International Journal of Data Science and Analytics
https://doi.org/10.1007/s41060-025-00846-4

REGULAR PAPER

An evaluation review of user similarity metrics in sparse collaborative filtering datasets

Kiriakos Sgardelis[1] · **Dionisis Margaris**[1] · **Dimitris Spiliotopoulos**[2] · **Costas Vassilakis**[3]

Received: 5 October 2024 / Accepted: 2 June 2025

Abstract

Collaborative filtering (CF) is one of the most prominent recommender system (RecSys) techniques of the recent years. CF generates rating predictions for the items that the user has not evaluated yet, using the evaluations of users with similar likings to the same items. Therefore, in CF the task of finding these users (which can be considered as reliable recommenders) is of high importance, while this task is especially challenging on sparse datasets. To this end, many user similarity metrics have been introduced and used in the literature, such as the Vector (or Cosine) Similarity metric, the Spearman rank correlation, the Pearson Correlation Coefficient (PCC), and others. For a CF RecSys, the use of the most efficient similarity metric is of great importance. This paper assesses the effectiveness of 15 user similarity metrics in sparse CF datasets, by conducting an extensive set of experiments. These experiments include 10 sparse CF datasets with diverse item domains, two neighbour selection approaches, two rating prediction formulas, and three rating prediction accuracy metrics. The evaluation results show that the metrics that achieve the best prediction results are found to be the Spearman rank correlation, followed by the Adjusted Rand Index, the Constrained PCC, and the Chebysev distance. Interestingly, the most widely used similarity metrics in CF research, i.e. the PCC and the Cosine Similarity, are not among the best performing metrics.

Keywords Recommender systems · Collaborative filtering · User similarity metrics · Sparse datasets · Evaluation review

1 Introduction

Over the last years, the World Wide Web has become an information chaos, which makes it challenging for users to find useful and interesting information, from products and services to news and weather forecasts. In this direction, research on RecSys can be very encompassing [1]. The target of a RecSys is to produce useful and personalised recommendations to its users. RecSys are applied to a very broad spectrum of domains, ranging from web services [2] and social networks [3, 4] to cultural heritage [5, 6] and open-source software [7].

One of the most widespread RecSys methods of the recent years is CF. The main idea behind CF is that, as in real life, people tend to trust individuals with similar views and likeness to them when asking for recommendations of products and services. Therefore, CF users that are calculated to be close to the active user can be used as (reliable) recommenders [8, 9].

Typically, a CF RecSys encompasses three steps. During the first step, similarities between all users are calculated to find the set of users that will operate as recommenders for

Kiriakos Sgardelis, Dionisis Margaris and Dimitris Spiliotopoulos contributed equally to this work.

✉ Costas Vassilakis
costas@uop.gr

Kiriakos Sgardelis
k.sgardelis@go.uop.gr

Dionisis Margaris
margaris@uop.gr

Dimitris Spiliotopoulos
dspiliot@uop.gr

[1] Department of Digital Systems, University of the Peloponnese, Valioti's building, 23100 Kladas, Peloponnese, Greece

[2] Department of Management Science and Technology, University of the Peloponnese, Thesi Sechi, 22131 Tripolis, Peloponnese, Greece

[3] Department of Informatics and Telecommunications, University of the Peloponnese, Akadimaikou G.K. Vlachou, 22131 Tripolis, Peloponnese, Greece

Published online: 23 June 2025

each user. This user set is referred to as Near Neighbours (NNs). Afterwards, in the second step, predictions are generated for the items that each user has not already evaluated. Lastly, in the third step, the items achieving the highest prediction values for each user are presented to them [10].

Over the last years, numerous research works have targeted all three aforementioned steps. The evaluation of a CF RecSys algorithm is accomplished either by engaging real users (i.e. humans who receive and evaluate recommendations), or using CF datasets that are widely used and accepted by the RecSys research community, such as the Amazon datasets [11], the MovieLens datasets [12], the CiaoDVD dataset [13], the Epinions dataset [14], and others.

The majority of the research works fall into the latter category, where evaluation is conducted by splitting the original dataset into test and train parts. Then the training part is used to compute similarities and identify NNs, while subsequently the CF algorithm uses similarities and NNs computed during the training stage in order to predict the ratings for the remaining test part [15]. The lower the difference between these predictions and the actual user ratings is, the more successful the algorithm is regarded as. In this context, numerous user similarity metrics have been introduced and used, such as the Vector (or Cosine) Similarity metric (COS), the Spearman rank correlation, the Kendall's Tau, the Jaccard index, the PCC, and so on. As a result, different research works employ different user similarity metrics.

CF datasets differ in the number of user ratings and/or item ratings per user and/or per item they contain, with the ratio $\frac{\#ratings}{(\#users * \#items)}$ being referred to as *density*, while the quantity $1-density$ is referred to as *sparsity*. Datasets are considered to be very sparse if their density is $\ll 1\%$ [16]. These datasets exhibit lower rating prediction accuracy, since it is much more challenging to find near and, hence, reliable neighbours, making the selection of the user similarity metric(s) of great importance. For example, the work by [17] uses only the PCC metric, the work by [18] uses only the COS metric, the work by [19] uses both the Vector Similarity and the PCC, and the work by [20] uses the PCC, the COS, and the Jaccard Similarity Index.

1.1 Research questions

Considering the above, this work aims to answer the following important research questions (RQs):

- *RQ1:* Which user similarity metrics yield the best results when the CF algorithm is applied to sparse CF datasets?
- *RQ2:* Do other parameters of the CF algorithm (e.g. neighbour selection approach, rating prediction formula, dataset density) affect the performance of user similarity metrics and the relative ordering of the metrics' performance?
- *RQ3:* Considering that the COS and the PCC are the most frequently used metrics in CF research/practice in general and CF research/practice on sparse datasets in particular, should researchers and practitioners consider shifting to other metrics or widening their selections to include additional metrics?

1.2 Research objective

To answer the stated RQs, the aim of this work is to assess the effectiveness of user similarity metrics in sparse CF datasets, by conducting a comprehensive multi-parameter series of experiments, which:

1. includes 15 similarity metrics, commonly used in CF RecSys research,
2. uses both the top-k technique (which selects the k-nearest users for each user) and the correlation threshold technique (which sets a threshold and keeps only the users whose similarities with the active user, meet or exceed this threshold),
3. uses both the mean-centred formula and the weighted sum method in the rating prediction formulation phase,
4. includes 10 sparse CF datasets, from five different sources, commonly used in CF RecSys research, and
5. uses three rating prediction accuracy metrics, broadly used by researchers in the CF RecSys domain.

To ensure the reliability of the results, all similarity metrics are tested on the same 10 datasets, using the same rating prediction configurations (near neighbour selection methods, parameters related to neighbour selection, and rating prediction formulas).

In this work, only the basic similarity metrics for determining user likeness are considered. The hybrid and combined metrics that were introduced by recent research are not currently widely employed in CF approaches, and therefore, the effectiveness of these metrics will be analysed in our future work.

The remainder of the paper is ordered as follows: Sect. 2 overviews the related work. Section 3 presents the preliminaries and methodology of the CF procedure, including the CF user similarity metrics that are evaluated. Section 4 presents the parameters and settings of the experiment (datasets, metrics, alternatives, etc.), as well as the assessment of the user similarity metrics. Finally, Sect. 5 concludes the paper and outlines the future work.

2 Related work

CF algorithm accuracy is a very active research field, where some of the research works are applied to dense CF datasets, while others are applied to sparse CF datasets.

Regarding dense datasets, [21] introduces a CF technique that identifies groups of objects related to each other, in order to develop user-based local similarity models. This algorithm combines rating normalisation with automatic object clustering, to develop a vicinity model for each cluster. In the work reported in [21], the COS, the PCC, and the sigmoid function-based sigmoid function-based PCC metrics are used, in order to predict ratings from the MovieLens 100k old (100K ratings by 943 users on 1682 movies, giving a sparsity level of 93.7%), the MovieLens 1M (1,000,209 ratings by 6,040 users on 3,883 giving a sparsity level of 95.7%) and a subset of the Netflix (sparsity 98.9%) datasets, while the prediction error metrics used are the Root Mean Square Error (RMSE) (computed as $\sqrt{\frac{\sum_{i=1}^{N}(p_i - v_i)^2}{N}}$, where N is the number of measurements, v_i is the real value for measurement i and p_i the corresponding prediction) and Mean Absolute Error (MAE) (computed as $\frac{\sum_{i=1}^{N}|p_i - v_i|}{N}$). The work in [22] introduces INH-BP, a prognostic model that aims to improve the RecSys accuracy. This model adjusts the predictor for each active user by including an optimisation tool. A COS-based and a PCC-based CF techniques are used to predict ratings from the MovieLens Last (sparsity 98.3%) and the MovieLens 100K old (sparsity 93.7%) datasets, while the MAE is used as the prediction error metric. [23] introduces ΓUICF, a hybrid recommendation model which synthesises the gamma linear regression model with user-based and item-based CF targeting at modelling the scalability and sparsity and issues of the CF rating matrix. That work uses the COS and the PCC coefficient similarity metrics to predict ratings from the MovieLens 1 M (sparsity 95.8%) and the MovieLens 100K old (sparsity 93.7%) datasets, while the prediction error metrics used are the MAE and RMSE. [24] presents an item-based CF method that combines the item-based prediction technique with neighbour option strategy. First, that work proposes the Kullback–Leibler divergence, and then, it loosens the rating prediction dependency to explicitly entered ratings only and adjusts the rating prediction output. The experiments reported in [24] consider the PCC, the Jaccard Similarity Index, the mean absolute difference, the mean squared difference, the Adjusted COS similarity, and the Sigmoid PCC similarity metrics for rating prediction computation. Each of the metrics is used to generate predictions for the MovieLens 1 M (sparsity 95.8%) and the MovieLens 100K old datasets (sparsity 93.7%), while the prediction error metrics used are the MAE, the RMSE, the rate of successful predictions, and the F1-measure. The method introduced by [25] assesses user-based CF and matrix factorisation, using different dataset partitions, based on time, genre, and age, and presented a new hybrid algorithm by integrating time, genre, and age into the definition of the COS function. The authors of [25] use the PCC, the Euclidean Similarity, the COS Similarity, the Spearman Correlation Similarity, and the Jaccard coefficient in order to predict ratings from the MovieLens 100k (sparsity 93.7%) and the MovieLens 1 M (sparsity 95.8%). Concerning the accuracy of predictions, this was quantified using the RMSE error metric.

Although all the previous works involve the use and assessment of similarity metrics in CF algorithms, none of them is applied to sparse CF datasets, where it is much more difficult to find near and, therefore, reliable neighbours, which may lead to lower rating prediction accuracy, in many cases.

To this extent, the following research works belong to the second category, i.e. introduce CF algorithms that are applied (also) to sparse CF datasets. The work in [26] modifies the Proximity–Impact–Popularity vicinity metric, computed as the product of popularity, impact, and proximity values, by normalising each component's range into [0, 1], setting different weights to the aforementioned three components in each scenario. Furthermore, this work develops a modified prediction function which combines the user–user, the item–item, as well as the weighted average deviations to enhance the prediction accuracy. Eleven similarity metrics are used, including the PCC, the Jaccard coefficient, and the COS. That work also includes sparse datasets, such as the Epinions (sparsity 99.986%) and the CiaoDVD (sparsity 99.91%). [27] presents the CFVR algorithm which produces virtual ratings based on the users' real ratings and adds them to the rating matrix to effectively mitigate the low sparsity issue in sparse CF datasets. The PCC and the COS similarity metrics are utilised, and the evaluation process also includes sparse datasets, such as the Amazon's Videogames (sparsity 99.91%), the Digital Music (sparsity 99.7%), and Amazon Movies and TV (sparsity 99.93%). The work in [20] presents an algorithm that improves the PCC similarity metric by eliminating the rating styles differences. The algorithm uses the Bidirectional Encoder Representations from Transformers model to mine users' rating styles, and subsequently, a technique that eliminates the differences of users' rating styles is applied. The PCC is used to quantify user similarities and predict ratings from a pre-processed dataset of the Amazon Movies and TV core5 (final sparsity 99.4%). [28] employs numerous vicinity metrics based on both item–item and user–user to produce rating predictions in CF datasets. Furthermore, they present an algorithm that applies feature vectors to similarity interest values, derived from the rating matrix, based on the number of personal interests, as well as the user–user rating interpersonal diffusion behaviour. The CiaoDVD (sparsity 99.91%) is also among the datasets used. The work in [29] introduces a method that

includes a pre-processing step that assesses the trustworthiness of rating predictions, and incorporates the ones deemed as “trustworthy" into the rating matrix, thereby resulting in sparsity reduction and upgrades for both prediction coverage and quality in sparse CF datasets. This work employs the COS and the PCC metrics. The CiaoDVD (sparsity 99.91%), the Amazon Movies and TV core5 (sparsity 99.98%), and the Amazon Videogames core5 (sparsity 99.95%) are among the datasets used. The work in [30] presents an Improved Triangle similarity metric that is complemented with the users rating preference behaviour. The presented similarity metric not only considers items commonly evaluated by pairs of users, but also considers the ratings of products that are not commonly evaluated. It uses eight user similarity metrics, including the COS, the Euclidean similarity, the Jaccard Index, and the Constrained PCC. Among the datasets used are also the CiaoDVD (sparsity 99.91%) and the Epinions (sparsity 99.99%). All the aforementioned works listed for the second category use the MAE and the RMSE as prediction error metrics, while [26] and [30] also employ the F1 measure.

In the last five years, a number of research works focusing on evaluation reviews of CF user similarity metrics have been published; however, they mainly utilise high density datasets. More specifically, [31] presents an evaluation review of 13 user similarity metrics, using one NN formulation method, two rating prediction formulas, and three prediction error metrics on three datasets (MovieLens 100k, MovieLens 1 M, and Jester), which are of relatively high density (density $\geq$ 4.2%). [32] presents an evaluation review of 33 user similarity metrics (including hybrid and combined ones) using one NN formulation method, one rating prediction formula, and three prediction error metrics on two datasets (MovieLens 100k and MovieLens 1 M), which are of high density. [33] presents an evaluation review of 29 user similarity metrics (including hybrid and combined ones) using one NN formulation method, one rating prediction formula, and four prediction error metrics on three datasets (Film Trust, Movielens-100K, Movielens- 1 M). While the datasets are of relatively high density, the authors of this work present experiments where subsets of the datasets are used for training, simulating thus a sparse dataset, with the training portion of the simulated datasets exhibiting sparsity up to 99.6%. However, this sparsity level lags behind the corresponding levels of natively sparse datasets, which typically exceeds 99.9%. Furthermore, the authors point out that in their experiments, higher training dataset sizes are correlated with higher rating prediction accuracy.

An interesting approach is taken in works [34, 35], which aim to promote the understanding of regularities that are inherent within the data, allowing to assess the feasibility and achievability of attaining a specified level of accuracy. This will facilitate goal setting in recommender systems and direct algorithm planning before their implementation.

Notably, similarity evaluation metrics are assessed and compared in other domains and contexts too, such as link prediction [36, 37]. [38] presents an evaluation review of 7 user similarity metrics, using one NN formulation method, one rating prediction formula and two prediction error metric on one dataset. However, the dataset is both artificial and dense, while it is additionally severely limited in size. The work in [39] presents an evaluation review of 4 user similarity metrics using one NN formulation method, one rating prediction formula, and two prediction error metrics on 2 datasets (MovieLens 100K and FilmTrust datasets), which are, however, of high density. An overview of the aforementioned works that focus on the evaluation of CF similarity metrics is found in Table 1.

Considering the above, there is a research gap for an extensive comparative evaluation of similarity metrics in the context of natively sparse datasets. Sparse CF datasets, which are common in both the industry and the research domains, encounter lower rating prediction accuracy, since it is much harder to find close and, hence, reliable NNs [29, 40, 41]. Therefore, the selection of a more efficient similarity metric is bound to have a considerable effect on the overall performance of the RecSys. A comparative evaluation would provide a comprehensive coverage of different similarity metrics used in CF and take into account multiple accuracy quantification metrics, NN selection methods, and rating prediction calculation formulas. Finally, multiple natively sparse datasets from diverse domains should be utilised. In this paper, we present a study that aims to fill this this gap, and highlight the most effective user similarity metric(s) in sparse CF datasets. This would aid researchers and practitioners alike to tune their RecSys by selecting the most appropriate similarity metric, achieving thus more accurate predictions, which in turn lead to more successful recommendations [42, 43]. For e-commerce systems in particular, successful recommendations are critical, since they constitute an important determinant of user satisfaction [44, 45].

3 Preliminaries and Methodology

In CF, before implementing a rating prediction process, designers have to take three major decisions [10, 31]:

1. the similarity metric they want to use;
2. the criteria that will be applied for determining the NNs for users; and
3. the formula that will be used to calculate the prediction numeric value.

