ARTICLE IN PRESS

Journal of King Saud University – Computer and Information Sciences xxx (xxxx) xxx

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison

Fethi Fkih *

Department of Computer Science, College of Computer, Qassim University, Buraydah, Saudi Arabia
BIND Research Group, College of Computer, Qassim University, Buraydah, Saudi Arabia
MARS Research Laboratory, University of Sousse, Sousse, Tunisia

ARTICLE INFO

Article history:
Received 11 March 2021
Revised 23 August 2021
Accepted 11 September 2021
Available online xxxx

Keywords:
Recommender System
Collaborative Filtering
Similarity measure
User-based CF
Item-based CF

ABSTRACT

Collaborative Filtering (CF) filters the flow of data that can be recommended, by a Recommender System (RS), to a target user according to his taste and his preferences. The target user's profile is built based on his similarity with other users. For this reason, CF technique is very sensitive to the similarity measure used to quantify the dependency strength between two users (or two items). In this paper we provide an in-depth review on similarity measures used for CF-based RS. For each measure, we outline its fundamental background and we test its performance through an experimental study. Experiments are carried out on three standard datasets (MovieLens100k, MovieLens1M and Jester) and reveal many important conclusions. In fact, results show that ITR and IPWR are the most suitable similarity measures for a user-based RS while AMI is the best choice for an item-based RS. Evaluation metrics show that under the user-based approach, ITR obtains an MAE equal to 0.786 and 0.731 on MovieLens100k and MovieLens1M, respectively. Whereas, IPWR reach an MAE equal to 3.256 on Jester. Also, AMI gets under the item-based approach an MAE equal to 0.745, 0.724 and 3.281 on MovieLens100k, MovieLens1M and Jester, respectively.

Contents

* Address: Department of Computer Science, College of Computer, Qassim University, Buraydah, Saudi Arabia.
E-mail address: f.fki@qu.edu.sa

Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2021.09.014
1319-1578/© 2021 The Author. Production and hosting by Elsevier B.V. on behalf of King Saud University.

Please cite this article as: F. Fkih, Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2021.09.014

1. Introduction

As a direct consequence of the exponential growth of digital data, Recommendation Systems (RSs) become necessary for filtering the huge flow of information circulating on the web and supposed to be exploitable by an ordinary user. These systems are able to provide recommendations appropriate to user preferences and needs (Iovine et al., 2020; Quijano-Sánchez et al., 2020). Unlike Information Retrieval Systems (Fkih and Omri, 2020, 2018), a user doesn't need to formulate a query to get the required information. Instead, the Recommender Systems try to formulate an 'implicit' query that a given user may demand. This implicit query is built on a set of features extracted from the user profile, such as, his taste, his preferences, his friendship network, etc.

Generally, Recommender Systems (RSs) are classified into three main approaches: content-based (de Gemmis et al., 2015), collaborative-based (Najafabadi et al., 2017) and hybrid-based (Jain et al., 2020) techniques. Content-based filtering (CBF) utilizes user's information (gender, age, interactions on social media, etc.) to forecast his preferences without taking into account any information about other users. Indeed, CBF can be described as an information filtering task since it uses a variety of processes involved in supplying the relevant information to the person who needs it (Abramowicz, 2003). Filtering is often interpreted as removing unwanted data (considered as a noise) from an incoming stream, rather than looking for specific data on that stream. The most popular approach is based on the semantic content of items. It has its roots in the field of Information Retrieval, and uses many of its principles: items are recommended on the basis of a comparison of their content and the user's profile. This profile is presented as a set of items and weights, established from items that the user has deemed relevant. This method is simple, fast and has proven itself in classic Information Retrieval models.

The technique of collaborative Filtering (CF) recommends items, to a target user, based on the opinions of other users (Cacheda et al., xxxx). We should point out that CF approach has some major advantages over CBF since it can be applied on a context that doesn't contain much information associated with the user/item. Also it can be used where it is very difficult to automatically extract the content from the user/item profile (such as sentiments and opinions) (Isinkaye et al., 2015). Hybrid filtering approach combines two or more filtering techniques in order to exploit merits of each one of these techniques. Compared to many hybrid filtering techniques, CF has the ability to be easily implemented with a low complexity (temporal and spatial) (Bobadilla et al., 2013).

CF-based RS can be divided into two sub-approaches: memory-based and model-based approaches (Chen et al., 2018). The first approach uses the entire database in order to find a set of users/items that are similar to the target user/item. While the second approach tries to build a model (a machine learning) describing the user behavior in order to predict his choices. In practice, it has been shown that the memory-based approach offers better performance in terms of precision while the model-based approach is more efficient for handling large data sets (McCarey et al., 2006). Moreover, Model-based approaches are more complex since they involve training a model and tuning several hyperparameters (Valcarce et al., 2019).

Collaborative filtering is widely used in e-commerce environments where users assign scores to products that they viewed or purchased (Neapolitan and Jiang, 2007). This approach consists of making recommendations by looking for correlations between "liked" and "disliked" products among users of the system. For example, a movie Recommender System will search for users similar to the target user; and only movies well rated by these users will be recommended to the target user.

CF-based RS often proceed by comparing the new situation with previous situations. Indeed, the system analyzes the behavior of a group of users (customers) that are similar to the target user or the characteristics of a group of items (products) that are similar to the target item (Jain et al., 2020). This comparison process consists in putting in the same space the situations and to check how much they resemble each other and how much they differ from each other. For example, if a user purchases a product (a book, a car, a movie, etc.) the system will recommend the same item to a set of users having a similar taste as this user. This problem is considered as a clustering process since it aims to classify a set of users/items into homogeneous groups. This clustering is mainly relied on calculating the semantic distance or the similarity that connects elements within a group with each other. Thus, the more the elements share common features, the more the similarity value increases.

For this reason, choosing an appropriate similarity measure among a very large set of available measures is considered as a very important task when implementing a RS. It is obviously clear that the efficiency of the similarity measure has an influence on the RS performance.

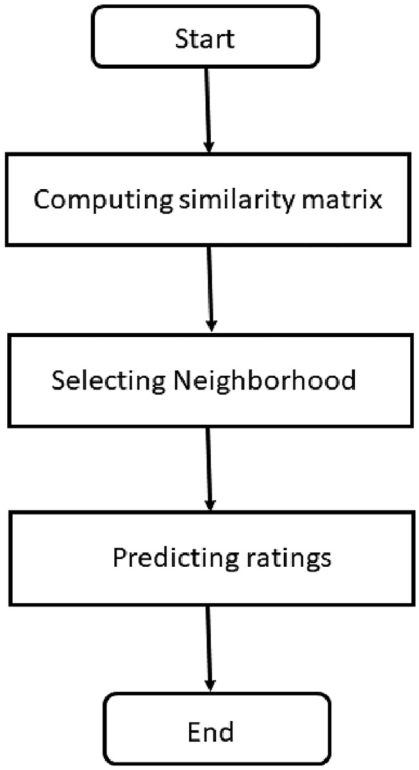


Fig. 1. Flowchart of the Collaborative Filtering approach.

In this work, we intend to carry out an in-depth theoretical literature review (Turner et al., 2018) supported by an experimental comparative study. In fact, we aim to present the theoretical basis of a set of well-known similarity measures frequently used in RS domain and, then, to experimentally compare their performances. Our main goal is to pick up the measure that offers a maximum efficiency to the RS (Fkih and Omri, 2013a). To reach this purpose, we applied these measures on the same datasets and using the same rating prediction method.

The main contributions of this work can be summarized in three points: first, we empirically prove the strong dependency between the RS performance and the used similarity measure. Second, we show that the best choice of a similarity measure varies with the density of the dataset and the type of the used filtering approach. Third, we demonstrate that recent similarity measures using classic measures improved with semantic information can significantly ameliorate the performance of a RS.

The remainder of this paper is structured as follows. In Section 2 we give preliminaries regarding the Collaborative Filtering domain. Therefore, we introduce in Section 3 a literature review of a set of widely used similarity measures in the Recommendation field. In Section 4, we present the methodology used for conducting the experimental comparative study. Then, in Section 5 we provide the obtained findings. Finally, we discuss the obtained results in the Section 6 and we provide some prospects that might improve the similarity measures performance for CF-based RS. We have to mention that all experiments are carried out on three standards datasets.

2. Preliminaries

Collaborative filtering strategy is based on the assumption that if a user is looking for an information, he may use what others have already found and evaluated. For each user of a collaborative filtering system, a set of similar neighbors is identified, and the decision whether or not to recommend an item to a user will depend on the opinions of members of his neighborhood. Collaborative filtering employs statistical methods to make predictions based on patterns of user interests. These predictions are used to make recommendation to a potential user, based on the correlation between his profile and the profiles of other users with similar interests and tastes. In this context, users provide feedback, in the form of ratings, to build their profiles. These ratings are compared with those provided by other users to generate a similarity matrix. Generally, there are two main methods under the umbrella of CF-based systems: User-based CF and Item-based CF (Najafabadi et al., 2017). We mention that Fig. 1 summarizes the process of the CF-based recommendation.

The following sections provide preliminaries on the fundamental background of the user-based and the item-based approaches.

2.1. User-based CF

The idea is to filter the flow of incoming items based on the evaluation given by other users in the community that have already reviewed the items. If an item has been found interesting by a user, it will automatically be recommended to users who have shared similar opinions in the past. To reach this goal, this system has to build a Users×Users matrix to store the similarity scores between users. Therefore, the potential rating given by the active user to an item will be calculated according to his similar neighbors. This process consists of three steps: similarity computation, neighborhood selection and rating prediction. We have to mention that the similarity computation phase will be further detailed in Section 3.

2.1.1. Neighborhood selection

In the literature, two methods were proposed for selecting the set of nearest neighbors: Max number (top-k) of neighbors (Shardanand and Maes, 1995) and correlation threshold (Resnick et al., 1994). The top-k technique selects the k-nearest users (in terms of similarity), where k denotes the number of users. The correlation threshold technique defines a threshold and keeps only the users whose similarities, with the active user, exceed the threshold.

2.1.2. Ratings prediction

In order to predict the rating of an active user, many measures were proposed. The most popular used measure in this field is the weighted sum method (Sarwar et al., 2001) as shown in Eq. (1):

$$\tilde{r_{ui}} = \frac{\sum_{v \in N_u^i} Sim_{uv} * r_{vi}}{\sum_{v \in N_u^i} \mid Sim_{uv} \mid} \tag{1}$$

where N_u^i is the set of neighbors ($top - k$ users) that are most similar to user u and have rated item i, v is a user that belongs to N_u^i and Sim_{uv} is the similarity value between users u and v.

Also, the mean-centered prediction function proposed by (Aggarwal, 2016) is a common aggregation measure. Eq. 2 shows the mean-centered formula (we keep the same variables as Eq. 1):

$$\tilde{r_{ui}} = \overline{r_u} + \frac{\sum_{v \in N_u^i} Sim_{uv} * (r_{vi} - \overline{r_v})}{\sum_{v \in N_u^i} \mid Sim_{uv} \mid} \tag{2}$$

where $\overline{r_u}$ and $\overline{r_v}$ are the average ratings assigned by users u and v, respectively.

2.2. Item-based CF

This technique uses an Items×Items matrix to store the similarity scores between items. In practice, the system will recommend items which are the most similar to a set of items already rated with a high score by the active user. Indeed, the predicted rating depends on the similarity value between the item and its neighbor: the more the similarity increases, the more the predicted rating is similar to rating's neighbor. As same as the User-based CF technique, the Item-based CF process can be summarized into three steps: similarity computation, neighborhood selection and ratings prediction. We mention that the similarity computation phase will be further detailed in Section 3.

2.2.1. Neighborhood selection

As same as the User-based technique, there are two proposed methods: Max number (top-k) of neighbors (Shardanand and Maes, 1995) and correlation threshold (Resnick et al., 1994). The top-k technique selects the k-nearest items, where k denotes the number of items. The correlation threshold technique sets a threshold and maintains items whose similarities, with the active item, exceed the threshold.

2.2.2. Ratings prediction

For ratings prediction, several measures were proposed. The more popular measures are as follows: Z-score (Ricci et al., 2010), Weighted Sum (Sarwar et al., 2001) and Mean-Centered aggregation (Aggarwal, 2016). The weighted sum formula is as follows (Eq. 3):

$$\tilde{r_{ui}} = \frac{\sum_{j \in N_i^u} Sim_{ij} * r_{uj}}{\sum_{j \in N_i^u} \mid Sim_{ij} \mid} \tag{3}$$

where N_i^u is the set of neighbor ($top-k$ items) that are most similar to item i and have been rated by the user u, j is an item that belongs to N_i^u and Sim_{ij} is the similarity value between items i and j.

