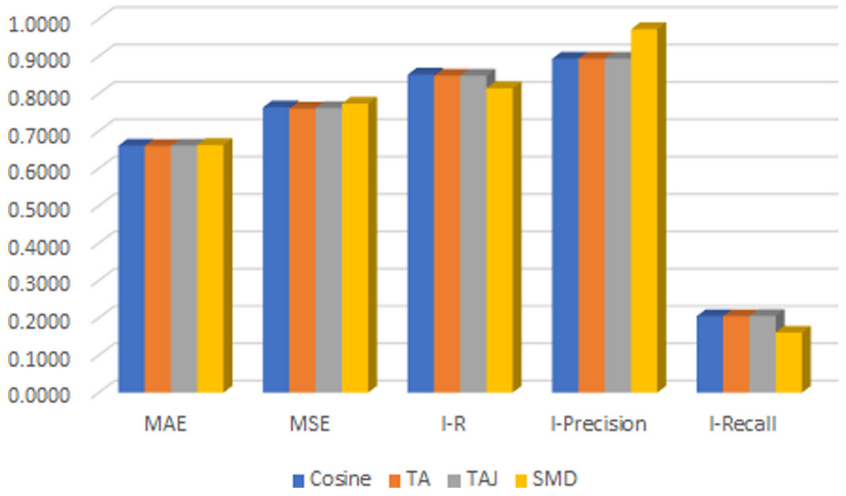

Fig. 8. Comparison of TA, SMD, Cosine, and TAJ with MAE, MSE, I-R, I-Precision, and I-Recall.

Finally, the speed is also a metric to evaluate the pre-eminent measures but the bias to calculate the speed metric in the evaluation process is high due to the casual factors of hardware and software. Therefore, the speed has not been taken as an important metric in this research. From experiments, however, the speed values, on Movielens – 100k, of NHSM, SMD, and TAJ are 0.3288, 0.3116, and 0.3411 in a millisecond, respectively. Hence, SMD has indisputably been the fastest measure.

7.1. Merits and limitations

To the best of our knowledge, the proposed work of this paper is the first work in CF literature that seeks to experimentally evaluate almost 30 similarity measures. Along with the evaluation of such big number of measures, the work have introduced new similarity measures that enjoy a very simplistic design with less complexity compared to an almost 95% of the similarity measures evaluated in this study. In particular, compared to the top performer measures like NHMS and PIP, our proposed measures are superior in terms of their simple design as well as high performance as shown in the drawn-above results. Moreover, we divided evaluation process into two sub-process and set the appropriate evaluation metrics accordingly. In the meanwhile, we proposed r parameter to confirm the behavior of all measures under different circumstances of data sparsity, and r value ranges from 0.1 to 0.9. To confirm randomness in the KNN, on the other hand, we set K to [5, 20, 50 and 100] to make the prediction and recommendation. With such settings for r and k parameters, the proposed work has been seen capable of yielding highly-accurate results, particularly for top performer measures. Consequently, more precise evaluation for similarity measures under different conditions of data sparsity has been allowed. As mentioned earlier, we used other parameters such as k and r, with r is more important than k. All previous researches used only one r value when our work used 9 cases of r to deeply diversify data sparsity, and thoroughly examine all measures fairly and rigorously under several conditions of data sparsity which is the ultimate goal of this work.

As a matter of fact, with each successive r, the experience of the proposed model using cross validation gets increased over time, resulting in higher performance. Therefore, after experiencing all r values and processing dataset in each r, the accuracy of our model is increased robustly. Consequently, the prediction made by our model has been very accurate. It is worth indicating that the uniqueness of our work compared to all earlier studies, relies in the fact that our work used parameter "r" to decide nine cases of training and testing data while taking several values for K of KNN {5, 20, 50 and 100}. While all earlier studies used one r value which is mostly 7:3 or 6:4 when 7 or 6 is 70% or

60% training and 3 or 4 is 30% or 40% testing, while using the traditional evaluation as several K values are used for the KNN. This intelligible difference makes our work unique and novel (not to mention the proposed methods for evaluation, the number of similarity measures studied along the paper, and the proposed dynamic formulas to compute estimation and recommendation).