Table 1 Comparison with recent (last 5 years) review works of CF metric evaluation

Ref	# metrics	# datasets (# sparse)	# item domains	NN form. methods	# Pred. comp. formulas	# Pred. error metrics
[31]	13	3 (0)	2 (Movies, Jokes)	top-K	Weighted sum, Mean-centred	MAE (NMAE), RMSE, R^2
[32]	33 (incl. hybrid)	2 (0)	1 (Movies)	top-K	Mean-centred	MAE, RMSE and F1
[33]	29 (incl. combined)	3 (3)	1 (Movies)	top-K	Weighted sum	MAE, RMSE, R, F1
[38]	7	1 (0)	0 (artificial)	top-K	Weighted sum	MAE
[39]	4	2(0)	1 (Movies)	top-K with clustering	Weighted sum	MAE, within-cluster
Presented work	15	10 (10)	8 (Movies & Series, Books, Videogames Cell Phones and Accessories, DVDs Kindle Store Digital Music Musical Instruments)	thresholded, and top-K	Weighted sum and Mean-centred	MAE (NMAE), RMSE, F1

This work focuses on the evaluation of the effectiveness of similarity metrics and therefore considers the most widely used options for the first factor. However, since existing research has provided evidence that the overall CF algorithm effectiveness is influenced by the interplay between all three factors, in this work we also take into account the most widely used techniques for the other two decisions in CF research, exploring the effectiveness of all possible combinations of the selected factor instantiations.

More specifically, in regard to the NN selection criteria (decision 2), this work tests both the top-k approach (for each active user, the k-nearest users are pre-selected as NNs) and the similarity threshold approach (for each active user, the users whose similarities exceed this threshold are pre-selected as NNs). In regard to the rating prediction calculation formula (decision 3), both the mean-centred formula and the weighted sum method are tested. As a result, since this work tests all the combinations of the aforementioned alternatives, a holistic view of the user similarity metric (decision 1) effectiveness is obtained.

In the following subsections, the process of the rating prediction formulation will be analysed, for self-containment purposes, along with the 15 user similarity metrics which will be evaluated.

3.1 CF user similarity metrics

In CF, numerous metrics are available to quantify the similarity among a pair of users. Recent research works commonly utilise the COS (or Vector) Similarity and the PCC. This paper evaluates 15 CF user similarity metrics as follows: (1) the Jaccard Index [46], (2) the Manhattan (or taxicab or city blocks) distance [47], (3) the Euclidean distance [43], (4) the Chebyshev distance [48], (5) the PCC [49], (6) the Constrained PCC [50], (7) the Sigmoid PCC [33], (8) the Vector or COS similarity [49], (9) the Adjusted COS [31], (10) the Spearman rank correlation [51], (11) the Kendall's Tau correlation [31], (12) the Mean Square Difference-based similarity [52], (13) the Normalised Sum of Multiplications [33], (14) the Adjusted Rand Index [53], and (15) the Adjusted Mutual Information [54].

The selection of the similarity metrics was initially seeded by five recent surveys that evaluated similarity metrics [31–33, 38, 39], and was further verified by searching the Scopus database for works that are more recent than the surveys and introduce new metrics for use in collaborative filtering, using the query

TITLE-ABS-KEY("collaborative filtering") AND TITLE-ABS-KEY("similarity metric") AND PUBYEAR > 2021

In this search, no new metrics were identified (Table 2).

The similarity metrics are briefly overviewed in the following subsections. The notations used in the definitions and formulas of the similarity metrics are presented in Table 3.

3.1.1 Jaccard Index

The Jaccard Index (JACC) between two CF users U and V is defined as the ratio between the intersection of their common

Table 2 Similarity measures considered in recent surveys [31–33, 38, 39]

Similarity metric	Fkih [31]	Khojamli [32]	Amer [33]	Jain [38]	Bojorque [39]
Jaccard	✓	✓	✓	✓	✓
Manhattan	✓	✓		✓	
Euclidean	✓	✓		✓	✓
Chebyshev	✓				
Pearson	✓	✓	✓	✓	✓
Constrained Pearson			✓	✓	
Sigmoid Pearson			✓	✓	
Cosine	✓	✓	✓	✓	
Adjusted cosine	✓		✓		
Spearman rank	✓	✓	✓		
Kendall's Tau	✓				
Mean square difference		✓	✓		✓
Normalised sum of multiplications			✓		
Adjusted Rand index	✓				
Adjusted mutual information	✓				

Table 3 Notations used in the user similarity metrics

Notation	Description
$r_{U,i}$	The numeric rating user U has given to item i
$\bar{r_i}$	The average value of all ratings entered for item i
$\bar{r_U}$	The average value of all ratings entered by user U
I_U	The set of items user U has rated
$\|I_U\|$	The cardinality of I_U, i.e. the number of items user U has rated
$I_{U,V}$	The set of items both users U and V have rated (i.e. their intersection)
Rank($r_{U,i}$)	The rank of item i in U's rating set (when sorting all items rated by user U in descending rating value order)
range	The range of the rating scale (i.e. maximum rating value - minimum rating value)
r_{med}	The median value in the rating scale
NU	The number of users in the dataset
TNR	The total number of ratings in the dataset
$ANRU$	The average number of ratings per user, i.e. $\frac{TNR}{NU}$

evaluated items and the union of their evaluated items, as shown in formula 1:

$$JACC(U, V) = \frac{|I_{U,V}|}{|I_U \cup I_V|} \quad (1)$$

The range of this metric is [0, 1], where higher values signify higher similarity between the two users. The Jaccard Index takes into account only interactions with the items, whereas it does not consider whether items have been rated favourably or not [55].

The computational complexity of the $JACC$ metric between two users U and V is $O(|I_U| + |I_V|)$, assuming an efficient implementation where set unions and intersections are computed using a hash set implementation, in which lookups are performed at a cost of $O(1)$. The complexity of computing the $JACC$ metric for all user pairs is $O(NU^2 * ANRU)$.

3.1.2 Manhattan Index

The Manhattan distance (MANH) between two users U and V (also known as the taxicab distance and city blocks distance) considers the 1-norm (i.e. the aggregate of the absolute value of differences) of the ratings of these users on the same items, as depicted in formula 2; this measure is subsequently normalised using formula 3, to compute the MANH similar-

ity metric [47].

$$d\_MANH(U,V) = \sum_{i \in I_{U,V}} |\, r_{U,i} - r_{V,i} |\tag{2}$$

$$MANH(U,V) = \frac{1}{1 + d\_MANH(U,V)}\tag{3}$$

The range of this metric is [0, 1], where higher values signify higher similarity between the two users.

The computational complexity of the $MANH$ metric between two users U and V is $O(|I_U| + |I_V|)$, corresponding to the computation of the set of commonly rated items. Then this set, whose cardinality is less than $|I_U| + |I_V|$ is iterated upon to compute d_MANH, which is subsequently used to compute $MANH$. The complexity of computing the $MANH$ metric for all user pairs is $O(NU^2 * ANRU)$.

3.1.3 Euclidean Index

The Euclidean distance (EUCL) between two users U and V considers the square root of the 2-norm (i.e. the sum of squares of the differences) of the ratings of these users on the same items, as depicted in formula 4.

$$d\_EUCL(U,V) = \sqrt{\sum_{i \in I_{U,V}} (r_{U,i} - r_{V,i})^2}\tag{4}$$

The result of formula 4 is subsequently normalised using formula 5 to compute the EUCL CF similarity metric.

$$EUCL(U,V) = \frac{1}{1 + d\_EUCL(U,V)}\tag{5}$$

We can observe that, while the Manhattan distance treats all the deviations equally, the Euclidean distance penalises the greater ones more severely. The range of this metric is also [0, 1], where higher values signify higher similarity between the two users [43].

The complexity of computing the $EUCL$ metric between two users U and V is $O(|I_U| + |I_V|)$, corresponding to the computation of the set of commonly rated items. Then the set, whose cardinality is less than $|I_U| + |I_V|$ is iterated upon to compute d_EUCL. Finally, d_EUCL is used to compute $EUCL$. The complexity of computing the $EUCL$ metric for all user pairs is $O(NU^2 * ANRU)$.

3.1.4 Chebyshev Index

The Chebyshev distance (CHEB) between two users U and V considers the infinity norm (also known as the maximum norm), which is equal to the maximum difference between the ratings of the common evaluated items, as depicted in formula 6.

$$d\_CHEB(U,V) = \max_{i \in I_{U,V}} |\, r_{U,i} - r_{V,i} |\tag{6}$$

Formula 6 is then normalised using equation 7 to compute the CHEB CF similarity metric [48].

$$CHEB(U,V) = \frac{1}{1 + d\_CHEB(U,V)}\tag{7}$$

The computational complexity of the $CHEB$ metric is $O(|I_U| + |I_V|)$, corresponding to the computation of the set of commonly rated items. Then this set, whose cardinality is less than $|I_U| + |I_V|$ is iterated upon to compute d_CHEB. Finally, d_CHEB is used to compute $CHEB$. The complexity of computing the $CHEB$ metric for all user pairs is $O(NU^2 * ANRU)$.

3.1.5 Pearson Correlation Coefficient

The PCC between two users U and V is probably the most commonly used similarity formula in CF, used by numerous research works that propose and evaluate CF algorithms [25, 56, 57]. It measures the linear correlation of U's and V's rating sets, while it is computed as shown in formula 8.

$$PCC(U,V) = \frac{\sum_{i \in I_{U,V}} (r_{U,i} - \bar{r_U}) * (r_{V,i} - \bar{r_V})}{\sqrt{\sum_{i \in I_{U,V}} (r_{U,i} - \bar{r_U})^2} * \sqrt{\sum_{i \in I_{U,V}} (r_{V,i} - \bar{r_V})^2}}\tag{8}$$

The range of this metric is [-1, 1], where higher values signify higher similarity between the two users [42].

In order to compute the PCC metric between two users U and V, the mean rating value for each of the users needs to be first computed. The complexity of this operation is $O(|I_U|) + O(|I_V|)$. Afterwards, the set of commonly rated items is determined. The complexity of this task is $O(|I_U| + |I_V|)$. Finally, this set (whose cardinality is less than $|I_U| + |I_V|$) is iterated upon to compute the numerator and the two factors in the denominator of the fraction in formula 8. Therefore, the complexity of this task is $O(|I_U| + |I_V|)$. The complexity of computing the PCC metric for all user pairs is $O(NU^2 * ANRU)$, while notably the mean rating for each user can be computed only once and stored for further perusal, delivering a more efficient implementation. Nevertheless, this optimisation does not reduce the overall complexity, since this is dominated by the computation of commonly rated items between each user pair.

3.1.6 Constrained Pearson Correlation Coefficient

The constrained PCC (CPCC) is a variant of the PCC. It is differentiated from the original PCC, through the replacement of the average value of all ratings entered by each user $\bar{r_U}$ and $\bar{r_V}$, in the numerator, by the median value in the rating scale (r_{med}), as shown in formula 9.

$$CPCC(U,V) = \frac{\sum_{i \in I_{U,V}} (r_{U,i} - r_{med}) * (r_{V,i} - r_{med})}{\sqrt{\sum_{i \in I_{U,V}} (r_{U,i} - r_{med})^2} * \sqrt{\sum_{i \in I_{U,V}} (r_{V,i} - r_{med})^2}} \tag{9}$$

The range of this metric is [-1, 1], where higher values signify higher similarity between the two users [50].

In order to compute the $CPCC$ metric between two users U and V, the median of all ratings needs to be firstly computed. The complexity of this task is $O(TNR)$ [58], i.e. linear in relation to TNR, which is the set over which the median is computed. Subsequently, the commonly rated items of the two users are determined, and this set is iterated upon to calculate the nominator and the factors in the denominator of equation 9. Thus, the overall complexity of the $CPCC$ metric computation between two users is $O(TNR)$, since TNR dominates the sum of the number of ratings of the two users. In order to compute the $CPCC$ metric for all user pairs, the median of all ratings is initially computed only once. Subsequently, for each user pair computations of complexity $O(|I_U| + |I_V|)$ need to be performed. Therefore, the overall complexity is $O(NU^2 * ANRU + TNR)$. However, since $TNR = n * ANRU$, the overall complexity is reduced to $O(NU^2 * ANRU)$.

3.1.7 Sigmoid Pearson Correlation Coefficient

The Sigmoid PCC (SPCC) is another variant of the PCC, which augments the similarity between users who share a lot of common rated items, as shown in formula 10.

$$SPCC(U,V) = PCC(U,V) * \frac{1}{1 + e^{-\frac{|I_{U,V}|}{2}}} \tag{10}$$

The range of this metric is [-1, 1], where higher values signify higher similarity between the two users [33].

The computation of the $SPCC$ metric requires the value of the PCC, which dominates the computational complexity. Therefore, the overall computational complexity of the $SPCC$ metric is $O(NU^2 * ANRU)$.

3.1.8 Vector or Cosine similarity

The COS or Vector Similarity is a very frequently used CF similarity metric [49, 59, 60]. It is derived from the Euclidean dot product formula of the two user rating vectors. It is calculated as depicted in formula 11.

$$COS(U,V) = \frac{\sum_{i \in I_{U,V}} r_{U,i} * r_{V,i}}{\sqrt{\sum_{i \in I_{U,V}} r_{U,i}^2} * \sqrt{\sum_{i \in I_{U,V}} r_{V,i}^2}} \tag{11}$$

In general, the range of the COS is [-1, 1], and higher values signify higher similarity between the two users. For datasets having non-negative rating values, such as the ones used in our experimental evaluation, the range of this metric is [0, 1].

In order to compute the COS metric between two users U and V, the set of commonly rated items need to be determined. The complexity of this operation is $O(|I_U| + |I_V|)$. Subsequently this set, whose cardinality is less than $|I_U| + |I_V|$ is iterated upon to compute the nominator and the factors in the denominator of Eq. 11. Therefore, the complexity of the overall computation is $O(|I_U| + |I_V|)$. In order to compute the COS metric for all user pairs, the related complexity is $O(NU^2 * ANRU)$.

3.1.9 Adjusted Cosine measure

The Adjusted Cosine measure (ACOS) is a variant of the COS. It differentiates from the original COS because it normalises all the rating values by item, as depicted in formula 12.

$$ACOS(U,V) = \frac{\sum_{i \in I_{U,V}} (r_{U,i} - \bar{r_i}) * (r_{V,i} - \bar{r_i})}{\sqrt{\sum_{i \in I_{U,V}} (r_{U,i} - \bar{r_i})^2} * \sqrt{\sum_{i \in I_{U,V}} (r_{V,i} - \bar{r_i})^2}} \tag{12}$$

The range of this metric is [-1, 1], where higher values signify higher similarity between the two users [31].

The computation of the $ACOS$ metric requires the computation of the means of ratings for each item. The complexity of this task is $O(TNR)$, where TNR is the overall number of ratings. Then, for each pair of users U and V the set of commonly rated items needs to be computed, and the complexity of this computation is $O(|I_U| + |I_V|)$. Considering that the means of ratings for each item is calculated only once, computation of the $ACOS$ metric for all user pairs has an overall complexity of $O(TNR) + O(NU^2 * ANRU)$. However, since $TNR = NU * ANRU$, the overall complexity is reduced to $O(NU^2 * ANRU)$.

3.1.10 Spearman rank correlation

The Spearman rank correlation (SP) metric between two sets of users' co-rated items computes the monotonic relationship of the aforementioned sets and is considered as the PCC between the rank variables. To compute the value of SP, the ratings of each user U are ordered in descending order, and the first (i.e. the highest ranked) one is assigned the rank 1, the next one is assigned the rank 2, etc. In the event of a tie, all identical ratings by the same user are assigned the same rank. When all rating values entered by the same user are distinct (i.e. a total ordering exists for rating values entered by the same user), SP is calculated using formula 13. In the general case where duplicate values for ratings may exist, SP is calculated using formula (14). In formula 14, R_U and R_V designate the ratings of the commonly rated items by these users, after these ratings have been converted to ranks, and $\sigma(R_x)$ designates the standard deviation of rank variable R_x.

$$SP(U,V) = 1 - \frac{6 * \sum_{i \in I_{U,V}} (Rank(r_{U,i}) - Rank(r_{V,i}))}{|I_{U,V}| * (|I_{U,V}| - 1)} \tag{13}$$

$$SP(U,V) = \frac{covariance(R_U, R_V)}{\sigma(R_U) * \sigma(R_V)} \tag{14}$$

The range of this metric is [-1, 1], where higher values signify higher similarity between the two users [51].

The complexity of computing the SP metric for two users U and V is $O(RlogR)$ [61], where R is the number of elements in both sets (i.e. equal to $|I_U| + |I_V|$). In order to compute the SP metric for all user pairs, the related complexity is $O(NU^2 * ANRU * log(ANRU))$.