The mean-centered aggregation formula is provided in Eq. (4) (we keep the same variables as Eq. (3)):

$$\tilde{r_{ui}} = \overline{r_i} + \frac{\sum_{j \in N_i^u} Sim_{ij} * (r_{uj} - \overline{r_j})}{\sum_{j \in N_i^u} \mid Sim_{ij} \mid} \tag{4}$$

where $\overline{r_i}$ and $\overline{r_j}$ are average ratings on items i and j, respectively.

3. Similarity measures: A review

In this section, we outline the theoretical foundation of a set of selected similarity measures. For each measure we supply a brief description and its corresponding formula.

3.1. Vector similarity (Cosine)

This technique (Breese et al., 1998) presents a user as a vector of ratings rated by himself and an item as a vector of ratings rated by the set of users (Cacheda et al., xxxx). The cosine between two vectors representing two users (or items) indicates the similarity value between each other. A value close to 1 indicates that it exists a strong correlation between the two variables. A value close to 0 indicates that there is no correlation (independent variables). Formulas (5) and (6) (Chen et al., 2018) represent the cosine measure for users and items, respectively.

$$Cosine(u,v) = \frac{\sum_{i \in I_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{u \in I_u} r_{ui}^2} \sqrt{\sum_{u \in I_v} r_{vi}^2}} \tag{5}$$

where I_u and I_v denote the sets of items rated by users u and v, respectively, and I_{uv} denotes the set of items commonly rated by both u and v. r_{ui} and r_{vi} are the ratings values on item i given by users u and v, respectively.

$$Cosine(i,j) = \frac{\sum_{u \in U_{ij}} r_{ui} r_{uj}}{\sqrt{\sum_{u \in U_i} r_{ui}^2} \sqrt{\sum_{u \in U_j} r_{uj}^2}} \tag{6}$$

where U_i and U_j denote the sets of users who rated the items i and j, respectively, and U_{ij} represents the set of users who rated both items i and j. r_{ui} and r_{uj} are the rating values assigned by the same user u on items i and j, respectively.

3.2. Adjusted cosine vector

The Adjusted Cosine measure (Adomavicius et al., 2011) calculates the correlation value between two users (formula (7)) or two items (formula (8)).

$$ACosine(u,v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \overline{r_i})(r_{vi} - \overline{r_i})}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \overline{r_i})^2} \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \overline{r_i})^2}} \tag{7}$$

where I_{uv} denotes the set of items commonly rated by both u and v. $\overline{r_i}$ denotes the average ratings on i. r_{ui} and r_{vi} denote, respectively, the ratings of the user u and v on the item i.

$$ACosine(i,j) = \frac{\sum_{u \in U_{ij}} (r_{ui} - \overline{r_u})(r_{uj} - \overline{r_u})}{\sqrt{\sum_{u \in U_{ij}} (r_{ui} - \overline{r_u})^2} \sqrt{\sum_{u \in U_{ij}} (r_{uj} - \overline{r_u})^2}} \tag{8}$$

where U_{ij} represents the set of users who rated both items i and j. $\overline{r_u}$ denotes the average ratings by u. We mention that r_{ui} and r_{uj} are the ratings of user u on items i and j, respectively.

3.3. Pearson Correlation Coefficient (PCC)

This measure was proposed by Karl Pearson (Pearson, 1895) to measure linear relationships and became widely used in statistics field. PCC formula returns a value between −1 and 1, where: 1 indicates a strong positive correlation, −1 indicates a strong negative correlation and 0 indicates no correlation at all (Resnick et al., 1994). The following formula 9 calculates the similarity between two users u and v:

$$PCC(u,v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \overline{r_u})(r_{vi} - \overline{r_v})}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \overline{r_u})^2} \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \overline{r_v})^2}} \tag{9}$$

where I_{uv} denotes the set of items commonly rated by both u and v. $\overline{r_u}$ and $\overline{r_v}$ denote the average ratings of the users u and v on item i in I_{uv}, respectively. r_{ui} and r_{vi} are ratings of users u and v on the same item i. Formula (10) calculates the similarity between two items i and j:

$$PCC(i,j) = \frac{\sum_{u \in U_{ij}} (r_{ui} - \overline{r_i})(r_{uj} - \overline{r_j})}{\sqrt{\sum_{u \in U_{ij}} (r_{ui} - \overline{r_i})^2} \sqrt{\sum_{u \in U_{ij}} (r_{uj} - \overline{r_j})^2}} \tag{10}$$

where U_{ij} represents the set of users who rated both items i and j, $\overline{r_i}$ and $\overline{r_j}$ denote the average ratings on i and j in U_{ij}, respectively. r_{ui} and r_{uj} are ratings of user u on items i and j, respectively.

3.4. Adjusted mutual information

According to Shannon (Shannon, 2001), the Mutual Information (MI) is a measure commonly used in the information theory domain. In our case, MI is used to compute the statistical dependence between two users (u and v) or two items (i and j) (Brun et al., 2009). To do this, a user is presented as a vector of his ratings on the set of items and an item is presented as a vector of its ratings rated by the set of users. The MI formula representing the correlation between two users u and v is as follows (11):

$$MI(u,v) = \sum_{i \in I_u} \sum_{j \in I_v} p(r_{ui}, r_{vj}) \log \frac{p(r_{ui}, r_{vj})}{p(r_{ui}) p(r_{vj})} \quad (11)$$

where I_u and I_v are the sets of items rated by users u and v, respectively. Also, r_{ui} denotes the rating of user u on item i and r_{vj} the rating of user v on item j. Eq. (12) provides the correlation value between two items i and j.

$$MI(i,j) = \sum_{u \in U_i} \sum_{v \in U_j} p(r_{ui}, r_{vj}) \log \frac{p(r_{ui}, r_{vj})}{p(r_{ui}) p(r_{vj})} \quad (12)$$

where U_i and U_j are the sets of users rated items i and j, respectively. r_{ui} denotes the rating value of an item i by a user u and r_{vj} denotes the rating assigned to an item j by a user v. Adjusted Mutual Information (AMI) is a variation of MI used to calculate statistical correlation (Vinh et al., 2009), it returns a value between 0 and 1. If an effect of agreement due to chance between statistical variables happens, then it will be corrected by the AMI (adjustment for chance). The following Eq. (13) calculates the Adjusted Mutual Information between two users u and v (the same for items i and j).

$$AMI(u,v) = \frac{MI(u,v) - E\{MI(u,v)\}}{Max(H(u), H(v)) - E\{MI(u,v)\}} \quad (13)$$

where $H(u)$ is the entropy of u and $E\{MI(u,v)\}$ is the expected mutual information between two users u and v (Vinh et al., 2009).

3.5. Adjusted Rand index

In order to use the Rand index (RI) measure, we suppose that users u and v (or items i and j) are clusters of ratings. In our context, a user is represented as a cluster of his ratings on the set of items and an item is considered as a cluster of its ratings rated by the set of users. To define the formula of Rand index, we have to define the following 3 parameters (Rand, 1971):

- a is the number of pairs of ratings that are grouped together in both clusters.
- b is the number of pairs of ratings that are not grouped together in both clusters.
- N is the total number of ratings in both clusters.

Then, the Rand index is defined as follows (Eq. 14):

$$RI = \frac{a+b}{\binom{N}{2}} \quad (14)$$

where $\binom{N}{2}$ is the number of unordered pairs in a set of N ratings.

The adjusted Rand index (ARI) is the corrected for chance version of the Rand index (Rand, 1971; Vinh et al., 2009; Sinnott et al., 2016). It returns a value between 0 (absence of correlation) and 1 (identical clustering). The formula is defined in Eq. 15 (the same for items i and j):

$$ARI(u,v) = \frac{RI(u,v) - E\{RI(u,v)\}}{Max(RI(u), RI(v)) - E\{RI(u,v)\}} \quad (15)$$

where $E\{RI(u,v)\}$ is the expected Rand index between two clusters u and v (Vinh et al., 2009).

3.6. Spearman rank-order correlation coefficient

The Spearman correlation (Spearman, 2010) evaluates the monotonic relationship between two variables. In a monotonic relationship, if the value of the first variable changes then the value of the second variable changes as well, but without a constant rate (is not linear). Spearman rank-order correlation coefficient, named ρ, can take a value between -1 and 1. It exists a strong relation between the Spearman correlation and Pearson Correlation Coefficient since $\rho(u,v)$ is considered as the PCC between the rank variables (Szczepanska, 2011). To calculate the Spearman rank correlation ρ, we use the following formulas: Eq. (16) for users and Eq. (17) for items.

$$\rho(u,v) = 1 - \frac{6\sum_{i \in I_{uv}} (Rank(r_{ui}) - Rank(r_{vi}))^2}{n(n^2-1)} \quad (16)$$

where $\rho(u,v)$ represents the Spearman rank correlation between two users u and v, I_{uv} denotes the set of items commonly rated by both u and v, r_{ui} and r_{vi} denote the rating of the user u and v, respectively, on the item i. $Rank(r_{ui})$ and $Rank(r_{vi})$ denote the rank of r_{ui} and r_{vi} in the vector u and v, respectively. n denotes the number of common ratings between u and v.

$$\rho(i,j) = 1 - \frac{6\sum_{u \in U_{ij}} (Rank(r_{ui}) - Rank(r_{uj}))^2}{n(n^2-1)} \quad (17)$$

where $\rho(i,j)$ represents the Spearman rank correlation between two items i and j, U_{ij} represents the set of users who rated both items i and j, r_{ui} and r_{uj} denote the rating of the user u on items i and j, respectively. $Rank(r_{ui})$ and $Rank(r_{uj})$ denote, respectively, the rank of r_{ui} and r_{uj} in the vector u, n denotes the number of common ratings between i and j.

3.7. Kendall's tau

As Spearman's rank correlation coefficient, Kendall's Tau (τ) (Kendall, 1938; Kendall and Gibbons, 1990) evaluates statistical monotone relationships based on the ranks of the data. The value returned by τ ranges from -1 (as the rank of one variable increases the other one decreases) to 1 (the ranks of both variables increase together), while 0 indicates that is no relationship between the two variables. This measure is mainly based on counting the concordant pairs (ordered in the same way) and discordant pairs (ordered differently) (Conover, 1971; Koh and Owen, 2000). Kendall's tau for computing the association strength between two vectors of ratings is defined as (Eq. (18)):

$$\tau = \frac{c-d}{c+d} \quad (18)$$

where c is the number of concordant pairs and d is the number of discordant pairs.

3.8. Jaccard

The Jaccard index (Jaccard, 1912), denoted by J, computes the similarity and diversity of two sets. The Jaccard coefficient between two finite set is defined as the cardinality of the intersection divided by the cardinality of the union. That is to say, it measures the ratio of the number of elements shared between the two sets to the total number of elements in both sets. J index takes a value between 0 and 1, the closer the index to 1, the more similar the two vectors. The following Eq. (19) calculates the Jaccard index of two vectors u and v, while u and v can be users (set of ratings assigned by the same user) or items (a set of ratings assigned to the same item).

$$J(u,v) = \frac{|u \cap v|}{|u \cup v|} \tag{19}$$

3.9. Euclidean distance

The Euclidean distance (O'Neill, 2006), denoted by d, from a user u to a user v (or from an item i to an item j) is the length of a line segment between the two users (or items) in the Euclidean space. In practical terms, each user is represented by its Cartesian coordinates with respect to the basis of items (the same thing for an item which is represented with respect to the basis of users) and the distance between two users (or two items) is the absolute value of the numerical difference of their coordinates. The Euclidean distance (d) formula representing the correlation between two users u and v is as follows (20):

$$d(u,v) = \sqrt{\sum_{i \in I_{uv}} (r_{vi} - r_{ui})^2} \tag{20}$$

where I_{uv} denotes the set of items commonly rated by both u and v, r_{ui} and r_{vi} denote the rating of the user u and v, respectively, on the item i. The formula (21) provides the Euclidean distance between two items i and j.

$$d(i,j) = \sqrt{\sum_{u \in U_{ij}} (r_{uj} - r_{ui})^2} \tag{21}$$

where U_{ij} represents the set of users who rated both items i and j, r_{ui} and r_{uj} denote the rating of the user u on items i and j, respectively. The Euclidean distance should be normalized to become a similarity measure. Formulas (22) and (23) define the Euclidean similarity (ES) measures for users and items, respectively.

$$ES(u,v) = \frac{1}{1 + d(u,v)} \tag{22}$$

$$ES(i,j) = \frac{1}{1 + d(i,j)} \tag{23}$$

3.10. Manhattan distance

The Manhattan distance, also known as city blocks and taxicab, between two vectors is equal to the one-norm of the distance between the vectors (Szabo, 2015). To adapt this measure to the RS domain, we have to represent a user by its Cartesian coordinates with respect to the basis of items (the same thing for an item). The Manhattan distance (d_1) between two users u and v is as follows (formula 24):

$$d_1(u,v) = \sum_{i \in I_{uv}} (|r_{vi} - r_{ui}|) \tag{24}$$

where I_{uv} denotes the set of items commonly rated by both u and v, r_{ui} and r_{vi} denote the rating of the user u and v, respectively, on the item i. Formula (25) presents the Euclidean distance between two items i and j.