However, as a cost of proving that the trade-off is unescapable, the drawn-above evaluation has suffered a limitation. This limitation represented in that the work would need a lengthy time to apply the same evaluation process on huge datasets like Movielens 100M. Because of the rigorous procedure followed in our work for the evaluation process to get the most accurate results, the evaluator machine was not able to break $K = 100$ for Movielens-100k or even when $= 50$ for Movielens-1M. Accordingly, to follow the same evaluation procedure on huge datasets, powerful resources have to be used which would be one of our future work to generate the global CF framework. Nevertheless, from the research point of view, such limitation would not devalue the proposed work as several works including some recently-published [11,43] were evaluated on only one single dataset (Movielens 100K). In the follow-up work, to solve this problem and to make our work of high caliber, it is planned to use the parallel collaborative filtering algorithm [17] to perform such kind of evaluation on bigger datasets.

8. Conclusions and future work

In this work, the intrinsic pursuit has been directed toward finding influential solutions for the data sparsity problem by effectively making full use of all rated and non-rated items. To meet this goal, three main "new" measures, namely, SMD, HSMD, and TA have been proposed. According to the findings of this study, the proposed similarity measures contributed overwhelmingly to maximize the recommendation accuracy. Besides splitting the evaluation processes into the estimation process and recommendation process as well as proposing the new measures; this research has been presented with an intention of introducing a practical guidance of CF similarity measures performance.

Using the cross-validation based KNN, we evaluated and compared almost 30 similarity measures succinctly. From the experimental study, it was recorded that our proposed measures were shown efficient and effective, particularly SMD and TA. In its turn, SMD has been proven to be a preeminent measure in top-5 lists with metrics MAE, recall, MSE, and R. Moreover, in this study, we found that both SMD and Jaccard have a common feature in that both measures are concerned about the existence of ratings while disregarding the magnitude of ratings. Nevertheless, Jaccard has not been a dominant measure, yet it has been an important factor to improve any measure [32]. In fact, good measures such as TAJ, TANJ, NHSM and MSDJ have been combined with Jaccard, and collectively have provided effective results. Consequently, from all experiments, and considering the fact that SMD is obviously extremely better than Jaccard, it is highly potential that SMD combination with any other measures would produce highly effective results than that of Jaccard's combinations.

We have made the experimental study in two key evaluation phases. The initial evaluation phase was comprehensively done with all 30 measures so that the top performers could be carefully picked for further evaluation. The further evaluation phase has been done with the best nine measures on three more datasets to accentuate the superiority of top performers including the proposed measures of this work. The results also showcased the superiority of proposed measures. In addition, by evaluating the similarity-based CF algorithm, we found that the issue of choosing how many tested metrics is not as important as choosing the right representative metrics for both the estimation process and recommendation processes noting that both processes are different. For instance, it is not an accurate if we calculate MAE metric for the recommendation process, or, F1 metric for the estimation process. Of course, it would be better to calculate many right representative metrics for each process but it is also easier to draw the best measures with a small set of right representative metrics. In this work, based on results of initial evaluation phase, to draw a general comparison on the agreed-upon metrics of similarity measures, we made some transformation to make the comparison generally and practically acceptable. The comparison proved that SMD followed by TAJ and NHMS were the top performer similarity measures.

Finally, there are three important conclusions that can be derived from separating the evaluation into two processes (estimation and recommendation):

1. The evaluation tests are shown to be more accurate than traditional evaluation followed in almost all earlier CF work. As a result, these tests can be considered as a short experimental summary of the similarity measures for CF.
2. It is good if we calculate many right representative metrics for each process, yet it is easier to draw best measures with small set of right representative metrics, as shown in the discussion section.
3. Jaccard measure did not prove to be a dominant measure, but it proved to be an important factor to improve any numeric measures like TAJ and NHMS.
4. Given the fact that SMD has been shown a dominant and top performer measure as well as superior to Jaccard. SMD can replace Jaccard effectively in all earlier work that used Jaccard combination.

In the future work, therefore, besides leveraging the parallel processing to run this work on huge datasets, we propose combining SMD with other measures chiefly those combined with Jaccard, and record the impact of SMD combination. The aim is to establish a global framework of a recommender tool while taking the cognitive approaches [47] as well as some missing CF similarity measures like [34,48,49] into consideration.