3.1.11 Kendall's Tau correlation

The Kendall's Tau correlation (TAU) counts the number of pairwise disagreements between the two ranking user sets. The larger the distance, the more dissimilar the two lists are [31]. It is found that the number of swaps that the bubble-sort algorithm needs to place user U's rating list in the same order as user V's rating list is equivalent to the TAU distance. Due to this fact, the TAU is also called the "bubble sort distance". It is calculated using formula 15:

$$TAU(U,V) = \frac{c+d}{c-d} \tag{15}$$

where c is the number of the concordant rating pairs between the two user rating sets and d is the number of the discordant rating pairs between the two user rating sets. Two rating pairs $(r_{U,i}, r_{U,j})$ and $(r_{V,i}, r_{V,j})$ are considered concordant, iff, after converting ratings to rankings, both elements are either greater than, equal to, or less than the corresponding elements of the other pair. This is expressed formally in equation 16.

$$\begin{aligned}&(rank(r_{U,i}) > rank(r_{U,j}) \wedge rank(r_{V,i}) > rank(r_{V,j})) \vee \\ &(rank(r_{U,i}) = rank(r_{U,j}) \wedge rank(r_{V,i}) = rank(r_{V,j})) \vee \\ &(rank(r_{U,i}) < rank(r_{U,j}) \wedge rank(r_{v,i}) < rank(r_{V,j}))\end{aligned} \tag{16}$$

Rating pairs that are not concordant are termed as discordant. The range of this metric is [-1, 1], where higher values signify higher similarity between the two users.

The computational complexity for computing the TAU metric between two users U and V is $O(R \log R)$, where R is the number of individuals in both sets (i.e. equal to $|I_U| + |I_V|$). In order to compute the TAU metric for all user pairs, the related complexity is $O(NU^2 * ANRU * log(ANRU))$.

3.1.12 Mean square difference-based similarity

The mean square difference-based similarity (MSD) puts more emphasis on major rather than minor differences. It is calculated by squaring the difference of the commonly evaluated item ratings and then dividing the sum of the aforementioned squares by the square of the range of the rating scale (maximum rating value − minimum rating value) multiplied by the user commonly evaluated item quantity [33], as shown in formula 17:

$$d\_MSD(U,V) = \frac{\sum_{i \in I_{U,V}} (r_{U,i} - r_{V,i})^2}{(range)^2 * |I_{U,V}|} \tag{17}$$

The distance metric is then used to compute the similarity metric, using formula 18.

$$MSD(U,V) = 1 - d\_MSD(U,V) \tag{18}$$

The range of this metric is [0, 1], where higher values signify higher similarity between the two users [52, 62].

In order to compute the MSD metric between two users U and V, the set of commonly rated items by both users needs to be computed; the relevant computation for this task is $O(|I_U| + |I_V|)$. This set is subsequently iterated upon to compute the nominator, while the computation of the denominator is of complexity $O(1)$. To calculate MSD metrics for all user pairs, the relevant complexity is $O(NU^2 * ANRU)$.

3.1.13 Normalised Sum of Multiplications

The Normalised Sum of Multiplications (NSM) shares the same numerator with the COS formula. However, its denominator is the sum of the squares of the maximum ratings given to each individual item by the two users, as shown in for-

mula 19.

$$NSM(U,V) = \frac{\sum_{i \in I_{U,V}} r_{U,i} * r_{V,i}}{\sum_{i \in I_{U,V}} max(r_{U,i}^2, r_{V,i}^2)} \quad (19)$$

In general, the range of the NSM metric is [-1, 1] and higher values signify higher similarity between the two users. For datasets having non-negative rating values only, such as the ones used in our experimental evaluation, the range of this metric is [0, 1] [33].

In order to compute the NSM metric between two users U and V, the set of commonly rated items by both users needs to be computed. The relevant computation for this task is $O(|I_U| + |I_V|)$. This set is subsequently iterated upon to compute the nominator, while the computation of the denominator is also of linear complexity $O(|I_U|) + O(|I_V|)$. To calculate NSM metrics for all user pairs, the relevant complexity is $O(NU^2 * ANRU)$. Notably, the maximum rating for each user needs to be computed only once; however, this does not reduce the complexity, because it is dominated by the computation of commonly rated items.

3.1.14 Adjusted Rand Index

The Adjusted Rand Index (ARI) is a similarity metric between two data clusterings. ARI takes into account the fact that some agreement between two clusterings can occur by chance, and it adjusts the Rand Index (the basic measure of similarity) to account for this possibility [53]. In the domain of CF, for each two users U, V their ratings on commonly reviewed items $I_{U,V}$ are extracted, and afterwards each user's ratings on these items are clustered separately forming two clusterings $C_U = \{c_{U,1}, c_{U,2}, \ldots, c_{U,m}\}$ and $C_V = \{c_{V,1}, c_{V,2}, \ldots, c_{V,n}\}$. Each data point set $c_{X,i}$ includes ratings by user X (either U or V) that share the same rating value. Afterwards, the similarity between the two clusterings is quantified. ARI is calculated using formula 20:

$$ARI(U,V) = \frac{RI(U,V) - E\{RI(U,V)\}}{max(RI(U), RI(V)) - E\{RI(U,V)\}} \quad (20)$$

where RI is the Rand Index metric and E{RI(U,V)} is the expected Rand index between the two clusters U and V. The range of this metric is [-1, 1], where higher values signify higher similarity between the two users.

The complexity for computing the ARI metric for a pair of users U and V is $O(R + KL)$, where K and L are the sizes of the two clusterings to be compared and R is the number of individuals in both sets [63]. Since the elements are values and one cluster is formed for each distinct value, K and L have an upper bound of $range$; therefore, the complexity for computing the ARI metric for a pair of users U and V is $O((|I_U| + |I_V|) + range^2)$. In order to compute the ARI metric for all pairs of users, the related complexity is $O(NU^2 * (ANRU + range^2))$.

3.1.15 Adjusted Mutual Information

The Adjusted Mutual Information (AMI) metric is based on the Mutual Information (MI) metric, which is widely used in the information theory research subject. AMI calculates the statistical correlation between a pair of users U and V as shown in formula 21.

$$ARI(U,V) = \frac{MI(U,V) - E\{MI(U,V)\}}{max(H(U), H(V)) - E\{MI(U,V)\}} \quad (21)$$

where E{MI(U,V)} is the expected MI and H(U) and H(V) are the entropies of these two users, respectively. The range of the AMI metric is [-1, 1], where higher values signify higher similarity between the two users [54].

The complexity of computing the AMI metric for a pair of users U and V is $O(max(k,l)R)$ [64], where:

- R is the total number of items in both compared clusterings (i.e. equal to $O(|I_U| + |I_V|)$)
- k, l are the number of clusters in the two sets. Since the elements of both clusterings are ratings and a distinct cluster is formed for each rating value, the maximum number of clusters is equal to $range$.

Considering the above, the complexity of computing the AMI metric for a pair of users U and V is $O(range * (|I_U| + |I_V|))$. In order to compute the AMI metric for all pairs of users, the related complexity is $O(NU^2 * range * ANRU)$.

3.1.16 Summary of metrics

In Table 4, we present a brief summary of the metrics, discussing aspects according to their design rationale and functionality. Their performance in the context of recommender systems operating on sparse datasets is discussed in Sect. 4.

3.2 NN selection

For the selection of the set of NNs of a user U, two methods are proposed in the literature:

1. the similarity (or correlation) threshold: the algorithm uses as NNs the users V for which the similarity values with U surpass a pre-selected threshold THR [65, 66]; and

Table 4 Summary of similarity metrics

Similarity metric	Complexity	Notes
Jaccard	$O(NU^2 * ANRU)$	Captures only co-rating of items, not rating preferences
Manhattan	$O(NU^2 * ANRU)$	Users with many commonly rating items are bound to have lower similarities, since the metric is not amortised according to the number of considered ratings. The MANH and EUCL metrics are also sensitive to the rating scale.
Euclidean	$O(NU^2 * ANRU)$	
Chebyshev	$O(NU^2 * ANRU)$	Considers only the maximum distance of ratings, which may not be representative of the users' rating behaviour
Pearson	$O(NU^2 * ANRU)$	May favour users with high rating variance; additionally it exhibits sensitivity to outliers (few extreme ratings may excessively affect the similarity metric value)
Constrained Pearson	$O(NU^2 * ANRU)$	Demoted personalisation compared to PCC due to the use of the global rating median instead of the personal rating average. The metric is also sensitive to outliers
Sigmoid Pearson	$O(NU^2 * ANRU)$	The use of the Sigmoid function increases the density of values towards the scale middle, therefore blurring differences that may be of high discriminating value. It introduces the need to configure the slope of the sigmoid function, which considerably affects performance. A single slope is used for all users, therefore individual rating biases may not be captured appropriately
Cosine	$O(NU^2 * ANRU)$	Disregards the magnitude of the ratings, as well as differences in uses of the rating scale by individual users (lenient vs. strict), only considering angles. If only a few common ratings exist, small divergencies in ratings may lead to disproportionate variations in the result
Adjusted cosine	$O(NU^2 * ANRU)$	Tackles some issues present in COS, at the expense of additional computational load. For users with small rating variances, the metric may underestimate the actual similarity
Spearman rank	$O(NU^2 * ANRU * log(ANRU))$	Spearman rank and Kendall's Tau disregard absolute values in favour of rankings. If only a few common ratings exist, small divergencies in ratings may lead to disproportionate variations in the result. Both metrics incur higher computing overhead.
Kendall's Tau	$O(NU^2 * ANRU * log(ANRU))$	
Mean square difference	$O(NU^2 * ANRU)$	Does not adjust according to personal rating biases, therefore user rating strictness/leniency is not amortised. The importance of large differences are overemphasised, reducing similarities excessively
Normalised sum of multiplications	$O(NU^2 * ANRU)$	Does not adequately handle different rating practices, therefore users having highly similar rating patterns but different degrees of strictness/leniency will be assigned low similarity values
Adjusted Rand index	$O(NU^2 * (ANRU + range^2))$	The ARI and AMI metrics consider only rating clusterings disregarding actual values, therefore clusters corresponding to different rating values may be matched. The magnitude of rating differences is also not taken into account, due to the adoption of a binary logic examining the presence or not of an item in a cluster. Both metrics incur higher computational load.
Adjusted mutual information	$O(NU^2 * range * ANRU)$	

2. the KNNs: the algorithm uses as NNs the K users found to have the highest similarity values with the active user [67, 68].

In this work, we use both methods. Regarding the threshold-based method, we conduct experiments with three settings, $THR = 0.0$, $THR = 0.25$, and $THR = 0.5$. The reason behind this option is that, as shown in the previous subsection, the similarity metrics have divergent value

ranges (some have a range of [0,1], while others have a range of $[-1, 1]$). Thus, we strive to cover all the instances. In regard to the KNNs method, we conduct experiments with two settings, $K = 250$ and $K = 500$, following the selection of K in related works [22, 31, 68] and given the fact that in (very) sparse datasets higher numbers of K are needed in order to achieve satisfactory coverage, considering that—due to high sparsity—each of the NNs can only contribute towards the rating prediction of few items only.

3.3 Rating prediction value formulation

The most popular and widely used functions to formulate the rating prediction value in the literature are the weighted sum function and the mean-centred prediction function [69–72]. In more detail:

1. the weighted sum function, where in order to predict the rating $p_{U,I}$ that a user U would assign to item i, the ratings on U's NNs for item i are considered, with each V's rating (V ∈ NN(U)) weighted according to the U-V similarity. The U-V similarity value (or vicinity value) is derived from the similarity method used (c.f. Sect. 3.1). This is formally expressed in formula 22:

$$p_{U,i} = \frac{\sum_{V \in NN_U} sim(U, V) * r_{V,i}}{\sum_{V \in NN_U} sim(U, V)} \tag{22}$$

2. the mean-centred prediction function, which extends the weighted sum function by compensating for divergent rating practices employed by different users, i.e. the fact that some users assign ratings more strictly while others are more lenient. This is achieved by subtracting the mean of the corresponding NN's rating from each NN's rating. The outcome of this calculation is then adjusted by the mean rating value of the target user U, to produce the final rating prediction numeric value. This is formally expressed in formula 23:

$$p_{U,i} = \bar{r_U} + \frac{\sum_{V \in NN_U} sim(U, V) * (r_{V,i} - \bar{r_V})}{\sum_{V \in NN_U} sim(U, V)} \tag{23}$$

In this work, we use both rating prediction formulation functions.

4 Experimental settings, evaluation, and results

In this section, we evaluate the 15 similarity metrics, presented in Sect. 3.1, primarily in regard to rating prediction accuracy and secondarily in regard to rating prediction coverage (i.e. the percentage of rating predictions that the CF algorithm is able to calculate). In regard to prediction accuracy, we consider the three most used ones in CF prediction research, the F1-measure, the RMSE, and the MAE. The main difference between the latter two rating prediction error metrics is that the MAE treats all divergences uniformly, while the RMSE penalises the larger ones harsher. For the F1-measure, we adopt the approach followed by numerous works including [19, 73], where the recommendation for a user U includes all items for which the prediction value is included within the upper 30% of the rating scale. For the 9 datasets that have a ratings range of [0, 5], the rating prediction threshold for an item to be included in the recommendation is 3.5. The respective threshold for the book-crossing dataset, whose rating range of [0, 10], is 7.

Our work focuses on 10 sparse CF datasets. As noted in Sect. 1, a dataset is considered very sparse if $\frac{\#ratings}{\#users * \#items} \ll 1\%$ [16]. The datasets used in our work, along with their attributes, are summarised in Table 5. Their sparsity spans from 99.76% (sparse dataset) to 99.997% (highly-sparse dataset). Their inclusion ensures that all the levels of sparsity are considered in our experimentation. Furthermore, we diversify the dataset sources, utilising datasets from Amazon, Yahoo, Epinions.com, and others. To ensure that the evaluation is not biased by the item domain, the selected datasets span across several product fields, from Music and Movies, to Books and Videogames.

In the following subsections, we firstly analyse the settings of the experiment, while subsequently we present and discuss the evaluation results.

4.1 Experimental settings and procedure

For each dataset, we follow the 10-fold cross-validation process [74, 75], where the original dataset is split into 10 folds, and each time the training set is derived from the union of the 9 folds, while the test set is the remaining 1 fold. After all 10 iterations are performed, the 10 prediction results are merged. From each dataset, only the tuple <user_id, item_id, numeric_rating> is used. Any additional information that may be available in the dataset, such as user demographic information, item product category, item features (e.g. price, condition, etc.), rating reviews, user relations (e.g. social), and others is disregarded. This is to ensure that (a) results are not affected by the introduction of other methods and algorithms or specialised techniques and (b) the results are

Table 5 Datasets used in the experiments

Dataset Name	Sparsity	#users	#items	#ratings	Ratings range
R4-Yahoo! Movies[1]	99.76%	8K	12K	221K	1–5
Book-crossing (books)[2]	99.997%	105K	341K	1 M	0–10
Amazon VideoGames[3]	99.951%	17K	55K	473K	1–5
Amazon CellPhones and Accessories[3]	99.985%	48K	157K	1.1M	1–5
Amazon Movies and TV[3]	99.982%	60K	298K	3.3M	1–5
Amazon Kindle Store[3]	99.984%	99K	140K	2.2M	1–5
Amazon Digital Music[3]	99.926%	12K	17K	145K	1–5
Amazon Musical Instruments[3]	99.925%	11K	28K	219K	1–5
Epinions (general consumer reviews)[4]	99.986%	22K	296K	922K	1-5
CiaoDVD (DVDs)[5]	99.91%	2K	17K	36K	1-5

[1] https://webscope.sandbox.yahoo.com/catalog.php?datatype=r [2] https://www.kaggle.com/datasets/somnambwl/bookcrossing-dataset [3] https://nijianmo.github.io/amazon/index.html [4] https://www.kaggle.com/datasets/masoud3/epinions-trust-network [5] https://www.cse.msu.edu/~tangjili/datasetcode/truststudy.htm

generalisable for all cases, since these data are utilised by every CF-based RecSys.

Within each iteration, the vicinities of all pairs of users are computed, utilising all the 15 similarity metrics analysed in the previous section. For the majority of the 15 metrics, it suffices to have 1 co-rated item between two users, to calculate their similarity. However, for some metrics, the calculation of the similarity requires at least two co-rated items, due to the mathematical structure of the respective formula (e.g. SP includes the standard deviation in the denominator, which requires at least two values in order to be calculated). Furthermore, some of the similarity metrics, such as the TAU, return the value of 1.0 (the highest value in the metric range) when only one co-rated item exists between the two users. As a result, in order (a) to overcome the aforementioned issue and (b) to ensure fairness among the 15 metrics in the context of the experiment, only users with at least 2 co-rated items are considered as potential NNs. The examination of different threshold values for the co-rated item quantity in the stage of NN selection is considered as part of our future work. Lastly, to ensure the reliability of the results, the baseline comparison is on the same 10 datasets, with the same parameter alternatives and settings for all 15 metrics tested in this work.