$$d_1(i,j) = \sum_{u \in U_{ij}} (|r_{uj} - r_{ui}|) \tag{25}$$

where U_{ij} represents the set of users who rated both items i and j, r_{ui} and r_{uj} denote the rating of the user u on items i and j, respectively. In order to convert the Manhattan distance into a similarity measure (MS), we use the following formulas (30 and 31) for users and items, respectively.

$$MS(u,v) = \frac{1}{1 + d_1(u,v)} \tag{26}$$

$$MS(i,j) = \frac{1}{1 + d_1(i,j)} \tag{27}$$

3.11. Chebyshev distance

The Chebyshev distance between two vectors is the greatest of their differences along any coordinate dimension (Abello et al., 2002). A user can be modeled by its coordinates with respect to the basis of items (the same thing for an item). Thus, the Chebyshev distance ($d_{Chebyshev}$) between two users u and v is provided as follows (formula 28):

$$d_{Chebyshev}(u,v) = max_{i \in I_{uv}} (|r_{vi} - r_{ui}|) \tag{28}$$

where I_{uv} denotes the set of items commonly rated by both u and v, r_{ui} and r_{vi} denote the rating of the user u and v, respectively, on the item i. Formula (29) presents the Chebyshev distance between two items i and j.

$$d_{Chebyshev}(i,j) = max_{u \in U_{ij}} (|r_{uj} - r_{ui}|) \tag{29}$$

where U_{ij} represents the set of users who rated both items i and j, r_{ui} and r_{uj} denote the rating of the user u on items i and j, respectively. The similarity measures for users and items, using the Chebyshev distance, are defined as follows (formulas 30 and 31):

$$Ch(u,v) = \frac{1}{1 + d_{Chebyshev}(u,v)} \tag{30}$$

$$Ch(i,j) = \frac{1}{1 + d_{Chebyshev}(i,j)} \tag{31}$$

3.12. Improved triangle similarity complemented with user rating preferences (ITR)

This similarity measure, called ITR, was recently proposed in (Iftikhar et al., 2020) which consists of a product of two terms: the improved triangle similarity $sim^{TRIANGLE'}$ (Iftikhar et al., 2020) and the user rating preferences URP (Ahn, 2008). In practice, $sim^{TRIANGLE'}$ is considered as an improvement of the triangle multiplying Jaccard (TMJ) similarity (Sun et al., xxxx). Thus, Improved triangle similarity (ITR) not only focuses on common ratings, such as TMJ measure, but also takes into account the non-common rating of users. The ITR similarity between two users u and v is defined as follows (formula 32):

$$sim^{ITR}(u,v) = sim^{TRIANGLE'}(u,v) * sim^{URP}(u,v) \tag{32}$$

We note that $sim^{TRIANGLE'}(u,v)$ and $sim^{URP}(u,v)$ are defined in formulas (33) and (34), respectively.

$$sim^{TRIANGLE'}(u,v) = 1 - \frac{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - r_{vi})^2}}{\sqrt{\sum_{i \in I_{uv}} r_{ui}^2} + \sqrt{\sum_{i \in I_{uv}} r_{vi}^2}} \tag{33}$$

where I_{uv} denotes the set of items either rated by u or v. r_{ui} and r_{vi} are ratings of users u and v on the same item i.

$$sim^{URP}(u,v) = 1 - \frac{1}{1 + exp(-|\overline{r_u} - \overline{r_v}| * |\sigma_u - \sigma_v|)} \tag{34}$$

where $\overline{r_u}$ and $\overline{r_v}$ denote the mean ratings of users u and v on item i in I_{uv}, respectively. σ_u (formula 35) and σ_v represent the standard variance of u and v, respectively.

$$\sigma_u = \sqrt{\frac{\sum_{i \in I_u} (r_{ui} - \overline{r_u})^2}{|I_u|}} \tag{35}$$

where I_u is the set of items rated by the user u.

Also, the ITR measure can be used to calculate the similarity between two items, formula (36) presents the similarity between two items i and j:

$$sim^{ITR}(i,j) = sim^{TRIANGLE'}(i,j) * sim^{URP}(i,j) \quad (36)$$

where $sim^{TRIANGLE'}(i,j)$ and $sim^{URP}(i,j)$ are defined in formulas (37) and (38), respectively.

$$sim^{TRIANGLE'}(i,j) = 1 - \frac{\sqrt{\sum_{u\in U_{ij}}(r_{ui} - r_{uj})^2}}{\sqrt{\sum_{u\in U_{ij}} r_{ui}^2} + \sqrt{\sum_{u\in U_{ij}} r_{uj}^2}} \quad (37)$$

where U_{ij} represents the set of users who rated items i or j, r_{ui} and r_{uj} are ratings of user u on items i and j, respectively.

$$sim^{URP}(i,j) = 1 - \frac{1}{1 + exp(-|\overline{r_i} - \overline{r_j}| * |\sigma_i - \sigma_j|)} \quad (38)$$

We mention that $\overline{r_i}$ and $\overline{r_j}$ denote the mean ratings on i and j in U_{ij}, respectively. σ_i (formula 39) and σ_j represent the standard variance of i and j, respectively.

$$\sigma_i = \sqrt{\frac{\sum_{u\in U_i}(r_{ui} - \overline{r_i})^2}{|U_i|}} \quad (39)$$

where U_i is the set of users who rated the item i.

3.13. Improved PCC weighted with RPB (IPWR)

The improved PCC weighted with RPB (IPWR) similarity measure was proposed in (Ayub et al., 2019). This measure combines an improved PCC (see Section 3.3), denoted by *Sim_IPCC*, with user rating preference behavior (RPB). The IPWR similarity between two users u and v is presented as follows (Eq. 40):

$$IPWR(u, v) = \alpha * RPB(u, v) + \beta * Sim\_IPCC(u, v) \quad (40)$$

The best weights of α and β that offer the optimal performance to the RS are determined empirically as detailed in (Ayub et al., 2019). The RPB formula is given as follows (Eq. 41):

$$RPB(u, v) = cos(|\overline{r_v} - \overline{r_u}| * |SD_v - SD_u| \quad (41)$$

where $\overline{r_v}$ denotes the mean ratings of the user v. The standard deviation of the user u (SD_u) is given in Eq. (42):

$$SD_u = \sqrt{\frac{\sum_{i\in I_u}(r_{ui} - \overline{r_u})^2}{|I_u|}} \quad (42)$$

where I_u is the set of items rated by the user u. The improved PCC similarity between two users u and v ($Sim\_IPCC(u, v)$) is provided by formula (43):

$$Sim\_IPCC(u, v) = \frac{\sum_{i\in I_{uv}}[(r_{ui} * \overline{r_u}) - (r_{ui} * \overline{r_i})] * [(r_{vi} * \overline{r_v}) - (r_{vi} * \overline{r_i})]}{\sqrt{\sum_{i\in I_u}[(r_{ui} * \overline{r_u}) - (r_{ui} * \overline{r_i})]^2}\sqrt{\sum_{i\in I_v}[(r_{vi} * \overline{r_v}) - (r_{vi} * \overline{r_i})]^2}} \quad (43)$$

where I_{uv} denotes the set of items commonly rated by both u and v, I_u and I_v are the sets of items rated by the user u and the set of items rated by the user v, respectively. $\overline{r_u}$ and $\overline{r_v}$ denote the average ratings of the users u and v on I_{uv}, respectively. $\overline{r_i}$ is the mean ratings on the item i. r_{ui} and r_{vi} are ratings of users u and v on the same item i.

Also, IPWR can be adapted for calculating similarity between two items i and j as shown in formula 44:

$$IPWR(i,j) = \alpha * RPB(i,j) + \beta * Sim\_IPCC(i,j) \quad (44)$$

Eq. (45) presented the RPB formula:

$$RPB(i,j) = cos(|\overline{r_j} - \overline{r_i}| * |SD_j - SD_i| \quad (45)$$

where $\overline{r_i}$ denotes the mean ratings on the item i. The standard deviation of the item i (SD_i) is provided in Eq. (46):

$$SD_i = \sqrt{\frac{\sum_{u\in U_i}(r_{ui} - \overline{r_i})^2}{|U_i|}} \quad (46)$$

where U_i is the set of users who rated the item i. The improved PCC similarity between two users u and v ($Sim\_IPCC(u, v)$) is provided by formula (47):

$$Sim\_IPCC(i,j) = \frac{\sum_{u\in U_{ij}}[(r_{ui} * \overline{r_i}) - (r_{ui} * \overline{r_u})] * [(r_{uj} * \overline{r_j}) - (r_{uj} * \overline{r_u})]}{\sqrt{\sum_{u\in U_i}[(r_{ui} * \overline{r_i}) - (r_{ui} * \overline{r_u})]^2}\sqrt{\sum_{u\in U_j}[(r_{uj} * \overline{r_j}) - (r_{uj} * \overline{r_u})]^2}} \quad (47)$$

where U_{ij} denotes the set of users who commonly rated both i and j, U_i and U_j are the sets of users who rated the item i and the set of users who rated the item j, respectively. $\overline{r_i}$ and $\overline{r_j}$ denote the average ratings on the items i and j by U_{ij}, respectively. $\overline{r_u}$ is the mean ratings by the user u. r_{ui} and r_{uj} are the ratings of the user u on the items i and j, respectively.

4. Methodology

In this section, we carried out an experimental comparative study between the thirteen correlation measures described in the Section 3:

1. Vector Similarity (Cosine): formulas (5) and (6).
2. Adjusted Cosine Vector (ACosine): formulas (7) and (8).
3. Pearson Correlation Coefficient (PCC): formulas (9) and (10).
4. Adjusted Mutual Information (AMI): formula (13).
5. Adjusted Rand index (ARI): formula (15).
6. Spearman rank-order correlation coefficient (ρ): formulas (16) and (17).
7. Kendall's tau (τ): formula (18.
8. Jaccard index (J): formula 19).
9. Euclidean similarity (ES): formulas (22) and (23).
10. Manhattan similarity (MS): formulas (22) and (23).
11. Chebyshev similarity (Ch): formulas 30 and 31.
12. Improved triangle similarity complemented with user rating preferences (ITR): formulas (32) and (36).
13. Improved PCC weighted with RPB (IPWR): formulas (40) and (44).

To reach this purpose, we used the methodology described in the Section 2: we computed the similarity matrix, we selected the neighborhood and we predicted the missing ratings. For the prediction phase, we used the weighted sum equations (1) and (3). We have to mention that we performed this evaluation study for both: user-based and item-based CF.

The experimental tests were conducted on an intel (R) Core™ i7 machine having a clock frequency of 2.3 Ghz and 16 GB of RAM running Windows 10 and Python programming language. To estimate the skill of the correlation measures for CF-based RS and to reduce errors, we use a 10-fold cross-validation procedure. Besides, we split the data into 10 parts and we take for each time 9 parts (90%) as a training set and 1 part (10%) as a test. Note that each part consists of a randomly 10% ratings for each user. The final result will be calculated as an average of all the results provided by the 10 tests. We have to mention that we used Python libraries to implement the Recommender System and the similarity measures (Pedregosa et al., 2011).

In the following sections, we introduce the datasets used for the experimental phase and we present the different metrics used for evaluating the performance of each similarity measure.

Table 1
Datasets description.

Dataset	Ratings	Users	Items	Sparsity	Density	Rating range
MovieLens100k	100000	943	1682	93.7%	6.3%	1..5
MovieLens1M	1000209	6040	3900	95.75%	4.25%	1..5
Jester	1829974	24983	100	27.48%	72.52%	−10..10

4.1. Datasets

For the experimental study, we used 3 standards datasets: movieLens100k,[1] movieLens1M[2] (Harper and Konstan, xxxx and Jester[3] (Goldberg et al., 2001). Table 1 provides a description of each dataset. We have to mention that each user in movieLens100k and movieLens1M has rated at least 20 items. However, each user in Jester has rated at least 36 items which justify the high density of this dataset. We can remark that MovieLens100k and MovieLens1M are similar in many aspects like the high sparsity (the low density) and the rating range (1–5). By cons, Jester is a dense dataset (with a low sparsity) and its rating ranges between −10 and 10.