CRediT authorship contribution statement

Ali A. Amer: Conception and design, Implementing the approach and analyzing results of all experiments, Preparation, writing and revising the manuscript. **Hassan I. Abdalla:** Conception and design, Implementing the approach and analyzing results of all experiments, Preparation, writing and revising the manuscript. **Loc Nguyen:** Conception and design, Implementing the approach and analyzing results of all experiments, Preparation, writing and revising the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Datasets, detailed results and code availability

The datasets are being uploaded on Google Drive https://drive.google.com/drive/folders/1lz3-eVjAf-IZ5aulJSK4dX81Wt2_OFz3?fbclid=IwAR0fgDjrIUORMdhMg5TKVxd-tMHoFKooDOYH9g1rEXRFV7yJqV1L3_Q674U The detailed results and code are being uploaded on GitHub https://github.com/aliamer/Enhancing-Recommendation-Systems-Performance-Using-Highly-Effective-Similarity-Measures

Acknowledgments

The authors would like to thank and appreciate the support received from the Research Office of Zayed University for providing the necessary facilities to accomplish this work. The authors would also like to sincerely express their thanks to Journal of Knowledge-Based Systems including Editors and the unknown Reviewers for providing their valuable suggestions without which this work would not be enhanced.

Funding

This research has been supported by Research Incentive Fund (RIF) Grant Activity Code: R19093– Zayed University, UAE.

References

[1] J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Recommender system application developments: A survey, Decis. Support Syst. 74 (2015) 12–32, http://dx.doi.org/10.1016/j.dss.2015.03.008.

[2] A. Kouadria, O. Nouali, M.Y.H. Al-Shamri, A multi-criteria collaborative filtering recommender system using learning-to-rank and rank aggregation, Arab. J. Sci. Eng. 45 (4) (2020) 2835–2845, http://dx.doi.org/10.1007/s13369-019-04180-3.

[3] M. Ayub, M.A. Ghazanfar, T. Khan, A. Saleem, An effective model for jaccard coefficient to increase the performance of collaborative filtering, Arab. J. Sci. Eng. 45 (12) (2020) 9997–10017, http://dx.doi.org/10.1007/s13369-020-04568-6.

[4] Y. Shi, M. Larson, A. Hanjalic, Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges, ACM Comput. Surv. 47 (2014) 3:1–3:45, http://dx.doi.org/10.1145/2556270.

[5] D. Wang, Y. Liang, D. Xu, X. Feng, R. Guan, A content-based recommender system for computer science publications, Knowl.-Based Syst. 157 (2018) 1–9, http://dx.doi.org/10.1016/j.knosys.2018.05.001.

[6] S. Jiang, X. Qian, J. Shen, Y. Fu, T. Mei, Author topic model-based collaborative filtering for personalized POI recommendations, IEEE Trans. Multimed. 17 (6) (2015) 907–918, http://dx.doi.org/10.1109/TMM.2015.2417506.

[7] M. Ayub, M.A. Ghazanfar, M. Maqsood, A. Saleem, A Jaccard base similarity measure to improve performance of CF based recommender systems, in: International Conference on Information Networking, IEEE Computer Society, 2018, pp. 1–6, http://dx.doi.org/10.1109/ICOIN.2018.8343073.

[8] M.A. Ghazanfar, A. Prugel-Bennett, A scalable, accurate hybrid recommender system, in: 3rd International Conference on Knowledge Discovery and Data Mining, WKDD 2010, 2010, pp. 94–98, http://dx.doi.org/10.1109/WKDD.2010.117.

[9] L. Xiong, X. Chen, T.K. Huang, J. Schneider, J.G. Carbonell, Temporal collaborative filtering with Bayesian probabilistic tensor factorization, in: Proceedings of the 10th SIAM International Conference on Data Mining, SDM 2010, Society for Industrial and Applied Mathematics Publications, 2010, pp. 211–222, http://dx.doi.org/10.1137/1.9781611972801.19.

[10] M. Ayub, M.A. Ghazanfar, Z. Mehmood, T. Saba, R. Alharbey, A.M. Munshi, M.A. Alrige, Modeling user rating preference behavior to improve the performance of the collaborative filtering based recommender systems, PLoS One 14 (8) (2019) http://dx.doi.org/10.1371/journal.pone.0220129.

[11] S. Bag, S.K. Kumar, M.K. Tiwari, An efficient recommendation generation using relevant Jaccard similarity, Inform. Sci. 483 (2019) 53–64, http://dx.doi.org/10.1016/j.ins.2019.01.023.

[12] L. Ren, W. Wang, An SVM-based collaborative filtering approach for Top-N web services recommendation, Future Gener. Comput. Syst. 78 (2018) 531–543, http://dx.doi.org/10.1016/j.future.2017.07.027.