4.2 Evaluation

In this subsection, we report on the results of our experiments, aiming to evaluate the 15 user similarity metrics, primarily in regard to rating prediction accuracy and secondarily in regard to rating prediction coverage. The presentation of the results is organised into two parts, based on the NN selection phase. The first part reports on the results obtained when NNs are selected using the similarity threshold, and the second part reports on the results obtained when NNs are selected using the KNN method.

Within each part, results concerning the use of both functions for the rating prediction value formulation phase (the mean-centred prediction function and the weighted sum function) are presented. For conciseness, only the averaged results over all ten datasets are presented for each setting. The detailed results are available at [76].

Lastly, we aggregate the results and we present and discuss the overall evaluation outcome. The prediction errors observed for the Book-crossing dataset are normalised (due to its increased range, compared to the other 9 datasets), according to the Normalised Mean Absolute Error (NMAE) [77], before computing the datasets' averages.

4.2.1 Evaluation with NN selection using a neighbour similarity threshold

Figure 1 illustrates the average rating prediction MAE observed for the 15 similarity metrics tested in this paper, using the 10 sparse datasets presented in the beginning of the section, when setting the neighbour similarity threshold $THR = 0.0$. The SP metric achieved the lowest average MAE, among all 15 metrics with 0.63 and 0.64, when the mean-centred and the weighted sum prediction functions are used, respectively. At the dataset level, the SP scored the lower MAE in five and three out of 10 datasets, respectively. For the most of the rest of the datasets, it achieved the 2nd best MAE score. Low MAE scores are also achieved by the JACC, the ARI and the ACOS.

In Fig. 2, we can notice that the optimal average RMSE is again accomplished by the SP matric, with a score of 0.98

and 1.05, when the mean-centred and the weighted sum prediction functions are used, respectively, having the lowest RMSE score in five and three, respectively, out of 10 datasets. Very good RMSE results are also achieved by the ACOS, the CPCC, and the JACC.

Considering the F1-measure, the higher scores in both the prediction functions were found to have the ACOS, followed by the NSM and the SP (c.f. Fig. 3). For the mean-centred-based prediction calculation, F1 scores are very similar for all metrics, ranging from 0.831 to 0.836 (the highest value is 0.6% larger than the lowest one). For the weighted sum-based prediction calculation, F1 scores exhibit higher variations ranging from 0.811 to 0.827 (the highest value is 2% larger than the lowest one).

Regarding the rating prediction coverage, the majority of the metrics have satisfactory results, taking into consideration the high sparsity of the datasets, ranging from 31% (for the ARI) to 51% (for the EUCL and the CHEB).

Figures 4, 5 and 6 illustrates the effectiveness of similarity metrics when the THR increases at 0.25. The JACC presented a very low prediction coverage (∼7.5%), which renders it not usable in operational context, while additionally the rating prediction results cannot be considered representative. Under this rationale, JACC is excluded from the presentation of the results and the relevant discussions. All the other 14 metrics achieved satisfactory average coverage rating results, taking into consideration the high sparsity of the datasets, ranging from 26% (for the ARI similarity) to 50% (for the MSD similarity).

Regarding the MAE measure (Fig. 4), the SP metric achieved the lowest MAE, among all 14 metrics (0.62 and 0.63, when the mean-centred and the weighted sum prediction functions are used, respectively), followed by the ARI and the ACOS metrics. At the dataset level, the SP scored the lowest MAE in five and three (out of 10) datasets when the mean-centred and the weighted sum prediction functions are used, respectively.

Regarding the RMSE (Fig. 5), the optimal average RMSE is achieved by the SP with a score of 0.97 (mean-centred) and 1.04 (weighted sum), having the lowest RMSE score in 6 (mean-centred) and 4 (weighted sum), out of 10 datasets. Very good RMSE results are also achieved by the CPCC and the ACOS. Regarding the F1-measure (Fig. 6), the highest scores in both prediction functions were obtained by the ACOS, followed by the EUCL, the CHEB, and the SP. F1 scores exhibit higher variations compared to the case of $THR = 0.0$ falling in the range [0.815, 0.831] for mean-centred prediction computations (the highest value is 2% larger than the lowest one) or [0.825, 0.837] for weighted sum prediction computations (the highest value is 1.5% larger than the lowest one) (Fig. 6).

Lastly, when the THR increased to 0.5, the JACC attained an even lower prediction coverage (∼2.6%) and hence is again excluded from the result presentation and discussion. All the other 14 metrics achieved satisfactory average coverage rating results, again taking into consideration the high sparsity of the datasets, ranging from 21% (for the ARI) to 49% (for the MSD). Regarding the MAE results, as shown in Fig. 7, the SP metric achieved the lowest MAE among all 14 metrics, with 0.59 in both prediction formulas, followed by the MANH, the EUCL, and the CHEB similarity metrics.

The optimal average RMSE (Fig. 8) is achieved by the SP with a score of 0.95 (mean-centred) and 1.01 (weighted sum), followed by the CPCC with a score of 0.99 (mean-centred) and 1.05 (weighted sum). Regarding the F1-measure (Fig. 9), the highest score in both the prediction functions was achieved by the NSM, followed by the ACOS, the ARI, and the SP metrics. F1 scores fall in the range [0.829, 0.842] for mean-centred prediction formulation (the highest value is 1.3% larger than the lowest one) and [0.811, 0.833] for weighted sum prediction formulation (the highest value is 2.7% larger than the lowest one).

Overall, when the neighbour selection is based on the similarity threshold, the SP and the ACOS similarity metrics achieve the highest rating prediction accuracy results. For the detailed results, the interested reader is referred to the technical report [76].

4.2.2 Evaluation under the KNNs NN selection scheme

Figure 10 illustrates the average rating prediction MAE observed for the 15 similarity metrics considered in this paper, using the 10 sparse datasets presented in the beginning of Sect. 4. We can observe that the SP and the JACC metrics achieve the lowest MAE among all 15 metrics. The optimal average RMSE (Fig. 11) is achieved by the JACC, with 1.00 and 1.08, when the mean-centred and the weighted sum prediction functions are used, respectively. Satisfactory results are also achieved by the SP and the MSD similarities.

Regarding the F1-measure (Fig. 12), the highest score in both prediction functions is attained by the NSM, followed by the COS metric. F1 scores fall in the range [0.827, 0.836] for mean-centred prediction formulation (the highest value is 1% larger than the lowest one), and [0.809, 0.826] for weighted sum prediction formulation (the highest value is 2.1% larger than the lowest one).

Regarding the rating prediction coverage, the majority of the metrics have satisfactory results, taking into consideration the high sparsity of the datasets, ranging from 28% (for the ARI) to 44% (for the JACC).

When K increases to 500, the rating prediction coverage is slightly improved, as expected, ranging from 29% (for the ARI) to 45% (for the JACC and the MSD). The increase of K has no significant impact on the accuracy metrics, with the average MAE across all metrics and datasets increasing by

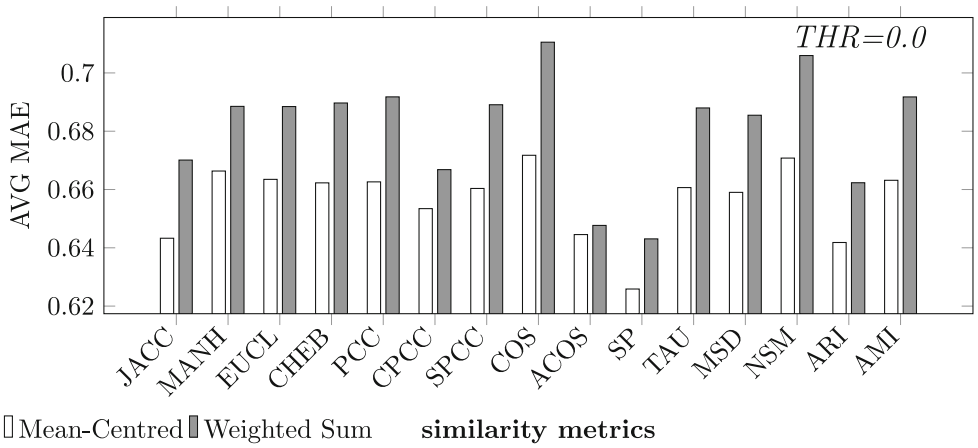

Fig. 1 Average MAE achieved by the 15 metrics, when NNs are selected using a similarity threshold and under neighbour THR=0.0

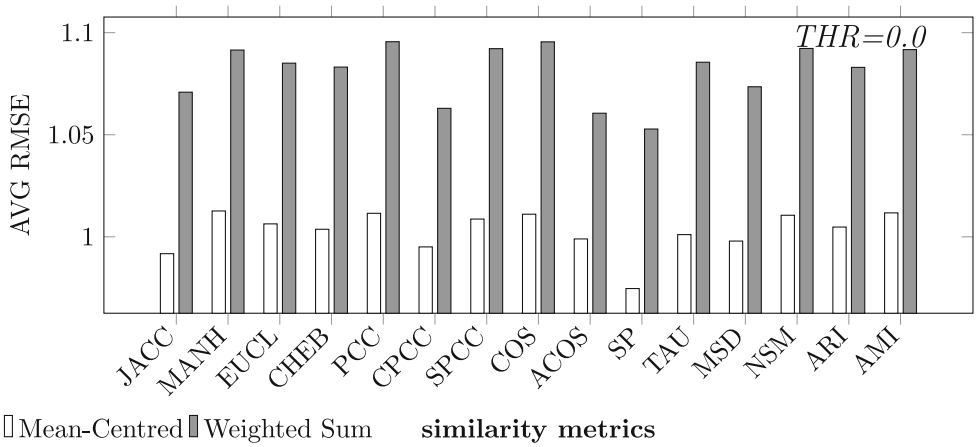

Fig. 2 Average RMSE achieved by the 15 metrics, when NNs are selected using a similarity threshold and under neighbour THR=0.0

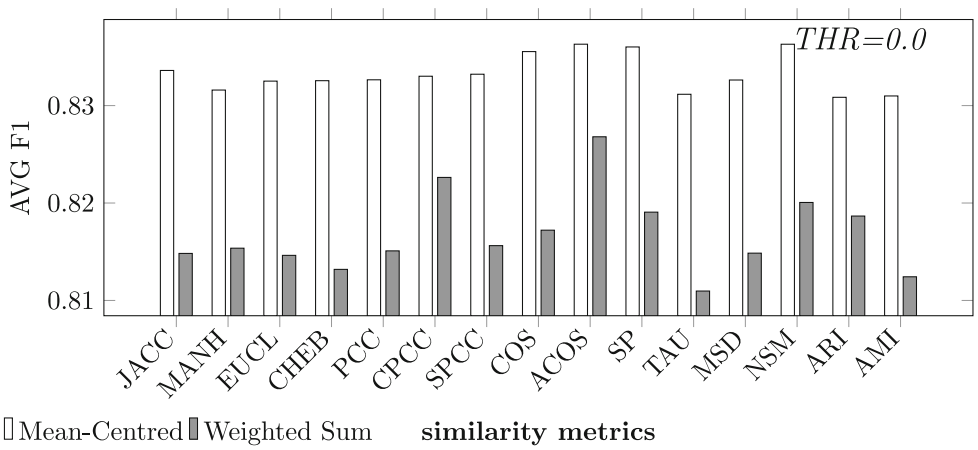

Fig. 3 Average F1 achieved by the 15 metrics, when NNs are selected using a similarity threshold and under neighbour THR=0.0

0.48% (Fig. 13) and the average RMSE increasing by 0.94% (Fig. 14).

Regarding the MAE, we can observe (Fig. 13) that both the SP and the JACC metrics achieve the lowest MAEs, among all 15 metrics.

Considering the RMSE (Fig. 14), the optimal results are achieved by the JACC, followed by the MSD metric in both prediction formulas used. Regarding the F1-measure (Fig. 15), the highest score is achieved by the NSM, followed by the COS, ACOS, CPSS and the MSD metrics. For the detailed results, the interested reader is referred to the technical report [76].

4.2.3 Aggregated results

Based on the experimental output, presented in the previous two subsections, Table 6 depicts the aggregated results of the rating prediction evaluation. More specifically, for each of the

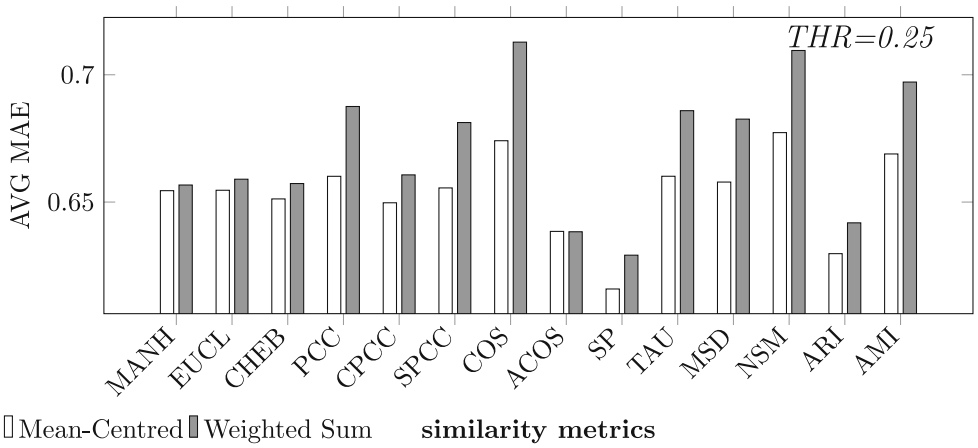

Fig. 4 Average MAE achieved by the 14 metrics (JACC is excluded), when NNs are selected using a similarity threshold and under neighbour $THR = 0.25$

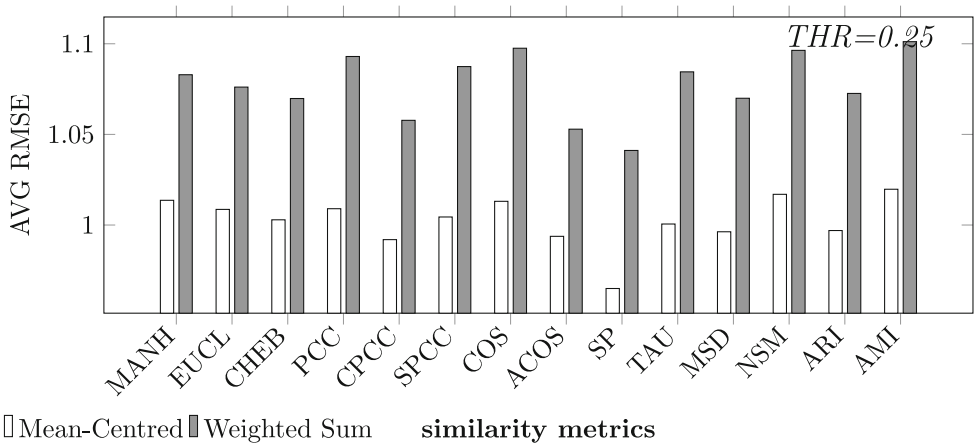

Fig. 5 Average RMSE achieved by the 14 metrics (JACC is excluded), when NNs are selected using a similarity threshold and under neighbour $THR = 0.25$

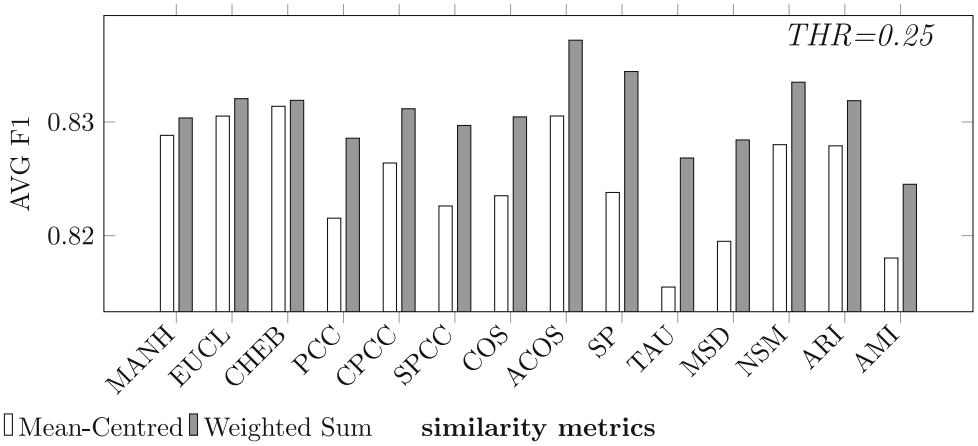

Fig. 6 Average F1 achieved by the 14 metrics (JACC is excluded), when NNs are selected using a similarity threshold and under neighbour $THR = 0.25$

similarity metrics evaluated in this work (corresponding to rows of Table 6) and each of the neighbour selection settings used (five primary columns), this table depicts the metric that achieved the optimal results, for the three prediction error metrics used (secondary columns—'M' for the MAE, 'R' for the RMSE, and 'F' for the F1-measure metrics), when the mean-centred prediction formula is utilised to formulate the rating prediction numeric value. The similarity metric that achieved the highest performance is marked with a double-plus sign "++", while the runner-ups whose performance is "very close" to the highest performance are marked with a single-plus sign "+".[1]