4.2. Evaluation metrics

To evaluate the performance of the RS, we intend to utilize each time a different similarity measure. Therefore, we evaluate the performance of the RS using 4 well-known evaluation metrics which are widely used for evaluating regression models (Silveira et al., 2019):

- Mean Absolute Error (MAE): it evaluates the difference between the ratings predicted by the RS and the ratings given by the users. It returns a positive value.
- Normalized Mean Absolute Error (NMAE): since we have used different datasets with different rating ranges, it will be more convenient to use a normalized version of the MAE in order to compare results provided by each dataset. This measure is independent of the rating scale and returns a value from 0 to 1 (Ekstrand et al., 2011).
- Root Mean Square Error (RMSE): it calculates a larger difference for large errors in the rating prediction (Herlocker et al., 2004). It returns a positive value.
- R-squared (R^2), is a widely used goodness-of-fit measure (Colin Cameron and Windmeijer, 1997). It quantifies the degree of collinearity between observations and model-calculated values (Ritter and Muñoz-Carpena, 2013).

MAE, NMAE and RMSE are defined in Eq. (48)–(50), respectively. Where n denotes the number of ratings to be predicted (the total number of ratings in the data test), $\tilde{r}_i$ is the predicted value of the rating i, r_i is the real value of the rating i, r_{max} is the maximum rating value available and r_{min} is the minimum rating value available in the rating scale (Polatidis and Georgiadis, 2017. Note that the closer to 0 the MAE, NMAE and RMSE, the better the RS performance.

$$MAE = \frac{\sum_{i=1}^{n} \mid \tilde{r}_i - r_i \mid}{n} \tag{48}$$

$$NMAE = \frac{MAE}{r_{max} - r_{min}} \tag{49}$$

Table 2
Minimum value of RMSE per measure for user-based approach. Best values are highlighted in bold.

Dataset	Measure	Min RMSE	Top-K users
MovieLens100k	PCC	1.0137	300
	AMI	1.0565	300
	τ	1.0542	300
	Cosine	1.0151	210
	ρ	1.0543	300
	ARI	1.0560	300
	ACosine	1.0238	300
	J	1.0057	300
	ES	1.1213	300
	MS	1.1496	300
	Ch	1.0456	300
	ITR	**0.9921**	300
	IPWR	1.0014	300
MovieLens1M	PCC	0.9670	300
	AMI	0.9821	300
	τ	0.9745	300
	Cosine	0.9672	300
	ρ	0.9741	300
	ARI	0.9841	300
	ACosine	0.9888	300
	J	0.9542	300
	ES	1.1845	300
	MS	1.2240	300
	Ch	1.0480	300
	ITR	**0.9428**	300
	IPWR	0.9529	300
Jester	PCC	4.2311	300
	AMI	4.8406	300
	τ	4.9063	300
	Cosine	4.3233	300
	ρ	4.9139	300
	ARI	5.0184	300
	ACosine	4.3343	300
	J	4.6329	170
	ES	4.6546	90
	MS	4.7858	300
	Ch	4.5055	300
	ITR	4.3189	210
	IPWR	**4.1677**	300

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\tilde{r}_i - r_i)^2}{n}} \tag{50}$$

In general, R^2 (Eq. 51) returns a value between 0 and 1. In a linear regression context, The model try to predict or explain an outcome. In our case, we have an observed vector of ratings Y and a predicted vector of ratings $\tilde{Y}$. In fact, Y is called the independent variable and $\tilde{Y}$ is called the dependent variable. R^2 shows the proportion of the variance in $\tilde{Y}$ that is predicted or explained by Y through the linear regression. In the best case, the measure returns 1 which means that the model performs an optimal prediction. A 0 value indicates that the model has a worse ability to predict the outcome variable ($\tilde{Y}$). However, R^2 can be negative which means that the correlation between Y and $\tilde{Y}$ is negative.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (r_i - \tilde{r}_i)^2}{\sum_{i=1}^{n} (r_i - \bar{r})^2} \tag{51}$$

[1] https://grouplens.org/datasets/movielens/100k/.
[2] https://grouplens.org/datasets/movielens/1m/.
[3] https://goldberg.berkeley.edu/jester-data/.

Table 3
Minimum value of MAE per measure for user-based approach. Best values are highlighted in bold.

Dataset	Measure	Min MAE	Top-K users
MovieLens100k	PCC	0.8041	210
	AMI	0.8395	300
	τ	0.8376	300
	Cosine	0.8058	180
	ρ	0.8378	300
	ARI	0.838	300
	ACosine	0.8111	300
	J	0.7913	180
	ES	0.8807	300
	MS	0.8900	300
	Ch	0.8277	300
	ITR	**0.7863**	220
	IPWR	0.7915	300
MovieLens1M	PCC	0.7674	300
	AMI	0.7742	300
	τ	0.7725	300
	Cosine	0.7704	300
	ρ	0.7726	300
	ARI	0.7782	300
	ACosine	0.7802	300
	J	0.7465	300
	ES	0.9184	300
	MS	0.9302	300
	Ch	0.8243	300
	ITR	**0.7315**	300
	IPWR	0.7431	300
Jester	PCC	3.3817	300
	AMI	3.9703	300
	τ	4.0296	300
	Cosine	3.4567	300
	ρ	4.0335	300
	ARI	4.1434	300
	ACosine	3.4943	300
	J	3.7554	110
	ES	3.8898	70
	MS	4.0098	230
	Ch	3.7309	300
	ITR	3.4692	190
	IPWR	**3.2561**	300

Table 4
Maximum value of R^2 per measure for user-based approach. Best values are highlighted in bold.

Dataset	Measure	Max R^2	Top-K users
MovieLens100k	PCC	0.1890	290
	AMI	0.1187	300
	τ	0.1226	300
	Cosine	0.1859	270
	ρ	0.1224	300
	ARI	0.1195	300
	ACosine	0.1725	300
	J	0.2014	300
	ES	0.0072	300
	MS	-0.0434	300
	Ch	0.1368	300
	ITR	**0.2271**	300
	IPWR	0.2139	300
MovieLens1M	PCC	0.2534	300
	AMI	0.2300	300
	τ	0.2417	300
	Cosine	0.2532	300
	ρ	0.2424	300
	ARI	0.2268	300
	ACosine	0.2194	300
	J	0.2731	300
	ES	-0.1201	300
	MS	-0.1960	300
	Ch	0.1231	300
	ITR	**0.2924**	300
	IPWR	0.2864	300
Jester	PCC	**0.3341**	300
	AMI	0.1453	300
	τ	0.1209	300
	Cosine	0.3182	300
	ρ	0.1192	300
	ARI	0.0813	300
	ACosine	0.3147	300
	J	0.2171	170
	ES	0.2097	90
	MS	0.1645	300
	Ch	0.2595	300
	ITR	0.3043	230
	IPWR	**0.3618**	300

where $\bar{r}$ is the mean of the observed (training) ratings (Eq. 52):

$$\bar{r} = \frac{\sum_{i=1}^{n} r_i}{n} \tag{52}$$

Since a RS is considered as a specific type of Information Retrieval systems, precision and recall metrics have been used by many researchers for the evaluation of recommender systems (Herlocker et al., 2004; Bobadilla et al., 2013). Precision (Eq. 53) is defined as the ratio of relevant recommendations to the total number of recommended items. Recall (Eq. 54) is defined as the ratio of relevant recommendations to the number of relevant items (Polatidis and Georgiadis, 2017; Jalili et al., 2018).

$$Precision = \frac{Correctly\ recommended\ items}{Total\ recommended\ items} \tag{53}$$

$$Recall = \frac{Correctly\ recommended\ items}{Relevant\ items} \tag{54}$$

Precision and recall metrics can be combined into a single metric, called F-score (F_1), as defined in Eq. (55):

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{55}$$

Here, higher values of three metrics (precision, recall and F-score) indicate better performance. These metrics were calculated per user by retrieving k recommendations (Wilson et al., 2014), where k varies from 10 to 50, and calculate their average values across users, corresponding to each k.

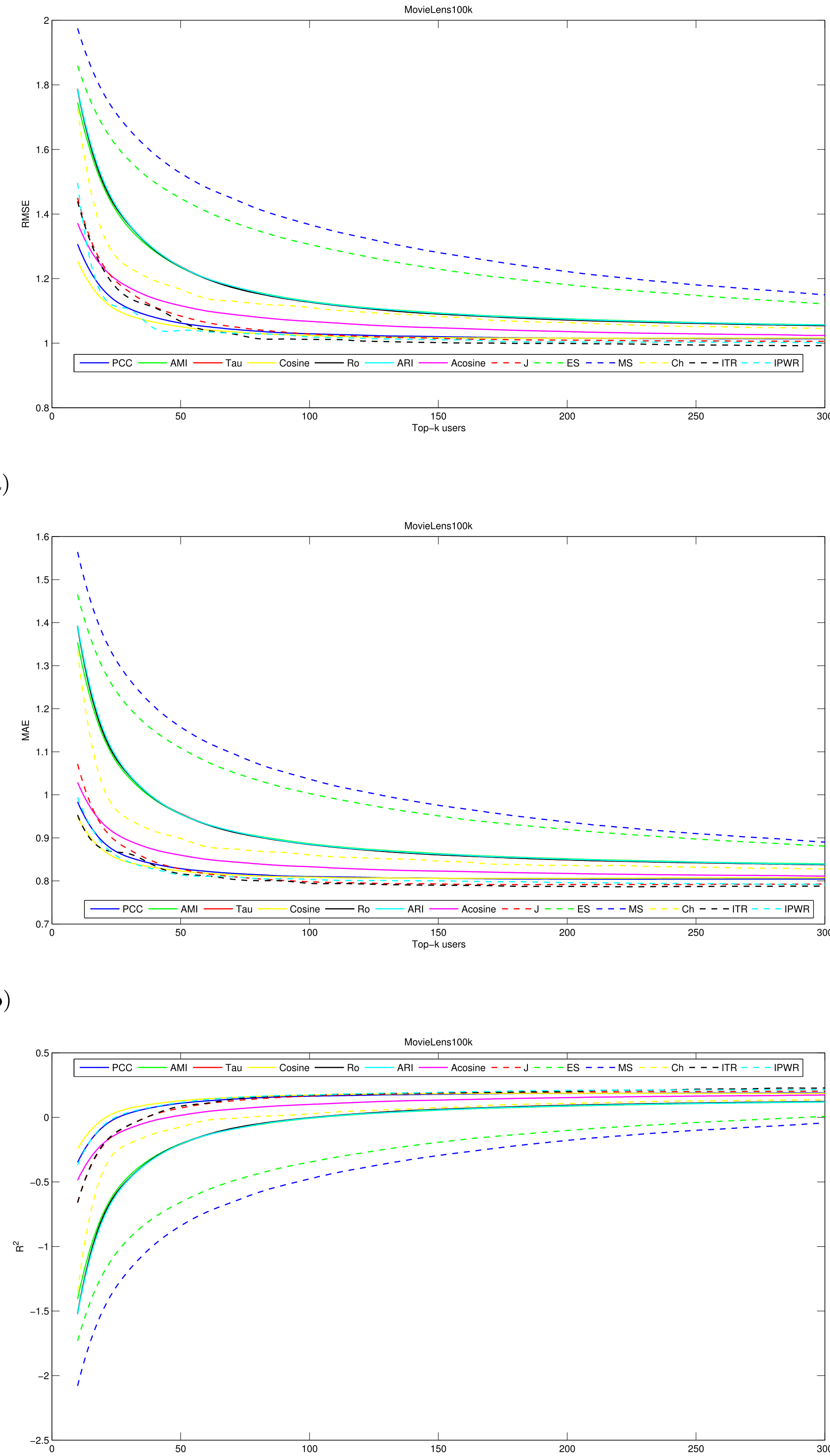
We have to mention that we used Python for the implementation of the evaluation metrics (Pedregosa et al., 2011).

5. Results

In this section we present the obtained findings. Basically, we perform two types of experiments, using two CF techniques: user-based CF and item-based CF. For both techniques, the experimental study was carried out on the same datasets, the same similarity measures and the same prediction formula: the weighted sum method (Eqs. 1 and 3). For each experiment, we start with a set of 10 nearest neighborhood (Top-10 users/items) of the active user/item. Then, we expand each time the set of neighborhood by 10 until 300, except for Jester dataset we stop at 100 for the item-based approach since it is the maximum number of items.

5.1. User-based CF

After applying the user-based CF algorithm, we obtained the results shown in Tables 2–4. In each table, we present the best value reached by each similarity measures in terms of RMSE,



(a)

(b)

(c)

Fig. 2. Evaluation results of the similarity measures using user-based CF algorithm on MovieLens100k. (a) RMSE. (b) MAE. (c) R^2.