[13] Y. Wang, J. Deng, J. Gao, P. Zhang, A hybrid user similarity model for collaborative filtering, Inform. Sci. 418–419 (2017) 102–118, http://dx.doi.org/10.1016/j.ins.2017.08.008.

[14] S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, et al., Video Suggestion and Discovery for Youtube, 895, Association for Computing Machinery (ACM), 2008, http://dx.doi.org/10.1145/1367497.1367618.

[15] https://www.amazon.com/. (Accessed 28 April 2019).

[16] https://news.google.com/. (Accessed 28 April 2019).

[17] Z. Wang, Y. Liu, S. Chiu, An efficient parallel collaborative filtering algorithm on multi-GPU platform, J. Supercomput. 72 (6) (2016) 2080–2094, http://dx.doi.org/10.1007/s11227-014-1333-4.

[18] M.-P.T. Do, D.V. Nguyen, L. Nguyen, Model-based approach for collaborative filtering, in: Proceedings of the 6th International Conference on Information Technology for Education (IT@EDU2010), Ho Chi Minh University of Information Technology, 2010, pp. 217–225, Retrieved from https://goo.gl/BHu7ge.

[19] J. Bobadilla, F. Serradilla, J. Bernal, A new collaborative filtering metric that improves the behavior of recommender systems, Knowl.-Based Syst. 23 (6) (2010) 520–528, http://dx.doi.org/10.1016/j.knosys.2010.03.009.

[20] J. Bobadilla, F. Ortega, A. Hernando, J. Bernal, A collaborative filtering approach to mitigate the new user cold start problem, Knowl.-Based Syst. 26 (2012) 225–238, http://dx.doi.org/10.1016/j.knosys.2011.07.021.

[21] J. Bobadilla, A. Hernando, F. Ortega, A. Gutiérrez, Collaborative filtering based on significances, Inform. Sci. 185 (1) (2012) 1–17, http://dx.doi.org/10.1016/j.ins.2011.09.014.

[22] K. Choi, Y. Suh, A new similarity function for selecting neighbors for each target item in collaborative filtering, Knowl.-Based Syst. 37 (2013) 146–153, http://dx.doi.org/10.1016/j.knosys.2012.07.019.

[23] M. Schwarz, M. Lobur, Y. Stekh, Analysis of the effectiveness of similarity measures for recommender systems, in: 2017 14th International Conference the Experience of Designing and Application of CAD Systems in Microelectronics, CADSM 2017 – Proceedings, Institute of Electrical and Electronics Engineers Inc., 2017, pp. 275–277, http://dx.doi.org/10.1109/CADSM.2017.7916133.

[24] Y. El Madani El Alami, N. El Habib, O. El Beqqali, Improving neighborhood-based collaborative filtering by a heuristic approach and an adjusted similarity measure, in: CEUR Workshop Proceedings, 1580, 2015, pp. 16–22.

[25] Suryakant, T. Mahara, A new similarity measure based on mean measure of divergence for collaborative filtering in sparse environment, Procedia Comput. Sci. 89 (2016) 450–456, http://dx.doi.org/10.1016/j.procs.2016.06.099.

[26] B.K. Patra, R. Launonen, V. Ollikainen, S. Nandi, A new similarity measure using bhattacharyya coefficient for collaborative filtering in sparse data, Knowl.-Based Syst. 82 (2015) 163–177, http://dx.doi.org/10.1016/j.knosys.2015.03.001.

[27] H. Cao, J. Deng, H. Guo, B. He, Y. Wang, An improved recommendation algorithm based on bhattacharyya coefficient, in: 2016 IEEE International Conference on Knowledge Engineering and Applications, Institute of Electrical and Electronics Engineers Inc, 2016, pp. 241–244, http://dx.doi.org/10.1109/ICKEA.2016.7803027.

[28] H. Koohi, K. Kiani, A new method to find neighbor users that improves the performance of collaborative filtering, Expert Syst. Appl. 83 (2017) 30–39, http://dx.doi.org/10.1016/j.eswa.2017.04.027.

[29] K.G. Saranya, G. Sudha Sadasivam, Modified heuristic similarity measure for personalization using collaborative filtering technique, Appl. Math. Inf. Sci. 11 (1) (2017) 307–315, http://dx.doi.org/10.18576/amis/110137.

[30] H.J. Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, Inform. Sci. 178 (1) (2008) 37–51, http://dx.doi.org/10.1016/j.ins.2007.07.024.