[1] [1]The MAE and the RMSE of a runner-up algorithm for an experiment are deemed to be "very close" to the highest performance, if they are found to be at most 5% larger than the corresponding lowest MAE and RMSE value observed in the specific experiment. For the F1-measure, the margin is set to 1%, considering that the F1-measure scores observed in the experiments vary by small margins, up to 2.7%. Depending on

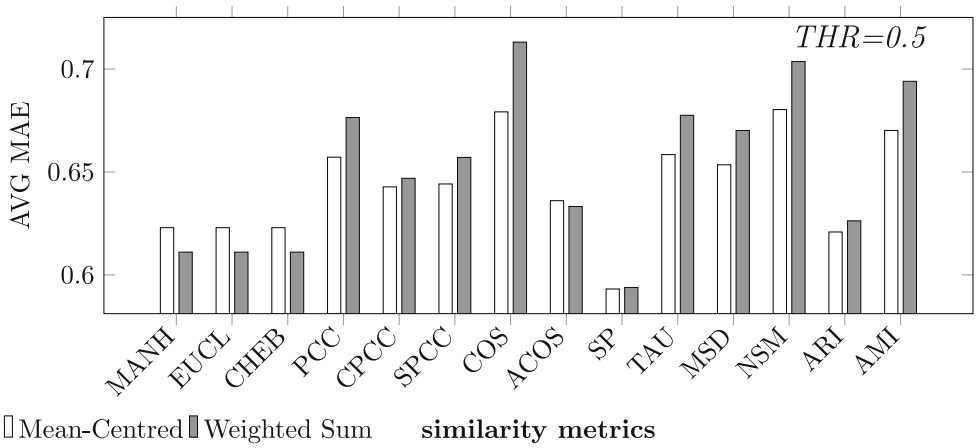

Fig. 7 Average MAE achieved by the 14 metrics (JACC is excluded), when NNs are selected using a similarity threshold and under neighbour $THR = 0.5$

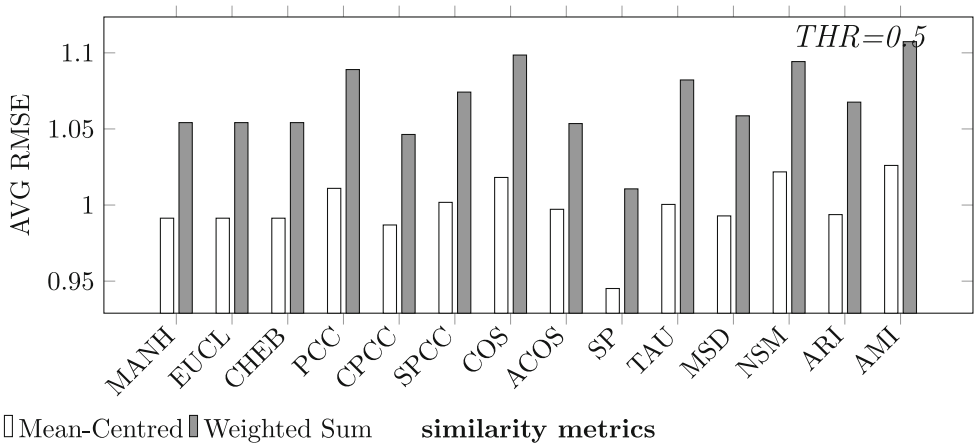

Fig. 8 Average RMSE achieved by the 14 metrics (JACC is excluded), when NNs are selected using a similarity threshold and under neighbour $THR = 0.5$

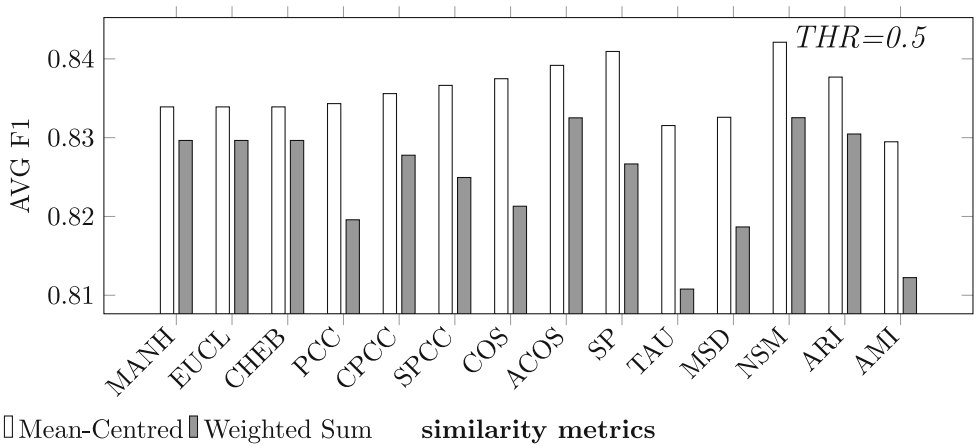

Fig. 9 Average F1 achieved by the 14 metrics (JACC is excluded), when NNs are selected using a similarity threshold and under neighbour $THR = 0.5$

In Table 6, we can notice that when the mean-centred prediction formula is used for prediction value computation, the SP similarity metric achieves either the highest performance or a performance "very close" to the highest one, in every setting. Furthermore, satisfactory results are achieved by the ARI and the CHEB. When either the KNN method is used for the neighbour selection or the similarity threshold with low THR value set, the JACC achieves very good results, as well.

Table 7 depicts the respective aggregated results, when the weighted sum function is used for the prediction value formulation. In Table 7, we can observe that when the weighted sum function is used for prediction value computation, the SP similarity metric achieves either the highest performance or a performance "very close" to the highest one, in every

the specific use case, other thresholds can be adopted for classifying runner-up performance as "very close".

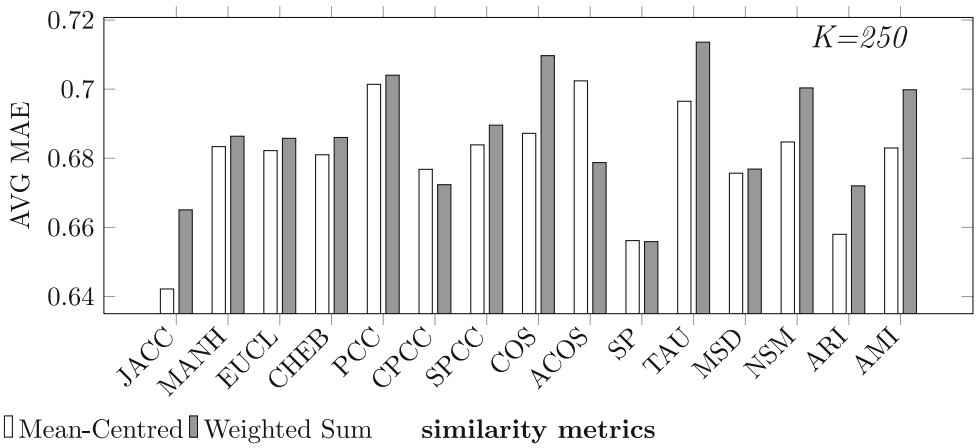

Fig. 10 Average MAE achieved by the 15 metrics, when the top-K NNs are selected, using K=250

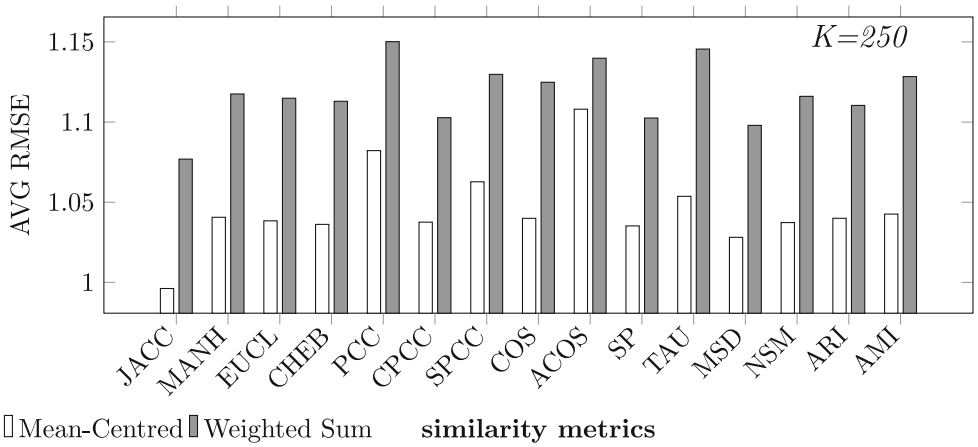

Fig. 11 Average RMSE achieved by the 15 metrics, when the top-K NNs are selected, using K=250

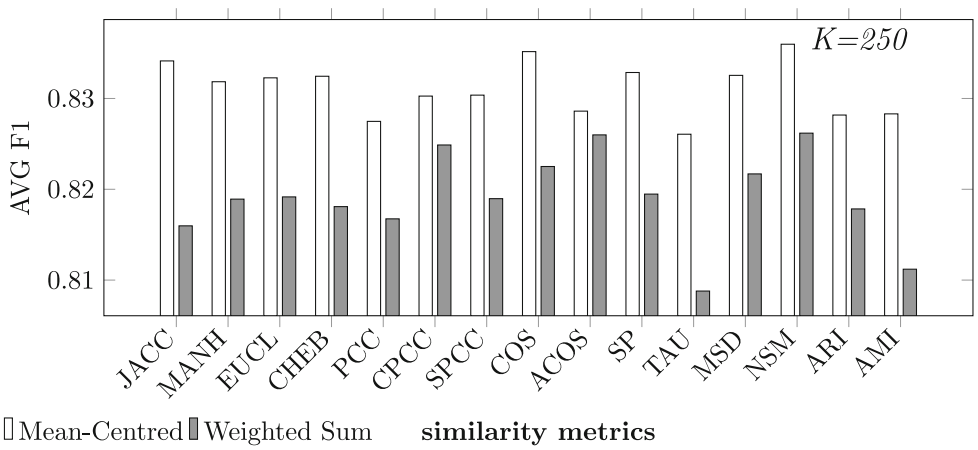

Fig. 12 Average F1 achieved by the 15 metrics, when the top-K NNs are selected, using K=250

setting. Moreover, satisfactory results are achieved by the CPCC and the ACOS metrics.

To further confirm the validity of the findings, we performed ANOVA tests on the results of different experiments. A distinct ANOVA test was conducted for each experiment configuration and result (method for NN selection and its related parameters, formulation of predictions, evaluation metric). All ANOVA tests designated that the obtained results are statistically significant at the level of 0.05. For conciseness purposes, Table 8 depicts selected ANOVA results; all ANOVA p-values have been found to be in the range [0.0087,0.0437].

The success of the SP metric is attributed to the following characteristics:

- the SP metric exhibits lower sensitivity to outliers [78], as compared to other metrics, such as PCC and COS. This is particularly important for sparse datasets, where the

Table 6 Aggregated results when the mean-centred prediction formula is used

mean-centred	K=250			K=500			THR=0.0			THR=0.25			THR=0.5		
	M	R	F	M	R	F	M	R	F	M	R	F	M	R	F
JACC	++	++	+	++	++	+	+	+	+						
MANH		+	+		+	+		+	+			+	+	+	+
EUCL		+	+		+	+		+	+		+	+	+	+	+
CHEB	+	+	+		+	+		+	+		+	++	+	+	+
PCC								+	+		+				+
CPCC		+	+	+		+	+	+	+		+	+		+	+
SPCC			+			+		+	+		+				+
COS		+	+		+	+		+	+		+	+			+
ACOS						+	+	+	++	+	+	+			+
SP	+	+	+	+	+	+	++	++	+	++	++	+	++	++	+
TAU								+	+		+				
MSD		+	+	+	+	+		+	+		+				
NSM		+	++		+	++		+	+			+			++
ARI	+	+	+	+		+	+	+	+	+	+	+	+		+

Table 7 Aggregated results when the weighted sum prediction function is used

weighted sum	K=250			K=500			THR=0.0			THR=0.25			THR=0.5		
	M	R	F	M	R	F	M	R	F	M	R	F	M	R	F
JACC	+	++		+	++		+	+							
MANH	+		+		+	+		+		+	+	+	+	+	+
EUCL	+		+		+	+		+		+	+	+	+	+	+
CHEB	+	+	+		+			+		+	+	+	+	+	+
PCC						+		+			+				
CPCC	+	+	+	+	+	+	+	+	+		+	+		+	+
SPCC		+	+			+		+			+	+			+
COS			+		+	+		+				+			+
ACOS	+		+	+		+	+	+	++	+	+	++		+	+
SP	++	+	+	++	+	+	++	++	+	++	++	+	++	++	+
TAU								+	+		+				
MSD	+	+	+	+	+	+		+			+			+	
NSM			++		+	++		+	+			+			++
ARI	+	+		+	+		+	+	+	+	+	+			+
AMI					+			+							

Table 8 ANOVA results for different experiments

Experiment	Metric	ANOVA p-value	Notes
Top-K ($K = 250$)/weighted sum	MAE	0.0356	Statistically significant at the level of 0.05
Top-K ($K = 250$)/weighted sum	RMSE	0.0092	Statistically significant at the level of 0.01
Top-K ($K = 250$)/weighted sum	F1	0.0406	Statistically significant at the level of 0.05
Threshold ($THR = 0.5$)/weighted sum	MAE	0.0358	Statistically significant at the level of 0.05
Threshold ($THR = 0.5$)/weighted sum	RMSE	0.0106	Statistically significant at the level of 0.05
Threshold ($THR = 0.5$)/weighted sum	F1	0.0363	Statistically significant at the level of 0.05

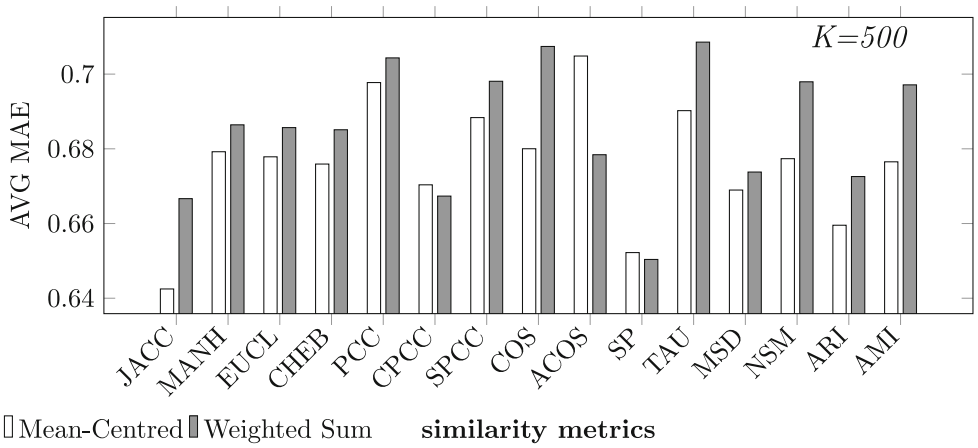

Fig. 13 Average MAE achieved by the 15 metrics, when the top-K NNs are selected, using K=500

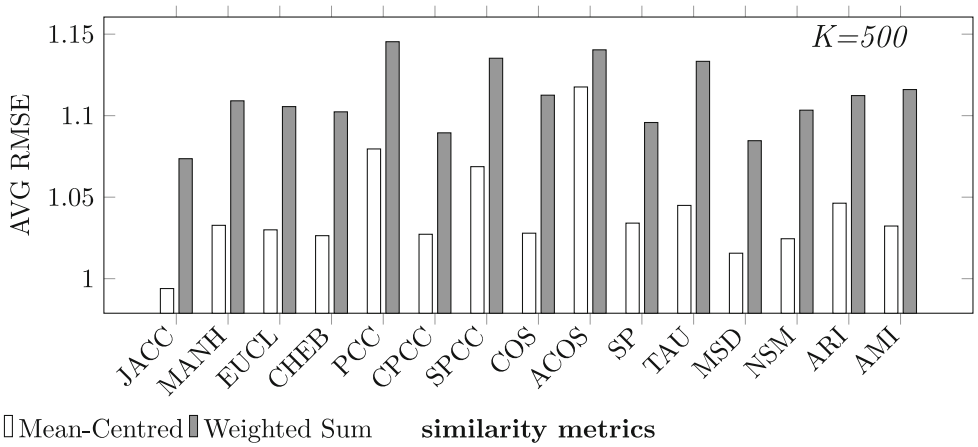

Fig. 14 Average RMSE achieved by the 15 metrics, when the top-K NNs are selected, using K=500

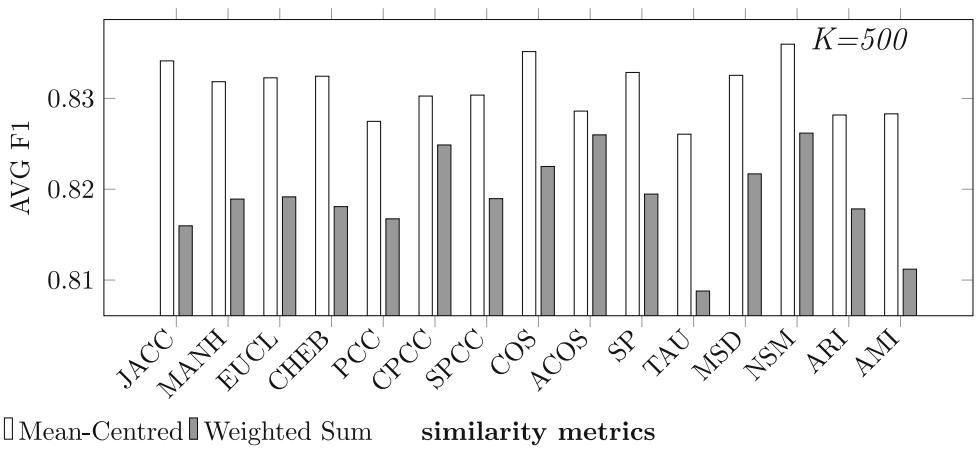

Fig. 15 Average F1 achieved by the 15 metrics, when the top-K NNs are selected, using K=500

number of co-rated items is small, and therefore, outliers will have high impact.