MAE and R^2, respectively, with the corresponding top-k users. As shown in Tables 2–4, ITR provides the best result for the 3 evaluation metrics on MovieLens datasets. For instance, ITR gets the best values on MovieLens1M: 0.9428 (RMSE), 0.7315 (MAE) and 0.2924 (R^2). On the other hand, IPWR surpasses all the measures on Jester:

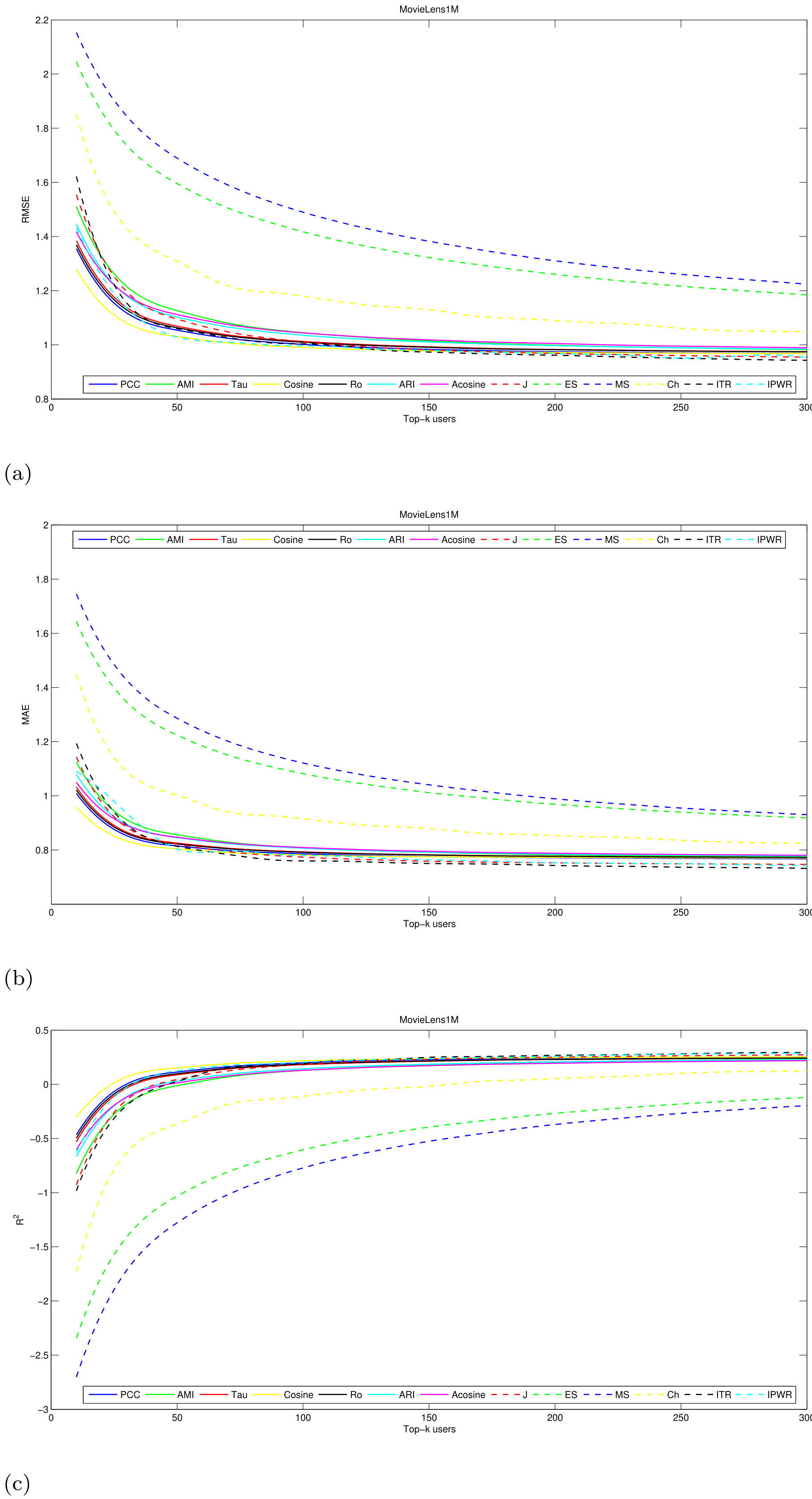
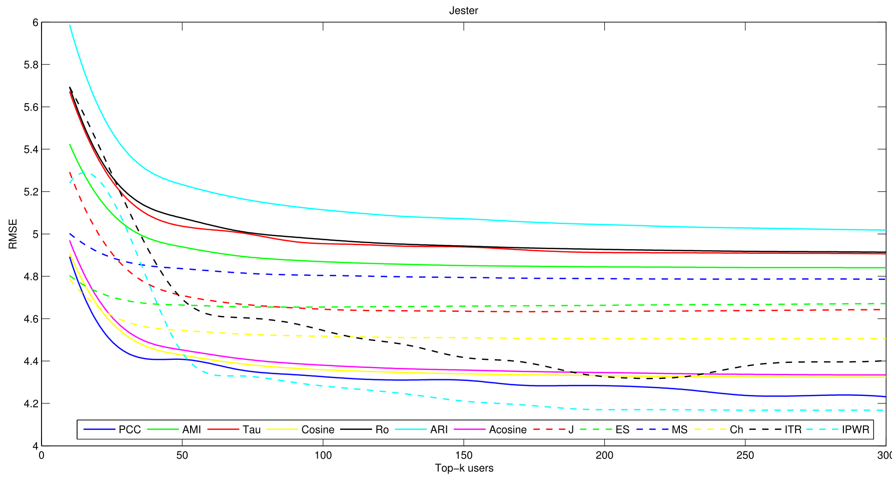
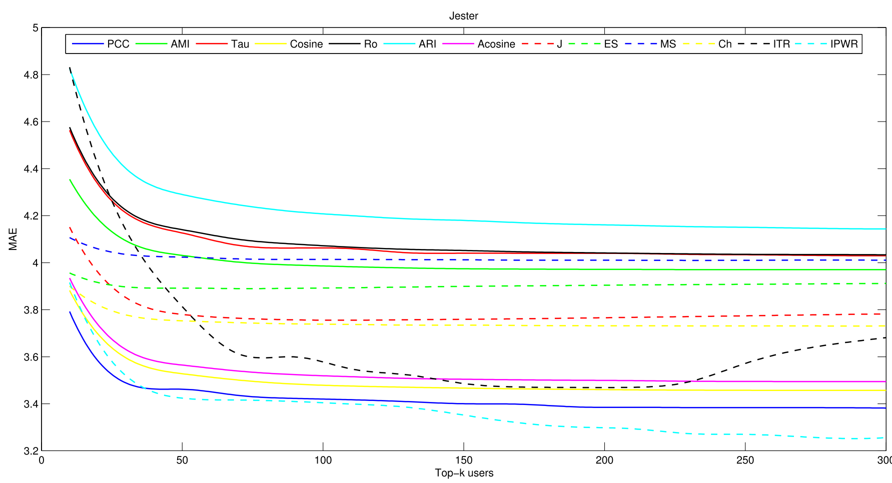


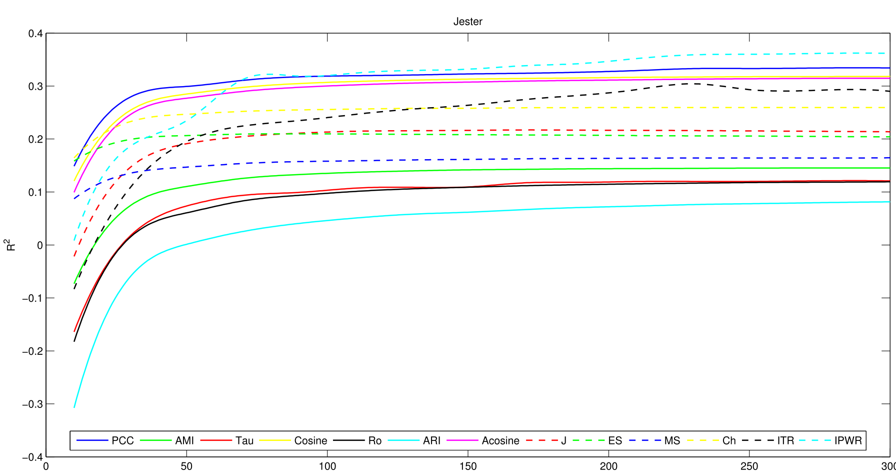
Fig. 3. Evaluation results of the similarity measures using user-based CF algorithm on MovieLens1M. (a) RMSE. (b) MAE. (c) R^2.



(a)



(b)



(c)

Fig. 4. Evaluation results of the similarity measures using user-based CF algorithm on Jester dataset. (a) RMSE. (b) MAE. (c) R^2.

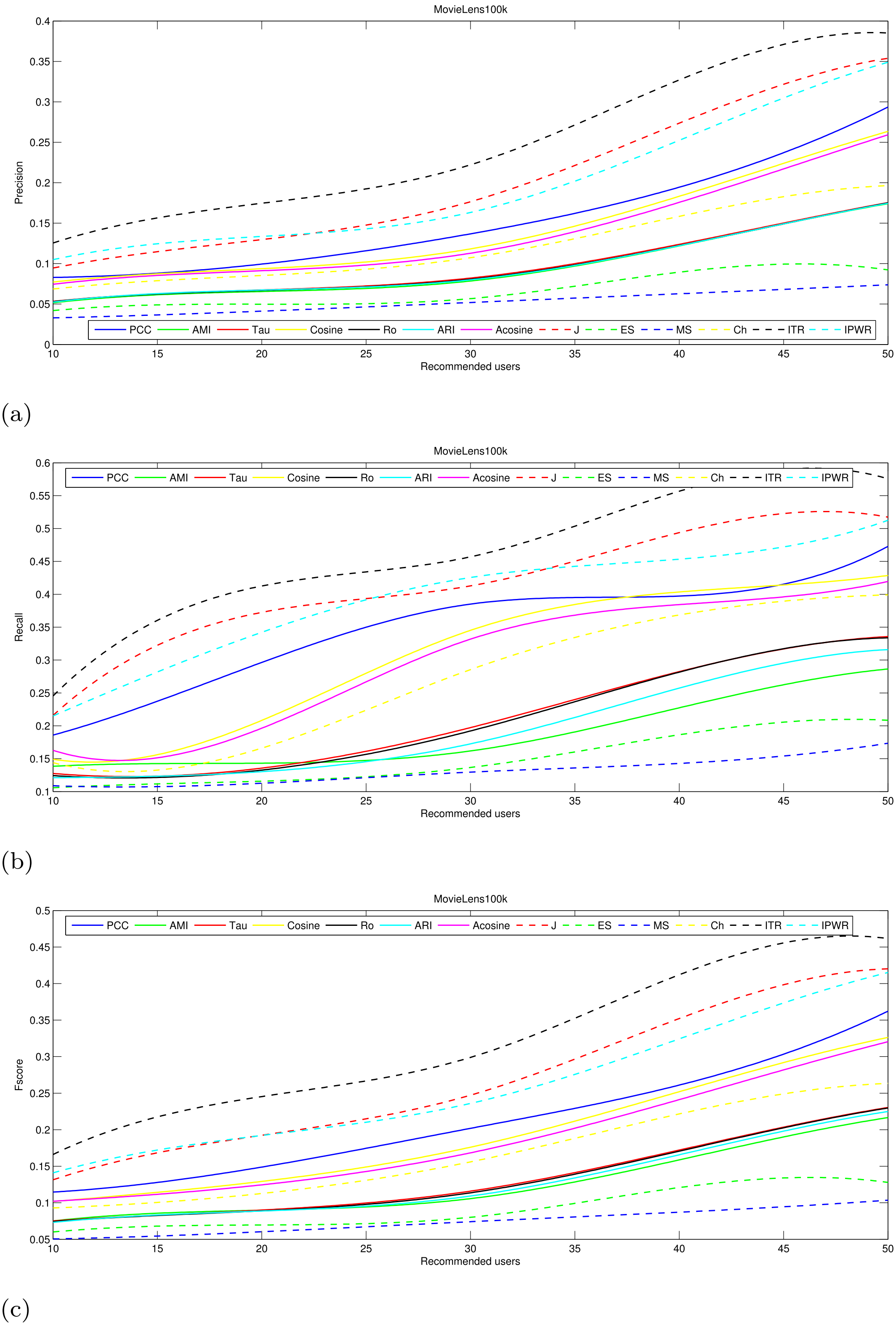
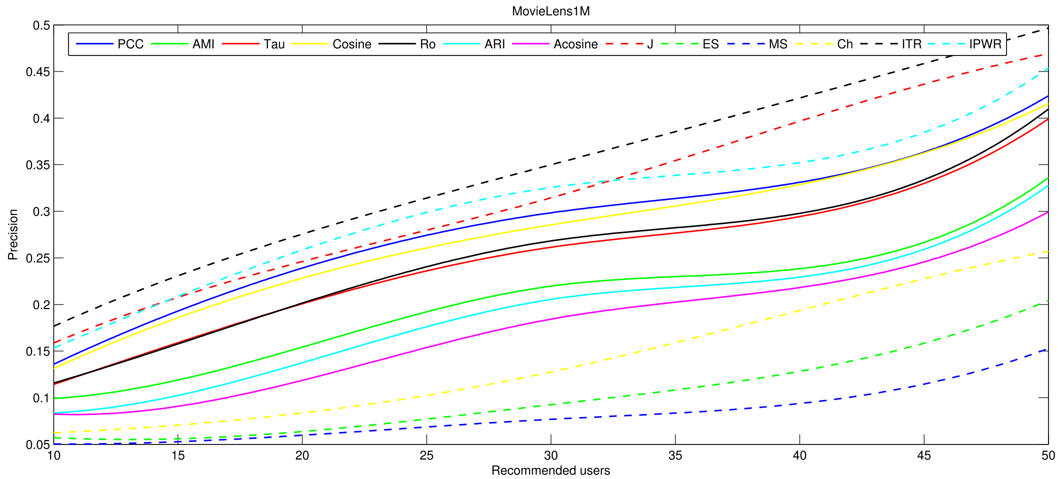