[31] H. Liu, Z. Hu, A. Mian, H. Tian, X. Zhu, A new user similarity model to improve the accuracy of collaborative filtering, Knowl.-Based Syst. 56 (2014) 156–166, http://dx.doi.org/10.1016/j.knosys.2013.11.006.

[32] S.B. Sun, Z.H. Zhang, X.L. Dong, H.R. Zhang, T.J. Li, L. Zhang, F. Min, Integrating triangle and jaccard similarities for recommendation, PLoS One 12 (8) (2017) http://dx.doi.org/10.1371/journal.pone.0183570.

[33] Q. Jin, Y. Zhang, W. Cai, Y. Zhang, A new similarity computing model of collaborative filtering, IEEE Access 8 (2020) 17594–17604, http://dx.doi.org/10.1109/ACCESS.2020.2965595.

[34] L.J. Chen, Z.K. Zhang, J.H. Liu, J. Gao, T. Zhou, A vertex similarity index for better personalized recommendation, Physica A 466 (2017) 607–615, http://dx.doi.org/10.1016/j.physa.2016.09.057.

[35] R.D.T. Júnior, Combining Collaborative and Content-Based Filtering to Recommend Research Papers. Distribution, 2004, pp. 1–71.

[36] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the 10th International Conference on World Wide Web, WWW 2001, Association for Computing Machinery, Inc., 2001, pp. 285–295, http://dx.doi.org/10.1145/371920.372071.

[37] J. Feng, X. Fengs, N. Zhang, J. Peng, An improved collaborative filtering method based on similarity, PLoS One 13 (9) (2018) http://dx.doi.org/10.1371/journal.pone.0204003.

[38] Y. Mu, N. Xiao, R. Tang, L. Luo, X. Yin, An efficient similarity measure for collaborative filtering, Procedia Comput. Sci. 147 (2019) 416–421, http://dx.doi.org/10.1016/j.procs.2019.01.258.

[39] Y.S. Lin, J.Y. Jiang, S.J. Lee, A similarity measure for text classification and clustering, IEEE Trans. Knowl. Data Eng. 26 (7) (2014) 1575–1590, http://dx.doi.org/10.1109/TKDE.2013.19.

[40] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowl.-Based Syst. 46 (2013) 109–132, http://dx.doi.org/10.1016/j.knosys.2013.03.012.

[41] E.F. Harris, T. Sjøvold, Calculation of Smith's mean measure of divergence for intergroup comparisons using nonmetric data, Dent. Anthropol. J. 17 (3) (2018) 83–93, http://dx.doi.org/10.26575/daj.v17i3.152.

[42] N. Loc, A.A. Amer, Advanced cosine measures for collaborative filtering, Adapt. Pers. (ADP) 1 (2019) 21–41.

[43] GroupLens, MovieLens Datasets, GroupLens Research Project, University of Minnesota, USA, 1998, Retrieved August 3, 2018, from GroupLens Research website: http://grouplens.org/datasets/movielens.
[44] F.M. Harper, J.A. Konstan, The movielens datasets: History and context, ACM Trans. Interacti. Intell. Syst. 5 (4) (2015) 19:1–19:19, http://dx.doi.org/10.1145/2827872.
[45] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems, ACM Trans. Inf. Syst. 22 (2004) 5–53, http://dx.doi.org/10.1145/963770.963772.
[46] D.C. Montgomery, G.C. Runger, Applied statistics and probability for engineers, Eur. J. Eng. Educ. 19 (3) (1994) 383, http://dx.doi.org/10.1080/03043799408928333.
[47] C. Angulo, I.Z. Falomir, D. Anguita, N. Agell, E. Cambria, Bridging cognitive models and recommender systems, Cogn. Comput. 12 (2020) 426–427, http://dx.doi.org/10.1007/s12559-020-09719-3.
[48] T. Zhou, Z. Kuscsik, J.G. Liu, M. Medo, J.R. Wakeling, Y.C. Zhang, Solving the apparent diversity-accuracy dilemma of recommender systems, Proc. Natl. Acad. Sci. USA 107 (10) (2010) 4511–4515, http://dx.doi.org/10.1073/pnas.1000488107.
[49] L. Lü, C.H. Jin, T. Zhou, Similarity index based on local paths for link prediction of complex networks, Phys. Rev. E 80 (4) (2009) http://dx.doi.org/10.1103/PhysRevE.80.046122.