- the SP metric uses rankings instead of rating values, and therefore is not affected by the scaling of ratings. The ratings entered by different users may exhibit varying scaling, either manifested as strictness/leniency or as the use of a portion of the rating scale vs. the use of the full rating scale.
- the SP metric is able to better capture non-linear relationships between data [79], and common user ratings in sparse datasets may be nonlinear. On the contrary, other metrics, such as PCC and COS, are capable of capturing only linear relationships.

4.2.4 Analysis of metric behaviour in relation to sparsity

In this subsection, we examine the behaviour of metrics in relation to dataset sparsity. Since different datasets have been found to exhibit divergent results, in order to warrant independence from the individual dataset characteristics, we do not directly compare the results of different datasets. Instead, we have subsampled datasets at subsets with size equal to 10%, 30%, 50%, 70%, and 90% of the full dataset and tested the performance of the metrics in these subsets.

Figure 16 depicts the MAE improvement achieved by the 15 metrics, at different subsampling ratios of the Yahoo!Movies dataset, when the top-K NNs are selected, using K=250. The base against which the improvement is computed is the MAE achieved by each metric when the 10% of the Yahoo!Movies dataset is used. We can observe that the general trend is that the availability of more data leads to MAE improvement (i.e. reduction). There do exist some cases for which when the size of the dataset increases, the MAE increases too (e.g. the cases of the CPCC and the NSM metrics when the percentage of data increases from 30 to 50%). This is attributed to the fact that the increase in the data sampling ratio leads to the inclusion of previously excluded users in the dataset. These users may have very low numbers of ratings present in the current extent of the dataset, and therefore, the predictions formulated for them are bound to exhibit high error magnitudes, leading to the deterioration of the overall MAE.

Correspondingly, Figures 17 and 18 illustrate the improvement in the RMSE and the F1 metrics under the same experiment. We can observe that the RMSE metric follows the same improvement pattern. This is expected since both metrics express qualitative properties of the prediction error magnitudes. Thus, both are computed against the same value sets. Regarding the F1-measure, this is also found to improve along with the increase in the subsampling ratio of the dataset. Small fluctuations are observed. These fluctuations are attributed to the same root cause inducing the anomalies for the MAE and RMSE metrics, i.e. the inclusion in the dataset of previously excluded users, the predictions for whom exhibit high errors.

When conducting the same experiments using the threshold method for NN selection is employed, with parameter THR set to 0.5, the same behaviour is observed, i.e. decreased sparsity leads to improvements in the MAE, RMSE, and F1 metrics. Since the JACC metric produces very few ratings for subsampling ratios less than 70%, conclusions can only be drawn for its behaviour for the experiments using subsumpling ratios 70%, 90%, and 100%. In this range, the JACC metric demonstrates a sharp increase in its performance when sparsity is reduced; however, this is due to the fact that exhibits very low performance (MAE, RMSE and F1) at subsampling rate equal to 70%, and therefore, it has substantial improvement margin. The detailed results for this experiment are included in the technical report [76].

To further validate that the conclusions drawn from observing the performance of different metrics when the YahooMovies dataset was subsampled at different ratios, we repeated the same experiment using the Ciao dataset. The results obtained from this experiment align closely with the findings presented above for the YahooMovies dataset, with higher subsampling rates (and henceforth decreased sparsity) leading to improvements in the MAE, RMSE, and F1 metrics. Again, some fluctuations are present, because the increase in the sampling rate leads to the inclusion of new users with low number of ratings, the predictions for whom have high error margins. For the detailed results, the interested reader is referred to the technical report [76].

4.2.5 Implicit feedback

The experiments presented above consider datasets with explicit feedback, i.e. datasets where users have explicitly rated items. In many cases, however, explicit feedback is unavailable, and RecSys resort to the use of *implicit feedback*, i.e. exploiting the interaction of users with items to infer the users' interest towards items. The inferred preference is subsequently used to identify similar users or items, and estimate user preferences towards items that users have not interacted with. Finally, items with the highest preference estimation are recommended to the user. Notably, implicit feedback can be also employed to build an RecSys aiming to specific qualitative properties; for instance, [80] reports that RecSys based on implicit feedback promote user engagement, while explicit rating-based RecSys promote accuracy.

In this context, user or item similarities may need to be computed; therefore, the issues of metric performance and selection are relevant for this class of RecSys as well. In this context, specialised similarity measurements have been developed [81, 82] or machine learning-based methods are used, where embeddings are constructed and these are subsequently fed into neural networks that formulate recommendations [83, 84]. The Bayesian Personalised Ranking (BPR) [85, 86] is a prominent approach in this category, which is gaining acceptance. Since the specialised similarity metrics are domain-specific and are typically developed for individual cases, whereas machine learning-based models learn similarities through the neural networks, without the explicit use of a mathematical similarity function, this work will not consider these cases.

An alternative approach is to exploit traits of user interest against items to produce estimations of ratings, converting thus implicit ratings to explicit [87, 88] and create a transformed dataset. Once (estimated) explicit ratings are available, the transformed datasets can be processed by CF-based algorithms to generate predictions and formulate

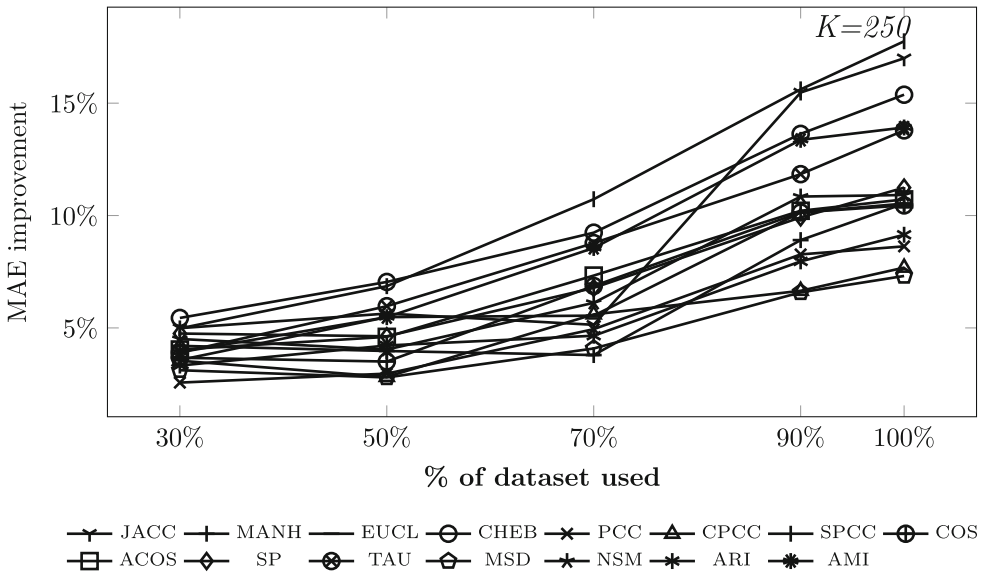

Fig. 16 MAE improvement achieved by the 15 metrics, at different subsampling ratios of the Yahoo!Movies dataset, when the top-K NNs are selected, using $K = 250$ (Base: metric performance for subsampling ratio=10%)

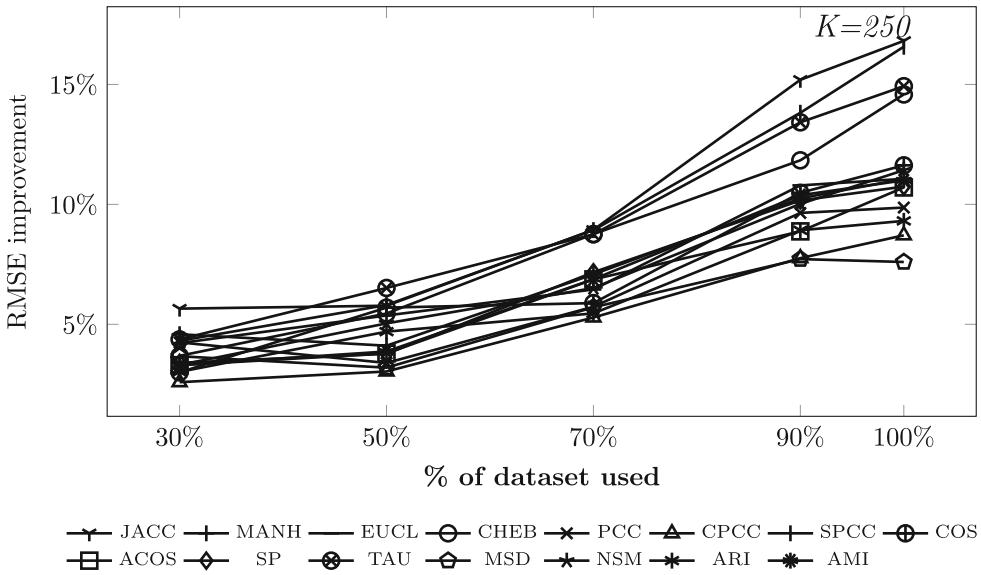

Fig. 17 RMSE improvement achieved by the 15 metrics, at different subsampling ratios of the Yahoo!Movies dataset, when the top-K NNs are selected, using $K = 250$ (Base: metric performance for subsampling ratio=10%)

recommendations. In the remainder of this subsection, we present an experiment aiming to provide insight to the capabilities of different metrics to be used in the phase of rating prediction within this process. We note here that since numerous approaches can be used for transforming implicit ratings to explicit, which entail the use of different parameters and are dependent on the nature of the implicit data, the topic of the performance of similarity metrics on rating estimations produced on the basis of implicit feedback cannot be exhaustively covered in this paper. The goal of this section is to provide some insight in this topic, while a more comprehensive analysis will be considered in our future work.

In the context of our experiment, we use the Last.FM dataset [89], which contains 166154 listening records, with each record indicating when a particular user listened to a specific song. Initially, the distribution of number of listenings was analysed, and the results are illustrated in Figure 19. Based on this distribution, the mapping between number of listenings and estimated ratings illustrated in Table 9 was adopted for the transformation of the original dataset to a dataset using explicit ratings. While alternative mappings can be applied, this mapping was used because it (a) uses the complete range of the rating scale and (b) provides the best possible balance between different estimated values of ratings.

After the dataset was converted to explicit ratings, it contained 125764 records with a sparsity of 84.41%. This dataset was of high density, and therefore, it was subsampled with a factor of 1%, producing a dataset with sparsity equal to 99.84%, which is classified as "medium sparsity". Finally, the experimental procedure described in introduction of Sect. 4 was applied.

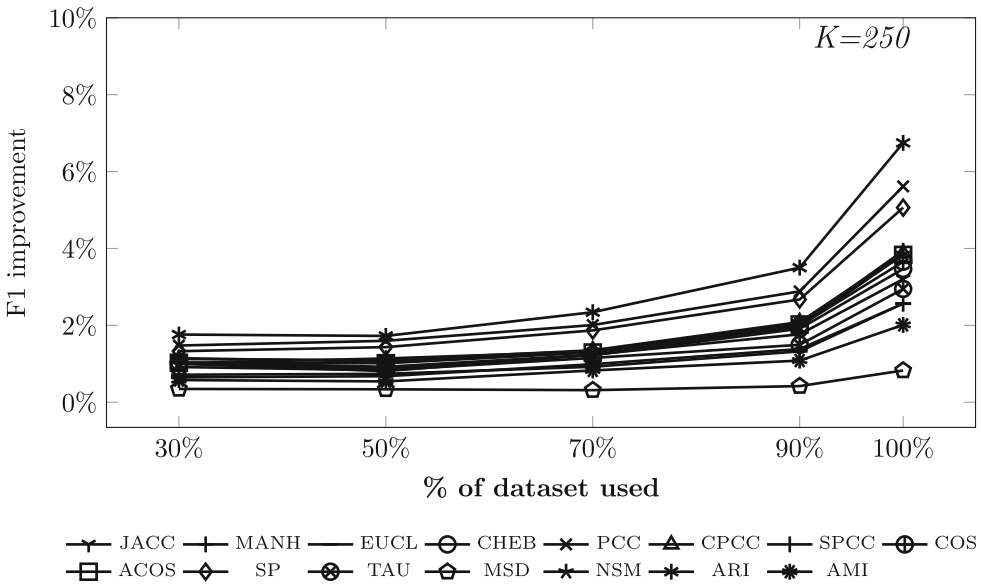

Fig. 18 F1 improvement achieved by the 15 metrics, at different subsampling ratios of the Yahoo!Movies dataset, when the top-K NNs are selected, using $K = 250$ (Base: metric performance for subsampling ratio=10%)

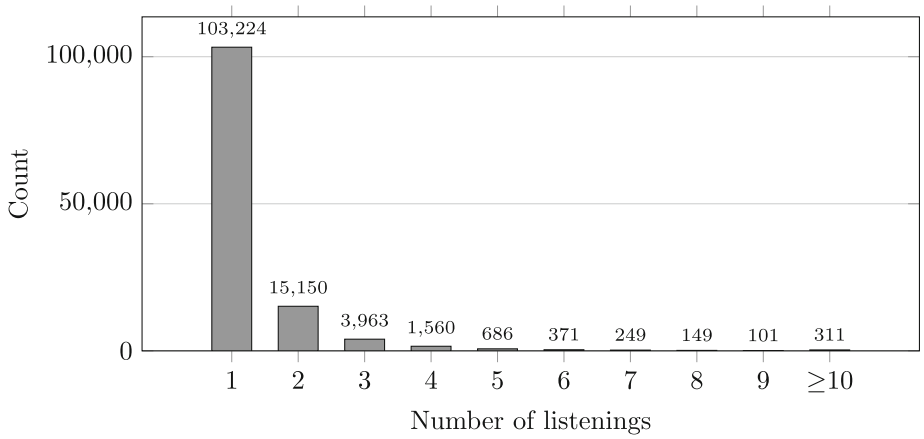

Fig. 19 Distribution of number of listenings for the Last.FM dataset

Table 9 Mapping between number of listenings and estimated ratings for the Last.FM dataset

Number of listenings	Estimated rating
1	1
2	2
3	3
4–5	4
≥ 6	5

In the results obtained, it was observed that the ACOS and MANH measures produced very low coverages (less that 10%) under all configurations and are therefore excluded from the presentations of the results, under the rationale that these low coverages render them inapplicable for practical use, regardless of the accuracy of their predictions. The same holds for the JACC, EUCL, and CHEB metrics, under the threshold-based NN selection with THR=0.5.

Figures 20, 21 and 22 illustrate the results of the experiments for with the subsampled Last.FM dataset when the Top-K approach was employed for NN selection, with K=250. Note that since the subsampled dataset contained less than 250 users, increasing the K parameter in the Top-K approach for NN selection would not have any effect. In the results, we can observe that the JACC, TAU, and ARI metric achieve the lowest MAE, while the SP, TAU, and ARI metrics attain the lowest RMSE. Regarding the F1 metric, SP exhibits the highest performance, followed by AMI and JACC.

Under the threshold-based approach, when THR is set to 0, the SP, TAU, and ARI metrics achieve the lowest MAE and the lowest RMSE too, while SP, AMI ARI, and JACC attain the highest F1 value. Finally, when the THR parameter is set to 0.5, SP, TAU and ARI again achieve the lowest MAE and the lowest RMSE, while the highest F1 value is attained by SPCC, SP, and PCC. These results are not presented in detail

for conciseness. The interested reader may retrieve them from the technical report [76].

These results indicate that again the SP metric exhibits optimal or close-to-optimal performance when explicit ratings are estimated on the basis of implicit feedback; however, as stated above, further investigation is needed to reach decisive conclusions. This research is considered as part of our future work.