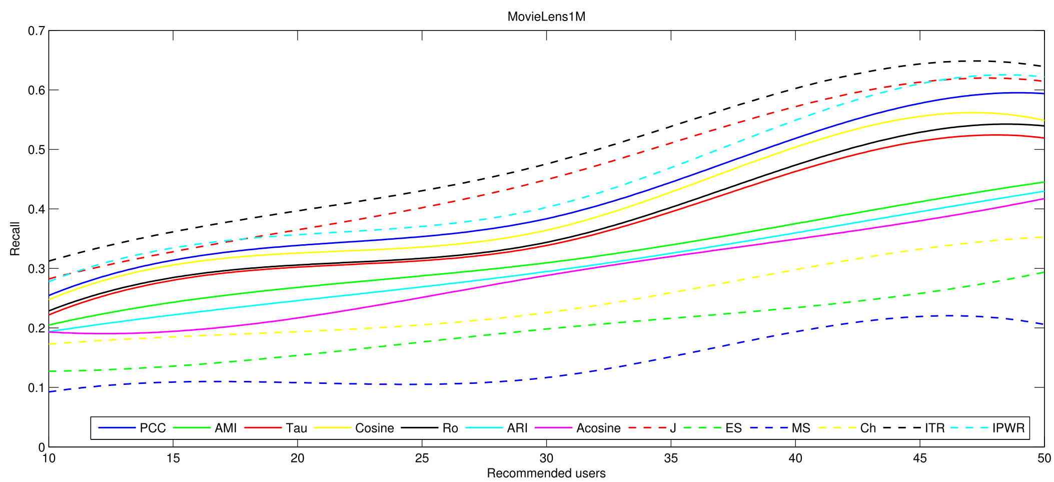
Fig. 5. Evaluation results of the similarity measures using user-based CF algorithm on MovieLens100k. (a) Precision. (b) Recall. (c) F-score.

4.1677 (RMSE), 3.2561 (MAE) and 0.3618 (R^2). Nevertheless, we can observe that the majority of the evaluated measures provide close results, except for Euclidean similarity (ES) and Manhattan similarity (MS) on MovieLens and Adjusted Rand Index (ARI) on Jester.

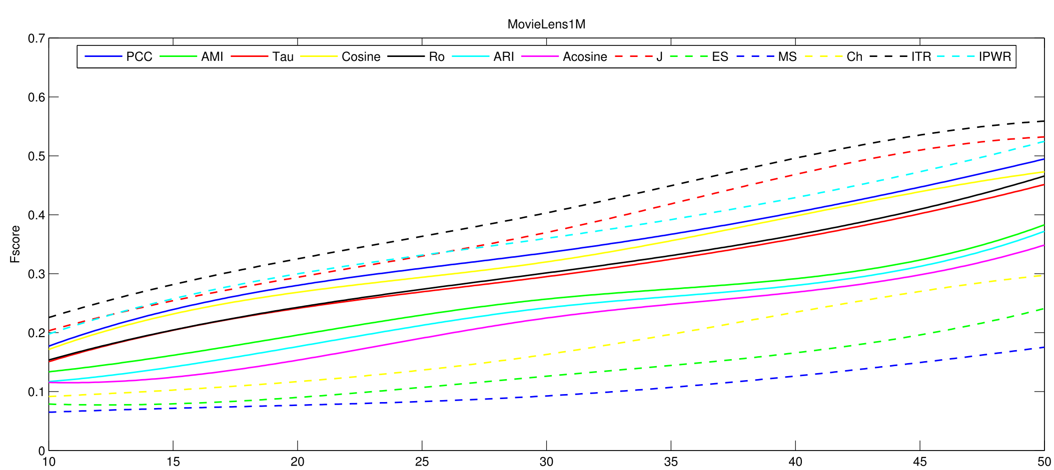
Figs. 2–4 show that all similarity measures have a very similar behavior, especially for the measures ρ, τ that present almost the same curve. Thus, the more users involved in the algorithm, the best the performance of the system would be. This behavior continue until the performance ceases improving or the improvement



(a)



(b)



(c)

Fig. 6. Evaluation results of the similarity measures using user-based CF algorithm on MovieLens1M. (a) Precision. (b) Recall. (c) F-score.

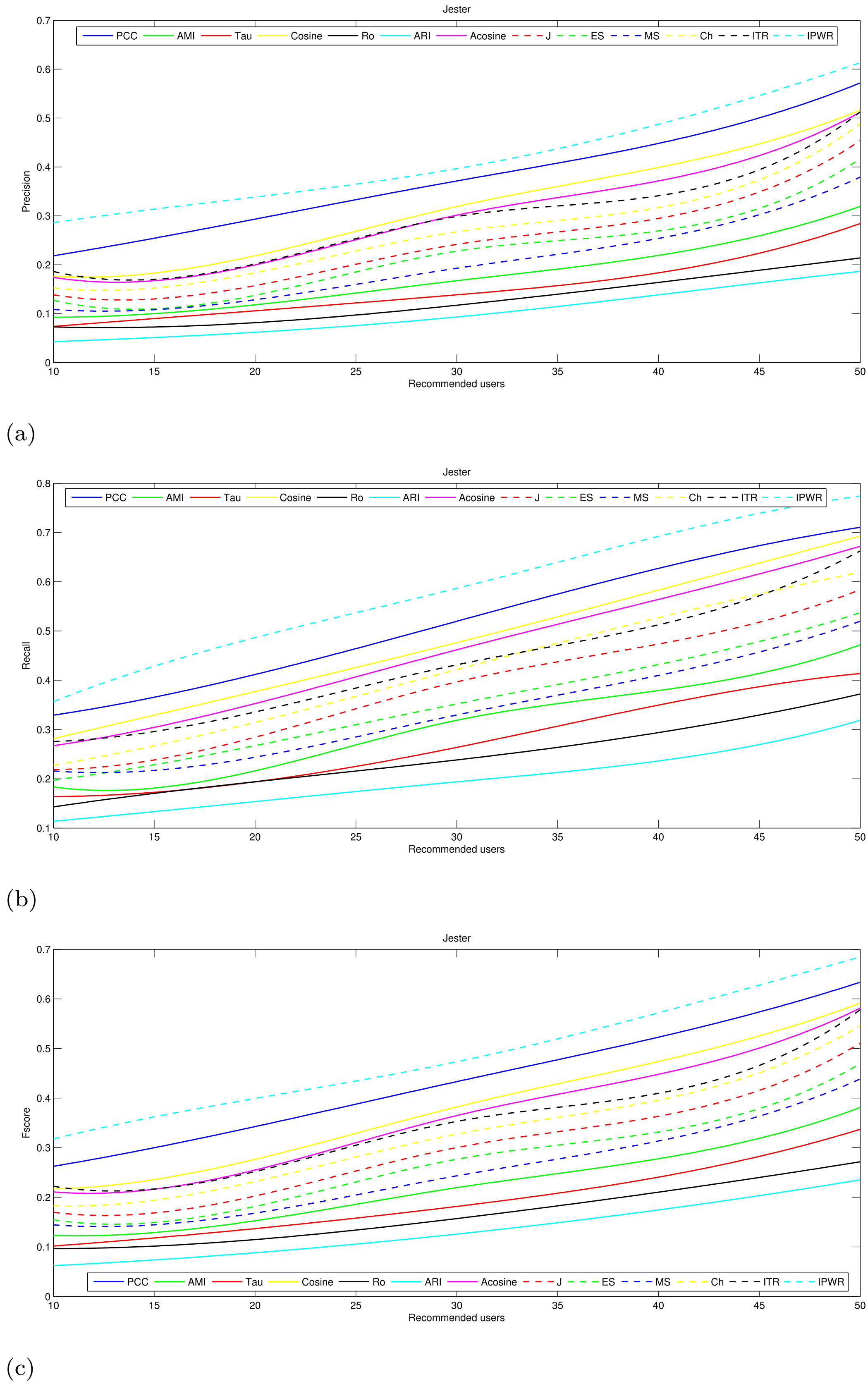


Fig. 7. Evaluation results of the similarity measures using user-based CF algorithm on Jester. (a) Precision. (b) Recall. (c) F-score.

Table 5
Comparison between best results provided by each dataset in terms of NMAE and F-score for user-based approach. Best values are highlighted in bold.

Dataset	MovieLens100k	MovieLens1M	Jester
NMAE	0.1965	0.1828	**0.1628**
F-score	46.1%	55.8%	**68.4%**

Table 6
Minimum value of RMSE per measure for item-based approach. Best values are highlighted in bold.

Dataset	Measure	Min RMSE	Top-K items
MovieLens100k	PCC	0.9610	160
	AMI	**0.9487**	300
	τ	0.9658	180
	Cosine	0.9847	100
	ρ	0.9671	190
	ARI	0.9605	200
	ACosine	1.0877	300
	J	0.9819	300
	ES	1.7688	300
	MS	1.9458	300
	Ch	1.1594	300
	ITR	0.9628	300
	IPWR	0.9513	190
MovieLens1M	PCC	0.9278	230
	AMI	**0.9252**	300
	τ	0.9298	250
	Cosine	0.9515	150
	ρ	0.9304	250
	ARI	0.9285	300
	ACosine	1.1681	300
	J	0.9350	300
	ES	2.0892	300
	MS	2.3765	300
	Ch	1.1305	300
	ITR	0.9327	300
	IPWR	0.9274	250
Jester	PCC	4.3659	30
	AMI	**4.2813**	40
	τ	4.3539	30
	Cosine	4.3000	40
	ρ	4.3653	30
	ARI	5.6192	30
	ACosine	4.3214	40
	J	4.3930	30
	ES	4.7938	40
	MS	4.9798	40
	Ch	4.6621	40
	ITR	4.3159	60
	IPWR	4.3066	50

becomes very slight, and then the performance decreases. Also, we remark that some measures reach their optimum earlier than the others. For instance, Fig. 2c shows that PCC attains its optimum at top-290 users whereas Cosine attains its optimum at top-270 and ITR at top-300. The speed to attain the optimum can be a very important criterion for evaluating Recommender Systems.

Furthermore, the decision-support metrics (precision, recall and F-score) provide similar results as the error metrics (RMSE, MAE and R^2). Similarly, Figs. 5 and 6 show that ITR outperforms the other measures on MovieLens100k and MovieLens1M. Indeed, ITR reaches an F-score equal to 46.1% and 55.8% on MovieLens100k and MovieLens1M, respectively. Concerning Jester dataset, we observe in Fig. 7 that IPWR surpasses all the other similarity measures and attains an F-score equal to 68.4%. Also, Jaccard and IPWR provide a good results on MovieLens100k and MovieLens1M, less than ITR but better than all the other measures. On the other hand, PCC and Cosine have good performances on Jester but modest results on MovieLens100k and MovieLens1M. We mention that Euclidean and Manhattan similarities give the worst results on MovieLens100k and MovieLens1M, ARI and ρ give the worst performances on Jester.

The main conclusion that can be deduced from this first part of the experimental study that ITR measure outscores the other measures if it is applied in the user-based approach, on MovieLens dataset. On the other hand, IPWR provides the best results on Jester dataset which is more dense than MovieLens. Also, Table 5 proves that the result quality is strongly depends on the dataset volume. In fact, the bigger the dataset is, the better the performance of the RS would be.

Table 7
Minimum value of MAE per measure for item-based approach. Best values are highlighted in bold.

Dataset	Measure	Min MAE	Top-K items
MovieLens100k	PCC	0.7523	130
	AMI	**0.7458**	240
	τ	0.7557	130
	Cosine	0.7669	100
	ρ	0.7570	150
	ARI	0.7520	150
	ACosine	0.8117	300
	J	0.7626	300
	ES	1.3039	300
	MS	1.4374	300
	Ch	0.8921	280
	ITR	0.7519	300
	IPWR	0.7589	300
MovieLens1M	PCC	0.7253	170
	AMI	**0.7244**	270
	τ	0.7287	170
	Cosine	0.7365	90
	ρ	0.7288	150
	ARI	0.7297	210
	ACosine	0.8419	300
	J	0.7248	300
	ES	1.5850	300
	MS	1.8386	300
	Ch	0.8802	300
	ITR	0.7246	300
	IPWR	0.7251	210
Jester	PCC	3.3573	20
	AMI	**3.2814**	30
	τ	3.3569	20
	Cosine	3.3057	20
	ρ	3.3657	20
	ARI	4.3469	10
	ACosine	3.3241	30
	J	3.3712	20
	ES	3.5958	60
	MS	4.2200	30
	Ch	3.7191	30
	ITR	3.3317	50
	IPWR	3.3256	40

5.2. Item-based CF

In this section, we show the results provided by the different similarity measures using the item-based CF algorithm on the datasets movieLens100k, movieLens1M and Jester. Best results for each evaluation metric are provided in Tables 6–8. In each table, we provide the best result attained by each metric with its corresponding top-k items. We can observe that item-based algorithm provides results better than those provided by the user-based algorithm for all similarity measures. This result is expected since items contain, in average, more common ratings than users. For example, each user in MovieLens100k rated at least 20 items

Table 8
Maximum value of R^2 per measure for item-based approach. Best values are highlighted in bold.

Dataset	Measure	Max R^2	Top-K items
MovieLens100k	PCC	0.2709	160
	AMI	**0.2893**	300
	τ	0.2634	180
	Cosine	0.2344	110
	ρ	0.2615	190
	ARI	0.2716	200
	ACosine	0.0657	300
	J	0.2386	300
	ES	−1.4720	300
	MS	−1.9909	300
	Ch	−0.0622	300
	ITR	0.2688	300
	IPWR	0.2793	180
MovieLens1M	PCC	0.3127	230
	AMI	**0.3166**	300
	τ	0.3097	250
	Cosine	0.2772	150
	ρ	0.3089	250
	ARI	0.3117	300
	ACosine	−0.0892	300
	J	0.3021	300
	ES	−2.4844	300
	MS	−3.5086	300
	Ch	−0.0202	300
	ITR	0.3093	300
	IPWR	0.3148	190
Jester	PCC	0.3047	30
	AMI	**0.3286**	30
	τ	0.3091	30
	Cosine	0.3255	40
	ρ	0.3049	30
	ARI	−0.1516	30
	ACosine	0.3188	40
	J	0.2960	30
	ES	0.2498	80
	MS	0.2347	90
	Ch	0.2199	60
	ITR	0.3107	40
	IPWR	0.3196	50

while an item might be rated by 1 user. For this reason, the similarity between two items provides a better accuracy than that provided by the similarity between two users (Ning et al., 2015).

Also, we can remark that AMI provides the best results for the three evaluation metrics: RMSE, MAE and R^2 and on the three datasets: MovieLens100k, MovieLens1M and Jester. For example, the result of AMI on Jester is as follows: 4.2813, 3.2814 and 0.3286. However, some similarity measures, like: τ, ρ, RI, ITR and IPWR provide results very close to those provided by AMI. Moreover, These measures reach, in many times, their optimum earlier then AMI. For instance, Table 7 shows that AMI attains its optimum at top-240 items (on MovieLens100k) while τ and ρ attain their optimum at top-130 and top-150, respectively. Nonetheless, we can observe that ES and MS provides the worst results on MovieLens100k and MovieLens1M. These two measures maintain the same weak performance as shown in Section 5.1 for the user-based algorithm. Also, results show that ARI have the lowest performance on Jester dataset.

As shown in Figs. 8–10, the performance achieved by AMI is significantly better than the other similarity measures considered in this study. Actually, AMI starts with a low performance compared to the others, its performance improves with increasing the number of items involved in the algorithm until reaching its maximum at a top-k items between 100 and 150 for MovieLens dataset and between 30 and 40 for Jester dataset. AMI overpowers the other similarity measures in all evaluation metrics.

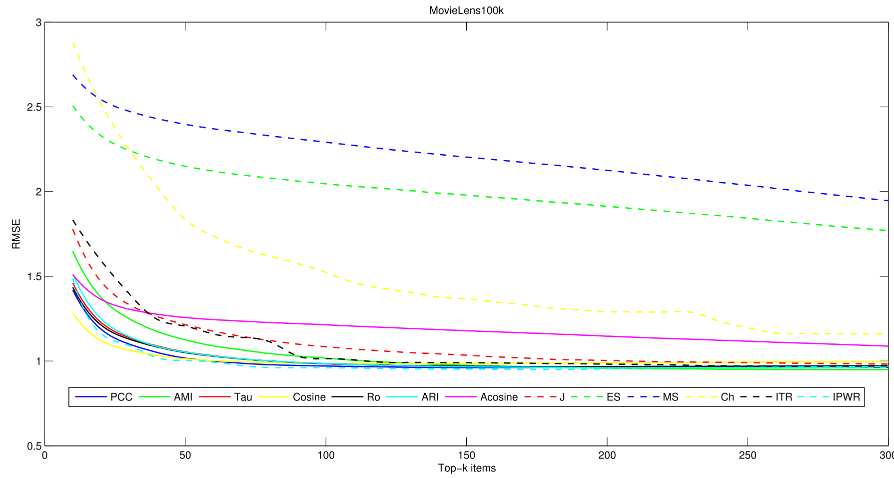
Figs. 11–13 present the performances provided by each measure using the decision-support metrics (precision, recall and F-score). It is obviously clear that AMI gives the best precision, recall and F-score over all the datasets. For example, the results obtained by AMI on Jester are es follows: 59.3% (precision), 71.3% (recall) and 64.8% (F-score). Also, we can observe that many other similarity measures provide good results, such as, ρ, τ, PCC, IPWR and ITR. However, Euclidean and Manhattan similarities provide the worst results on MovieLens100k and MovieLens1M, and ARI gives the worst performance on Jester.

The second part of the experimental study prove clearly that the AMI measure is the best choice for the item-based CF approach. In practice, AMI outperforms the other measures on all datasets. This result leads to an important observation: it doesn't necessary that the same similarity measure provides the best result in both approaches (user-based and item-based approaches). Also, Table 9 confirms the remark already deduced in the user-based approach: results get better with the increase of the volume of the dataset.

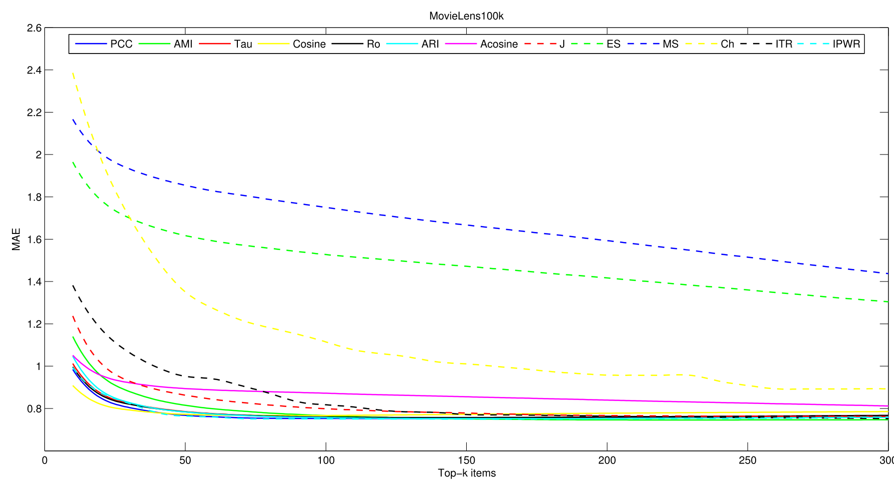
6. Discussion

The comparative study presented in this paper reveals many important clues related to the similarity measures and their relation with the RS performance. In fact, this experimental study prove that the performance of a given RS is strongly dependent on the used similarity measure. As shown in previous sections, a bad choice of a similarity measure will surely lead to a bad performance and vice versa. Also, we can deduce that there is no any specific similarity measure suitable for both user-based and item-based approaches and some measures perform better in an approach than the other. Another important conclusion can be inferred from this work that the similarity measure's performance depends, among other factors, on the dataset density. Actually, the similarity measure's performance can vary accordingly to the dataset density even under the same filtering approach. Furthermore, we can conclude that recent measure like ITR and IPWR perform better than classic measures under the user-based approach. This result can be explained with the fact that these measures improved existing classic measures by integrating semantic knowledge about the user (user rating preferences behavior). Nevertheless, many classic measure can provide good result such as Jaccard, PCC and AMI.

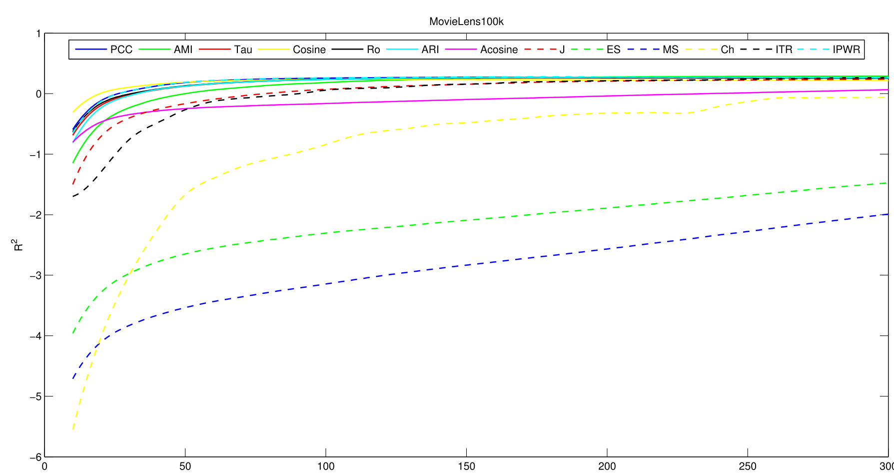
Moreover, this study demonstrates that similarity measure choice is not the only challenge for a CF based RS. In fact, CF approach is suffering from many problems that affect negatively its performance as, for example, the sparsity of the data and the cold start situation. Sparsity problem occurs when the number of ratings is small compared with the number of items and users (Natarajan et al., 2020). Also, cold start issue happens when the system handle a new user/item without any previous history: the user didn't rate any item before or the item wasn't rated before (Guo, 2012). In this case, the available information is not enough to establish a clear profile about the user and to identify with accuracy his similar neighbors. Consequently, similarity measures are highly sensitive to these problems since their performances depend, mainly, on the quality of the used data. Therefore, the more the data is rich and dense, the best the performance of the similarity measures. This conclusion can be easily deduced from Tables 9 and 5 which show that the best NMAE value depends on the dataset size and density. If we sort datasets according to their NMAE, we obtain the following order: Jester (with a significant superiority), MovieLens100k and MovieLens1M, which is the same order of their volume. Therefore, results obtained by MovieLens100k and MovieLens1M are very close since they have both a low density. Indeed, the good results obtained by Jester



(a)