4.3 Result discussion

Based on the output of the experimental evaluation presented in the preceding subsection, the following conclusions can be derived regarding the three research questions set in Sect. 1.1:

- *RQ1*: Which user similarity metric seem to yield the best results when the CF algorithm is applied to sparse CF datasets?
  Answer: Based on our set of experiments, the SP similarity metric was proved to achieve the best results when sparse CF datasets are used. Satisfactory results are also achieved by the ARI, the CHEB and the CPCC.
- *RQ2*: Do other parameters of the CF algorithm (e.g. neighbour selection approach, rating prediction formula, density of the dataset) affect the choice of the user similarity methods who yield the best results?
  Answer: Based on our set of experiments, the SP similarity metric achieves either the highest performance or a performance "very close" to the highest one, in every setting. Hence, the results of this metric seem to be independent of other parameters of the CF algorithm.
- *RQ3*: Are the PCC and the COS, among the best similarity metrics in sparse CF datasets, as expected, since they are used in most of the CF research works?
  Answer: Interestingly, based on the experiments presented in the preceding subsection, neither the PCC nor the COS are among the metrics that achieve satisfactory rating prediction results when the CF algorithm is applied to sparse CF datasets. More specifically, the COS metric is ranked near the middle of the evaluated metrics, while the PCC is ranked much lower.

4.4 Research implications

The work presented in this paper has a number of theoretical and practical implications. Considering the theoretical implications, to the best of the authors' knowledge, there is no previous work that assesses the effectiveness of user similarity metrics in sparse CF datasets. For the results of this work to be reliable, multiple CF prediction parameters are taken into account, considering NN selection techniques, rating prediction computation formulas, and rating prediction accuracy measures. Lastly, to ensure that the evaluation is not biased by the item domain, the experiments include 10 datasets (the baseline comparison is on the same 10 datasets for all 15 metrics) that span across several product fields, from Music and Movies, to Books and Videogames.

It is worth noting that due to the fact that this work followed the approach to warrant the reliability of the results and generalisability of the conclusions, i.e. the consideration of multiple CF prediction parameters settings and multiple NN selection techniques, rating prediction computation formulas and rating prediction accuracy measures, the results listed in this paper provide insight on the effect that each of these settings and parameters have on the CF rating prediction process.

In terms of practical implications, the results of this evaluation review point out the user similarity metrics that yield the best results when the CF algorithm is applied to sparse datasets. As a result, future CF research works can make use of this output and select to incorporate certain similarity metrics when applying a CF algorithm in sparse CF datasets, to accomplish better rating prediction results, which will lead to improved recommendation quality. Based on the experimental output presented in Sect. 4.2, this paper proposes the inclusion/use of primarily the SP similarity metric and secondarily the ARI, the CHEB, and the CPCC similarity metrics, when applying CF algorithms to sparse datasets, in order to accomplish better rating prediction results, which will lead to improved recommendation quality. In the industry domain, including e-commerce sites and streaming platforms, administrators can exploit the findings of this study and tune their systems to use the most prominent similarity metric, achieving thus more successful recommendations and consequently increasing user satisfaction [44, 45].

5 Conclusions and future work

The work presented in this paper is an extensive evaluation review of 15 widely used user similarity metrics in sparse CF datasets. The evaluation included (i) two neighbour selection approaches, namely the similarity threshold approach and the top-k approach, and (ii) two rating prediction formulas, the mean-centred formula, and the weighted sum method, (iii) 10 sparse CF datasets, from five different providers, having sparsities ranging from 99.76% to 99.997%, and (iv) three rating prediction accuracy metrics, the F1-measure, the RMSE, and the MAE, all three widely used in CF RecSys research.

The evaluation results showed that the metrics that achieved the higher prediction scores (the highest F1-measure and lowest RMSE and MAE) were found to be, primarily, the Spearman rank correlation, followed by the Adjusted Rand Index, the Constrained PCC, and the Chebysev distance. Therefore, based on the evaluation output presented in Sect. 4, this paper proposes the inclusion of

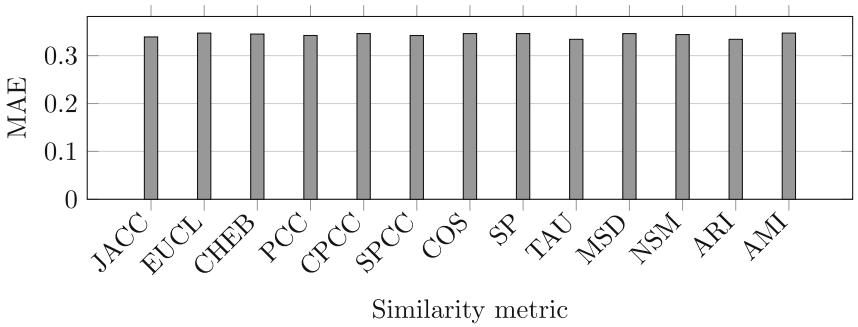

Fig. 20 MAE for the similarity metrics, while using the subsampled Last.FM dataset and employing the Top-K method for NN selection with K=250

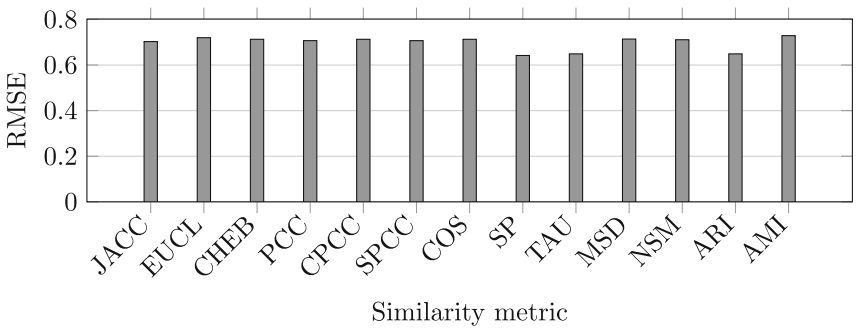

Fig. 21 RMSE for the similarity metrics, while using the subsampled Last.FM dataset and employing the Top-K method for NN selection with K=250

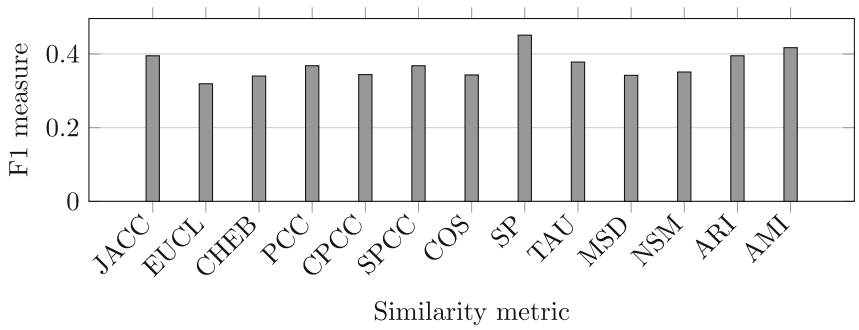

Fig. 22 F1-measure for the similarity metrics, while using the subsampled Last.FM dataset and employing the Top-K method for NN selection with K=250

at least one of the aforementioned four similarity metrics, when applying CF algorithms to sparse datasets.

Regarding future work, we plan to conduct research on the effect of the number of commonly evaluated items threshold for NN selection in sparse CF datasets, as discussed in Sect. 4.1. Furthermore, we plan to evaluate CF user similarity metrics with dense datasets (such as the MovieLens datasets), as well as domain-specific datasets, for broader applicability and increased generalisability.

Moreover, we plan to broaden the user similarity metrics, by evaluating newer ones introduced in recent works, and more specifically hybrid similarity metrics. Towards this direction, and considering that the SP metric has demonstrated high performance, the integration of the SP similarity metric with hybrid similarity metrics [23, 90] including the use of the SP user vicinity as a feature in machine learning-based recommendation models [91–93] will be analysed.

Deep learning-based approaches for computing user similarity [93, 94] have emerged as a promising development in the RecSys domain, and are currently gaining acceptance. The investigation of the effectiveness of these approaches and their comparison with traditional and hybrid similarity metrics will be also considered in the context of our future work. The effect of adopting alternative techniques for the identification of top-K neighbours, such as the Lower-Left Partial AUC [95], will also be studied.

Lastly, we plan to investigate the potential impact of backbone models, such as Matrix Factorisation (MF) or LightGCN, on the performance of the SP similarity metric, considering the growing use of neural CF approaches [96, 97].

Author Contributions K.S. and D.M. contributed in conceptualisation, methodology, software development, validation, investigation, and data curation. D.S. and C.V. contributed in conceptualisation, methodology, validation, investigation, and data curation. All authors reviewed the manuscript.

Funding Open access funding provided by HEAL-Link Greece.

Data Availability Open access datasets were used. The URLs of these datasets are given as footers of Table 3.

Declarations

Conflict of interest/Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Ethics Approval and Consent to Participate Not applicable.

Consent for Publication All authors consent to the publication of this manuscript.

References

1. Varlamis, I., Sardianos, C., Chronis, C., Dimitrakopoulos, G., Himeur, Y., Alsalemi, A., Bensaali, F., Amira, A.: Using big data and federated learning for generating energy efficiency recommendations. Int. J. Data Sci. Anal. **16**(3), 353–369 (2023). https://doi.org/10.1007/s41060-022-00331-2
2. Kobusinska, A., Boron, M., Kerebinska, A., Margaris, D.: Exploiting recommender service to enhance efficiency of replication. In: 2019 IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA), pp. 64–72 (2019). https://doi.org/10.1109/SOCA.2019.00017
3. Kobusinska, A., Margaris, D., Peikert, J.: Trust- and rating- based recommendations for on-line social networks. In: 2021 International Conference on Computer Communications and Networks (ICCCN), pp. 1–11 (2021). https://doi.org/10.1109/ICCCN52240.2021.9522182
4. Yilma, B.A., Naudet, Y., Panetto, H.: Personalisation in cyber-physical-social systems: A multi-stakeholder aware recommendation and guidance. In: Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization, pp. 251–255 (2021). https://doi.org/10.1145/3450613.3456847
5. Kamel, M.M., Gil-Solla, A., Guerrero-Vsquez, L.F., Blanco-Fernndez, Y., Pazos-Arias, J.J., Lpez-Nores, M.: A crowdsourcing recommendation model for image annotations in cultural heritage platforms. Appl. Sci. **13**(19), 10623 (2023). https://doi.org/10.3390/app131910623
6. Yilma, B.A., Naudet, Y., Panetto, H.: Personalisation in cyber-physical-social systems: A multi-stakeholder aware recommendation and guidance. In: Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization, pp. 251–255 (2021). https://doi.org/10.1145/3450613.345684
7. Kapitsaki, G.M., Charalambous, G.: Modeling and recommending open source licenses with findosslicense. IEEE Trans. Software Eng. **47**(5), 919–935 (2021). https://doi.org/10.1109/TSE.2019.2909021
8. Kermany, N.R., Zhao, W., Batsuuri, T., Yang, J., Wu, J.: Incorporating user rating credibility in recommender systems. Futur. Gener. Comput. Syst. **147**, 30–43 (2023). https://doi.org/10.1016/j.future.2023.04.029
9. Papadakis, H., Papagrigoriou, A., Panagiotakis, C., Kosmas, E., Fragopoulou, P.: Collaborative filtering recommender systems taxonomy. Knowl. Inf. Syst. **64**(1), 35–74 (2022). https://doi.org/10.1007/s10115-021-01628-7
10. Wang, S., Wang, Y., Sivrikaya, F., Albayrak, S., Anelli, V.W.: Data science for next-generation recommender systems. Int. J. Data Sci. Anal. **16**(2), 135–145 (2023)
11. Alshareet, O., Awasthi, A.: Enhancing e-commerce recommendations with a novel scale-aware spectral graph wavelets framework. Int. J. Data Sci. Anal. pp. 1–14 (2023)
12. Zaman, N., Jana, A.: Automated recommendation model using ordinal probit regression factorization machines. Int. J. Data Sci. Anal. pp. 1–15 (2024)
13. Park, S.H., Kim, K.: Collaborative filtering recommendation system based on improved jaccard similarity. J. Ambient. Intell. Humaniz. Comput. **14**(8), 11319–11336 (2023). https://doi.org/10.1007/s12652-023-04647-0
14. Sardianos, C., Papadatos, G.B., Varlamis, I.: Optimizing parallel collaborative filtering approaches for improving recommendation systems performance. Information **10**(5), 155 (2019). https://doi.org/10.3390/info10050155
15. Kelen, D.M., Benczúr, A.A.: A probabilistic perspective on nearest neighbor for implicit recommendation. Int. J. Data Sci. Anal. **16**(2), 217–235 (2023)
16. Xu, J., Xia, Z., Li, Y., Zeng, Y., Liu, Z.: Subgraph sampling for inductive sparse cloud services qos prediction. In: 2022 IEEE 28th International Conference on Parallel and Distributed Systems (ICPADS), pp. 745–753 (2023). https://doi.org/10.1109/ICPADS56603.2022.00102
17. Koohi, H., Kiani, K.: Two new collaborative filtering approaches to solve the sparsity problem. Clust. Comput. **24**(2), 753–765 (2021). https://doi.org/10.1007/s10586-020-03155-6
18. Gupta, S., Deep, K.: A novel hybrid sine cosine algorithm for global optimization and its application to train multilayer perceptrons. Appl. Intell. **50**(4), 993–1026 (2020)
19. Margaris, D., Vassilakis, C., Spiliotopoulos, D.: What makes a review a reliable rating in recommender systems? Inf. Process. Manage. **57**(6), 102304 (2020). https://doi.org/10.1016/j.ipm.2020.102304
20. Yang, W., Guo, S.H., Zhang, C.J.: A novel rating style mining method to improve collaborative filtering algorithm. In: Journal of Physics: Conference Series, vol. 1187, p. 052101 (2019). https://doi.org/10.1088/1742-6596/1187/5/052101
21. Veras De Sena Rosa, R.E., Guimaraes, F.A.S., Mendonca, R.d.S., Lucena, V.F.d.: Improving prediction accuracy in neighborhood-based collaborative filtering by using local similarity. IEEE Access **8**, 142795–142809 (2020) https://doi.org/10.1109/ACCESS.2020.3013733
22. Alhijawi, B., Al-Naymat, G., Obeid, N., Awajan, A.: Novel predictive model to improve the accuracy of collaborative filtering recommender systems. Inf. Syst. **96**, 101670 (2021)
23. Aljunid, M.F., Huchaiah, M.D.: An efficient hybrid recommendation model based on collaborative filtering recommender systems. CAAI Trans. Intell. Technol. **6**(4), 480–492 (2021). https://doi.org/10.1049/cit2.12048
24. Zhao, W., Tian, H., Wu, Y., Cui, Z., Feng, T.: A new item-based collaborative filtering algorithm to improve the accuracy of prediction in sparse data. Int. J. Comput. Intell. Syst. **15**(1), 15 (2022). https://doi.org/10.1007/s44196-022-00068-7

25. Salloum, S., Rajamanthri, D.: Implementation and evaluation of movie recommender systems using collaborative filtering. J. Adv. Inf. Technol. **12**(3) (2021) https://doi.org/10.12720/jait.12.3.189-196
26. Manochandar, S., Punniyamoorthy, M.: A new user similarity measure in a new prediction model for collaborative filtering. Appl. Intell. **51**(1), 586–615 (2021). https://doi.org/10.1007/s10489-020-01811-3
27. Margaris, D., Spiliotopoulos, D., Karagiorgos, G., Vassilakis, C., Vasilopoulos, D.: On addressing the low rating prediction coverage in sparse datasets using virtual ratings. SN Comput. Sci. **2**(4), 255 (2021). https://doi.org/10.1007/s42979-021-00668-8
28. Nudrat, S., Khan, H.U., Iqbal, S., Talha, M.M., Alarfaj, F.K., Almusallam, N.: Users rating predictions using collaborating filtering based on users and items similarity measures. Comput. Intell. Neurosci. **2022**, 1–13 (2022). https://doi.org/10.1155/2022/2347641
29. Margaris, D., Vassilakis, C., Spiliotopoulos, D., Ougiaroglou, S.: Rating prediction quality enhancement in low-density collaborative filtering datasets. Big Data Cognit. Comput. **7**(2), 59 (2023). https://doi.org/10.3390/bdcc7020059
30. Iftikhar, A., Ghazanfar, M.A., Ayub, M., Mehmood, Z., Maqsood, M.: An improved product recommendation method for collaborative filtering. IEEE Access **8**, 123841–123857 (2020). https://doi.org/10.1109/ACCESS.2020.3005953
31. Fkih, F.: Similarity measures for collaborative filtering-based recommender systems: Review and experimental comparison. J. King Saud Univ. Comput. Inf. Sci. **34**(9), 7645–7669 (2022). https://doi.org/10.1016/j.jksuci.2021.09.014
32. Khojamli, H., Razmara, J.: Survey of similarity functions on neighborhood-based collaborative filtering. Expert Syst. Appl. **185**, 115482 (2021). https://doi.org/10.1016/j.eswa.2021.115482
33. Amer, A.A., Abdalla, H.I., Nguyen, L.: Enhancing recommendation systems performance using highly-effective similarity measures. Knowl.-Based Syst. **217**, 106842 (2021). https://doi.org/10.1016/j.knosys.2021.106842
34. Xu, E., Zhao, K., Yu, Z., Zhang, Y., Guo, B., Yao, L.: Limits of predictability in top-n recommendation. Inf. Process. Manage. **61**(4), 103731 (2024). https://doi.org/10.1016/j.ipm.2024.103731
35. Xu, E., Zhao, K., Yu, Z., Wang, H., Ren, S., Cui, H., Liang, Y., Guo, B.: Upper bound on the predictability of rating prediction in recommender systems. Inf. Process. Manage. **62**(1), 103950 (2025). https://doi.org/10.1016/j.ipm.2024.103950
36. Jiao, X., Wan, S., Liu, Q., Bi, Y., Lee, Y.-L., Xu, E., Hao, D., Zhou, T.: Comparing discriminating abilities of evaluation metrics in link prediction. J. Phys. Comple. **5**(2), 025014 (2024). https://doi.org/10.1088/2632-072x/ad46be
37. Bi, Y., Jiao, X., Lee, Y.-L., Zhou, T.: Inconsistency among evaluation metrics in link prediction. PNAS Nexus **3**(11) (2024) https://doi.org/10.1093/pnasnexus/pgae498
38. Jain, G., Mahara, T., Tripathi, K.N.: A survey of similarity measures for collaborative filtering-based recommender system. In: Pant, M., Sharma, T.K., Verma, O.P., Singla, R., Sikander, A. (eds.) Soft Computing: Theories and Applications vol. 1053, pp. 343–352. Springer, (2020). https://doi.org/10.1007/978-981-15-0751-9_32
39. Bojorque, R., Hurtado, R., Inga, A.: A comparative analysis of similarity metrics on sparse data for clustering in recommender systems. In: Ahram, T.Z. (ed.) Advances in Artificial Intelligence, Software and Systems Engineering vol. 787, pp. 291–299. Springer, (2019).https://doi.org/10.1007/978-3-319-94229-2_28
40. Feng, C., Liang, J., Song, P., Wang, Z.: A fusion collaborative filtering method for sparse data in recommender systems. Inf. Sci. **521**, 365–379 (2020). https://doi.org/10.1016/j.ins.2020.02.052
41. Ramezani, M., Moradi, P., Akhlaghian, F.: A pattern mining approach to enhance the accuracy of collaborative filtering in sparse data domains. Phys. A **408**, 72–84 (2014). https://doi.org/10.1016/j.physa.2014.04.002
42. Jain, G., Mahara, T., Sharma, S.C.: Performance evaluation of time-based recommendation system in collaborative filtering technique. Proc. Comput. Sci. **218**, 1834–1844 (2023). https://doi.org/10.1016/j.procs.2023.01.161
43. Nguyen, L.V., Vo, Q.-T., Nguyen, T.-H.: Adaptive knn-based extended collaborative filtering recommendation services. Big Data Cognit. Comput. **7**(2), 106 (2023). https://doi.org/10.3390/bdcc7020106
44. Sivapalan, S., Sadeghian, A., Rahnama, H., Madni, A.M.: Recommender systems in e-commerce. In: 2014 World Automation Congress (WAC), pp. 179–184 (2014). https://doi.org/10.1109/WAC.2014.6935763
45. Alamdari, P.M., Navimipour, N.J., Hosseinzadeh, M., Safaei, A.A., Darwesh, A.: A systematic study on the recommender systems in the e-commerce. IEEE Access **8**, 115694–115716 (2020). https://doi.org/10.1109/ACCESS.2020.3002803
46. Ismail, S., Abdul-Barik, A., Abdul-Mumin, S.: An enhanced item recommendation approach using the sigmoid function and jaccard similarity coefficient. J. Math. Sci. Comput. Math. **4**(3), 98–120 (2023) https://doi.org/10.15864/jmscm.4310
47. Bao, L.H.Q., Khoa, H.H.B., Thai-Nghe, N.: Image recommendation based on pre-trained deep learning and similarity matching. In: Thai-Nghe, N., Do, T.-N., Haddawy, P. (eds.) Intelligent Systems and Data Science vol. 1949, pp. 258–270. Springer (2024). https://doi.org/10.1007/978-981-99-7649-2_20
48. Singh, P.K., Sinha, S., Choudhury, P.: An improved item-based collaborative filtering using a modified bhattacharyya coefficient and useruser similarity as weight. Knowl. Inf. Syst. **64**(3), 665–701 (2022). https://doi.org/10.1007/s10115-021-01651-8
49. Gupta, A., Shrinath, P.: A novel recommendation system comprising wnmf with graph-based static and temporal similarity estimators. Int. J. Data Sci. Anal. **16**(1), 27–41 (2023)
50. Lan, R., Tian, D., Wu, Q., Li, M.: An improved collaborative filtering model based on time weighted correlation coefficient and inter-cluster separation. Int. J. Mach. Learn. Cybern. **14**(10), 3543–3560 (2023). https://doi.org/10.1007/s13042-023-01849-y
51. Houshmand-Nanehkaran, F., Lajevardi, S.M., Mahlouji-Bidgholi, M.: Optimization of fuzzy similarity by genetic algorithm in user-based collaborative filtering recommender systems. Expert. Syst. **39**(4), 12893 (2022). https://doi.org/10.1111/exsy.12893
52. Behera, G., Nain, N.: Trade-off between memory and model-based collaborative filtering recommender system. In: Dua, M., Jain, A.K., Yadav, A., Kumar, N., Siarry, P. (eds.) Proceedings of the International Conference on Paradigms of Communication, Computing and Data Sciences, pp. 137–146. Springer, ??? (2022). https://doi.org/10.1007/978-981-16-5747-4_12
53. Mann, S.K., Chawla, S.: A proposed hybrid clustering algorithm using k-means and birch for cluster based cab recommender system (cbcrs). Int. J. Inf. Technol. **15**(1), 219–227 (2023). https://doi.org/10.1007/s41870-022-01113-6
54. Zhang, A., Ma, W., Zheng, J., Wang, X., Chua, T.-S.: Robust collaborative filtering to popularity distribution shift. ACM Trans. Inf. Syst. **42**(3), 1–25 (2024). https://doi.org/10.1145/3627159
55. Tran, T.T., Snasel, V., Nguyen, L.T.: Combining social relations and interaction data in recommender system with graph convolution collaborative filtering. IEEE Access **11**, 139759–139770 (2023). https://doi.org/10.1109/ACCESS.2023.3340209
56. Kumar, J., Patra, B.K., Sahoo, B., Babu, K.S.: Group recommendation exploiting characteristics of user-item and collaborative rating of users. Multimed. Tools Appl. **83**(10), 29289–29309 (2023). https://doi.org/10.1007/s11042-023-16799-4
57. Keyhanipour, A.H.: Graph-based comparative analysis of learning to rank datasets. Int. J. Data Sci. Anal. **17**(2), 165–187 (2024)

58. Dor, D., Zwick, U.: Selecting the median. SIAM J. Comput. **28**(5), 1722–1758 (1999). https://doi.org/10.1137/s0097539795288611
59. Xue, Z., Couch, A.: A recommendation system for scientific water data. Int. J. Data Sci. Anal. **12**(1), 61–75 (2021)
60. Permana, A.H.J.P.J., Wibowo, A.T.: Movie recommendation system based on synopsis using content-based filtering with tf-idf and cosine similarity. Int. J. Inf. Commun. Technol. **9**(2), 1–14 (2023)
61. Amit, H.: Spearman Correlation vs Pearson. https://medium.com/@heyamit10/spearman-correlation-vs-pearson-7471eb1d7dd8 (2024)
62. Jain, A., Nagar, S., Singh, P.K., Dhar, J.: Emucf: Enhanced multistage user-based collaborative filtering through non-linear similarity for recommendation systems. Expert Syst. Appl. **161**, 113724 (2020). https://doi.org/10.1016/j.eswa.2020.113724
63. Sundqvist, M., Chiquet, J., Rigaill, G.: Adjusting the adjusted rand index: A multinomial story. Comput. Statistics **38**(1), 327–347 (2022). https://doi.org/10.1007/s00180-022-01230-7
64. Lazarenko, D., Bonald, T.: Pairwise Adjusted Mutual Information. arXiv (2021). ARXIV:2103.12641
65. Ramakrishna, M.T., Venkatesan, V.K., Bhardwaj, R., Bhatia, S., Rahmani, M.K.I., Lashari, S.A., Alabdali, A.M.: Hcof: Hybrid collaborative filtering using social and semantic suggestions for friend recommendation. Electronics **12**(6), 1365 (2023). https://doi.org/10.3390/electronics12061365
66. Zheng, Y.: Context-aware collaborative filtering using context similarity: An empirical comparison. Information **13**(1), 42 (2022). https://doi.org/10.3390/info13010042
67. Anwar, T., Uma, V., Hussain, M.I., Pantula, M.: Collaborative filtering and knn based recommendation to overcome cold start and sparsity issues: A comparative analysis. Multimed. Tools Appl. **81**(25), 35693–35711 (2022). https://doi.org/10.1007/s11042-021-11883-z
68. Sharma, A., S, J.N.K., Rana, D., Setia, S.: A review on collaborative filtering using knn algorithm. In: 2022 OPJU International Technology Conference on Emerging Technologies for Sustainable Development (OTCON), pp. 1–6 (2023). https://doi.org/10.1109/OTCON56053.2023.10113985
69. Chaparala, P., Akurathi, L.S., Bandalapati, P., Akkala, S.: Exploring collaborative filtering methods for product recommendations in e-commerce: A study using amazon and bigbasket datasets. In: 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1–9 (2023). https://doi.org/10.1109/ICCCNT56998.2023.10307285
70. Fkih, F.: Enhancing item-based collaborative filtering by users similarities injection and low-quality data handling. Data Knowl. Eng. **144**, 102126 (2023). https://doi.org/10.1016/j.datak.2022.102126
71. Vahidnia, M.H.: Point-of-interest recommendation in location-based social networks based on collaborative filtering and spatial kernel weighting. Geocarto Int. **37**(26), 13949–13972 (2022). https://doi.org/10.1080/10106049.2022.2086626
72. Xiao, S., Shao, Y., Li, Y., Yin, H., Shen, Y., Cui, B.: Lecf: Recommendation via learnable edge collaborative filtering. Sci. China Inf. Sci. **65**(1), 112101 (2022). https://doi.org/10.1007/s11432-020-3274-6
73. Felfernig, A., Boratto, L., Stettinger, M., Tkalcic, M.: Evaluating group recommender systems. In: Group Recommender Systems. SpringerBriefs in Electrical and Computer Engineering, pp. 59–71. Springer, ??? (2018). https://doi.org/10.1007/978-3-319-75067-5_3
74. Kaya, T., Kaleli, C.: A novel top-n recommendation method for multi-criteria collaborative filtering. Expert Syst. Appl. **198**, 116695 (2022). https://doi.org/10.1016/j.eswa.2022.116695
75. Momanyi, B.M., Zulfiqar, H., Grace-Mercure, B.K., Ahmed, Z., Ding, H., Gao, H., Liu, F.: Cfncm: Collaborative filtering neighborhood-based model for predicting mirna-disease associations. Comput. Biol. Med. **163**, 107165 (2023). https://doi.org/10.1016/j.compbiomed.2023.107165
76. Sgardelis, K., Margaris, D., Spiliotopoulos, D., Vassilakis, C.: Experimental results for Evaluating User Similarity Metrics in Sparse Collaborative Filtering Datasets (Document SODA-TR-24001v2). Software and Database Systems Lab, University of the Peloponnese https://soda.dit.uop.gr/sites/soda.dit.uop.gr/files/publications/technical-reports/soda-TR-24001v2.pdf (2025)
77. Piotrowski, P., Rutyna, I., Baczynski, D., Kopyt, M.: Evaluation metrics for wind power forecasts: A comprehensive review and statistical analysis of errors. Energies **15**(24), 9657 (2022). https://doi.org/10.3390/en15249657
78. Boudt, K., Cornelissen, J., Croux, C.: The gaussian rank correlation estimator: robustness properties. Stat. Comput. **22**(2), 471–483 (2011). https://doi.org/10.1007/s11222-011-9237-0
79. Cheng, K., Khokhar, M.S., Ayoub, M., Jamali, Z.: Nonlinear dimensionality reduction in robot vision for industrial monitoring process via deep three dimensional spearman correlation analysis (d3d-sca). Multimed. Tools Appl. **80**(4), 5997–6017 (2020). https://doi.org/10.1007/s11042-020-09859-6
80. Zhao, Q., Harper, F.M., Adomavicius, G., Konstan, J.A.: Explicit or implicit feedback? engagement or satisfaction? a field experiment on machine-learning-based recommender systems. In: Proceedings of the 33rd Annual ACM Symposium on Applied Computing. SAC '18, pp. 1331–1340. Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3167132.3167275
81. Reusens, M., Lemahieu, W., Baesens, B., Sels, L.: A note on explicit versus implicit information for job recommendation. Decis. Support Syst. **98**, 26–35 (2017). https://doi.org/10.1016/j.dss.2017.04.002
82. Wang, B., Ye, F., Xu, J.: A personalized recommendation algorithm based on the users implicit feedback in e-commerce. Future Internet **10**(12), 117 (2018). https://doi.org/10.3390/fi10120117
83. Sidana, S., Trofimov, M., Horodnytskyi, O., Laclau, C., Maximov, Y., Amini, M.-R.: User preference and embedding learning with implicit feedback for recommender systems. Data Min. Knowl. Disc. **35**(2), 568–592 (2021). https://doi.org/10.1007/s10618-020-00730-8
84. Li, A., Yang, B., Huo, H., Hussain, F.K.: Leveraging implicit relations for recommender systems. Inf. Sci. **579**, 55–71 (2021). https://doi.org/10.1016/j.ins.2021.07.084
85. Liu, B., Luo, Q., Wang, B.: Debiased pairwise learning for implicit collaborative filtering. IEEE Trans. Knowl. Data Eng. **36**(12), 7878–7892 (2024). https://doi.org/10.1109/TKDE.2024.3479240
86. Liu, B., Luo, Q., Wang, B.: Debiased Pairwise Learning from Positive-Unlabeled Implicit Feedback. arXiv (2023). arXiv:2307.15973
87. Najafabadi, M.K., Mahrin, M.N., Chuprat, S., Sarkan, H.M.: Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data. Comput. Hum. Behav. **67**, 113–128 (2017). https://doi.org/10.1016/j.chb.2016.11.010
88. Nez-Valdz, E.R., Cueva Lovelle, J.M., Sanjun Martnez, O., Garca-Daz, V., Pablos, P., Montenegro Marn, C.E.: Implicit feedback techniques on recommender systems applied to electronic books. Comput. Hum. Behav. **28**(4), 1186–1193 (2012). https://doi.org/10.1016/j.chb.2012.02.001
89. Harshalsps19t: Last.FM_dataset. https://www.kaggle.com/datasets/harshal19t/lastfm-dataset (2023)
90. Alhijawi, B., Obeid, N., Awajan, A., Tedmori, S.: New hybrid semantic-based collaborative filtering recommender systems. Int. J. Inf. Technol. **14**(7), 3449–3455 (2022). https://doi.org/10.1007/s41870-022-01011-x
91. Wu, L.: Collaborative filtering recommendation algorithm for mooc resources based on deep learning. Complexity **2021**(1) (2021) https://doi.org/10.1155/2021/5555226

92. Liang, W., Xie, S., Cai, J., Xu, J., Hu, Y., Xu, Y., Qiu, M.: Deep neural network security collaborative filtering scheme for service recommendation in intelligent cyberphysical systems. IEEE Internet Things J. **9**(22), 22123–22132 (2022). https://doi.org/10.1109/jiot.2021.3086845
93. Zhou, H., Xiong, F., Chen, H.: A comprehensive survey of recommender systems based on deep learning. Appl. Sci. **13**(20), 11378 (2023). https://doi.org/10.3390/app132011378
94. Yang, P., Wang, H., Yang, J., Qian, Z., Zhang, Y., Lin, X.: Deep learning approaches for similarity computation: A survey. IEEE Trans. Knowl. Data Eng. **36**(12), 7893–7912 (2024). https://doi.org/10.1109/TKDE.2024.3422484
95. Shi, W., Wang, C., Feng, F., Zhang, Y., Wang, W., Wu, J., He, X.: Lower-left partial auc: An effective and efficient optimization metric for recommendation. In: Proceedings of the ACM Web Conference 2024. Association for Computing Machinery, Newyork, pp. 3253–3264. https://doi.org/10.1145/3589334.3645371
96. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009). https://doi.org/10.1109/mc.2009.263
97. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. arXiv (2020). arXiv:2002.02126