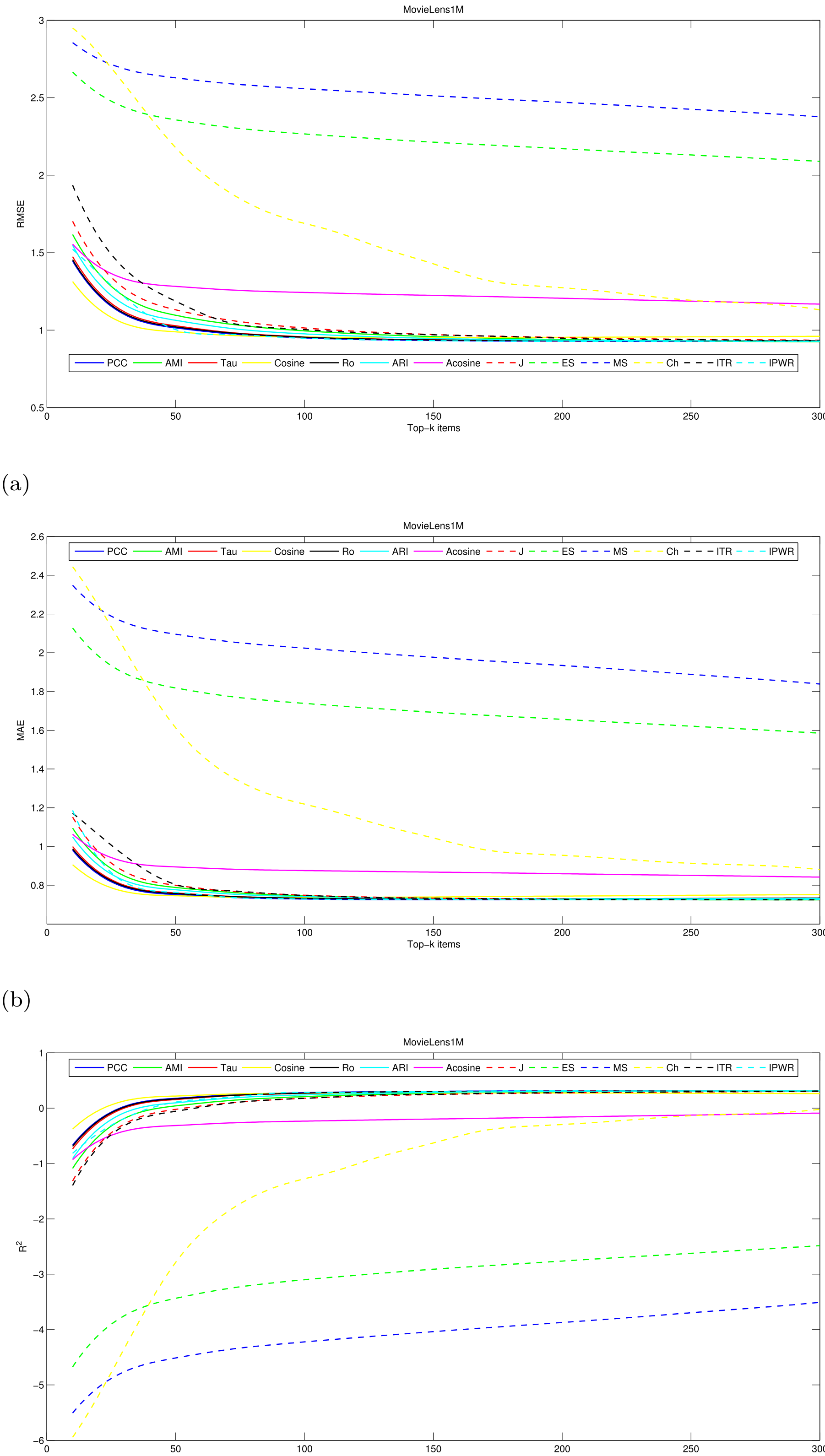


(b)



(c)

Fig. 8. Evaluation results of the similarity measures using item-based CF algorithm on MovieLens100k. (a) RMSE. (b) MAE. (c) R^2.



(a)

(b)

(c)

Fig. 9. Evaluation results of the similarity measures using item-based CF algorithm on MovieLens1M. (a) RMSE. (b) MAE. (c) R^2.

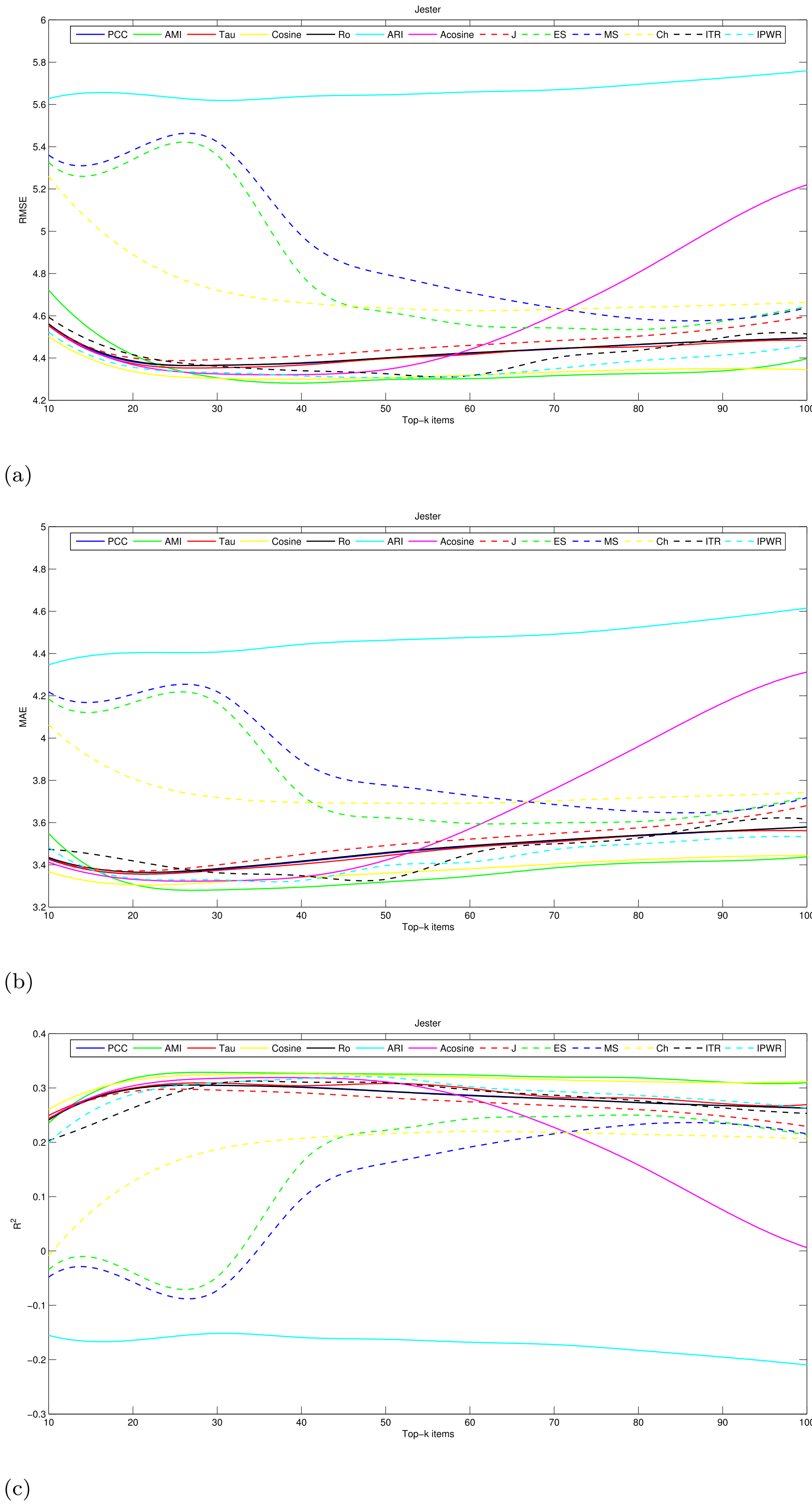
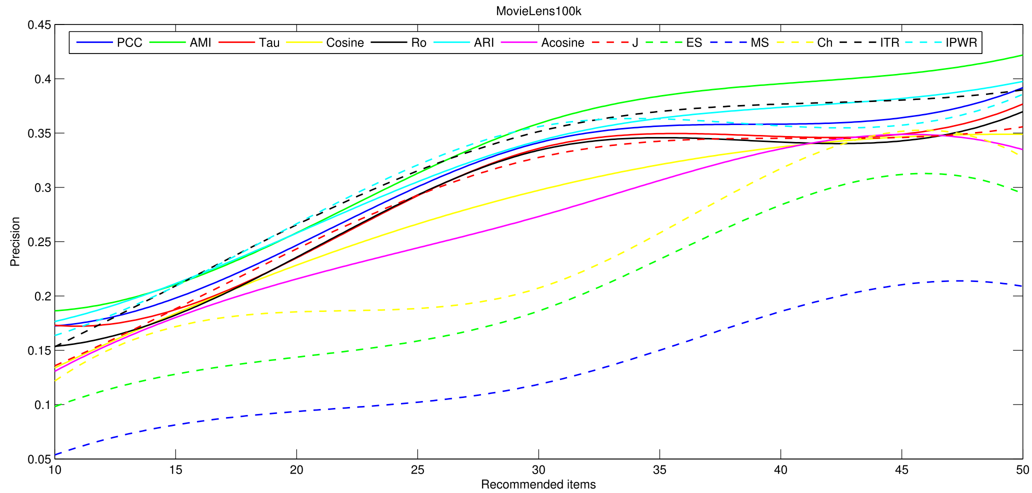
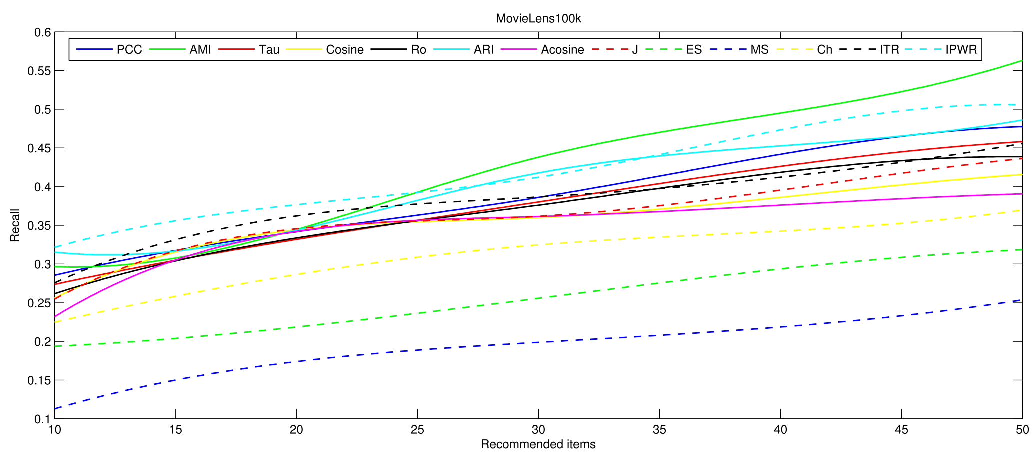


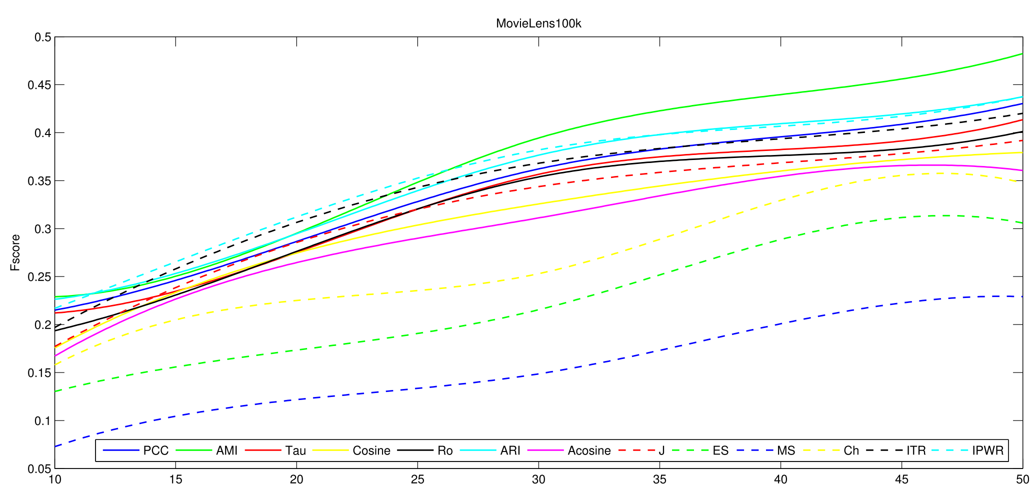
Fig. 10. Evaluation results of the similarity measures using item-based CF algorithm on Jester dataset. (a) RMSE. (b) MAE. (c) R^2.



(a)

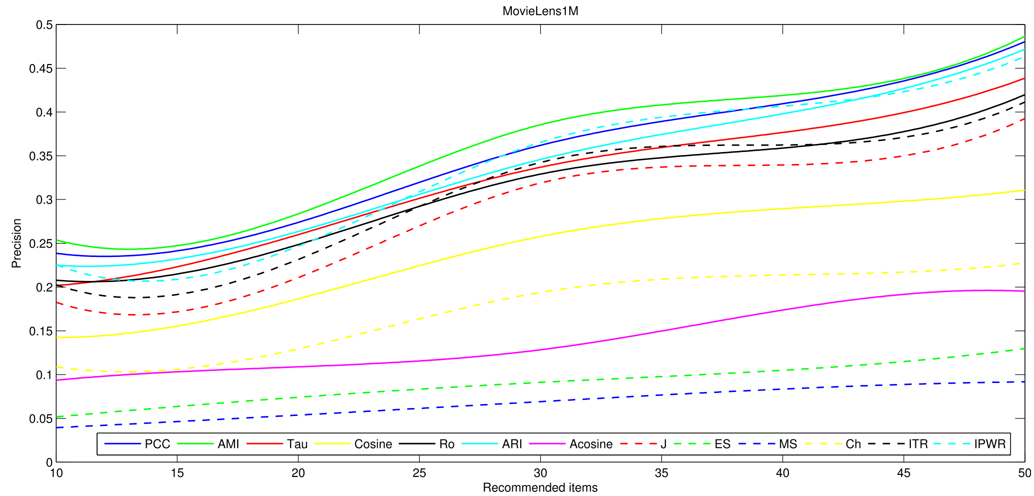


(b)

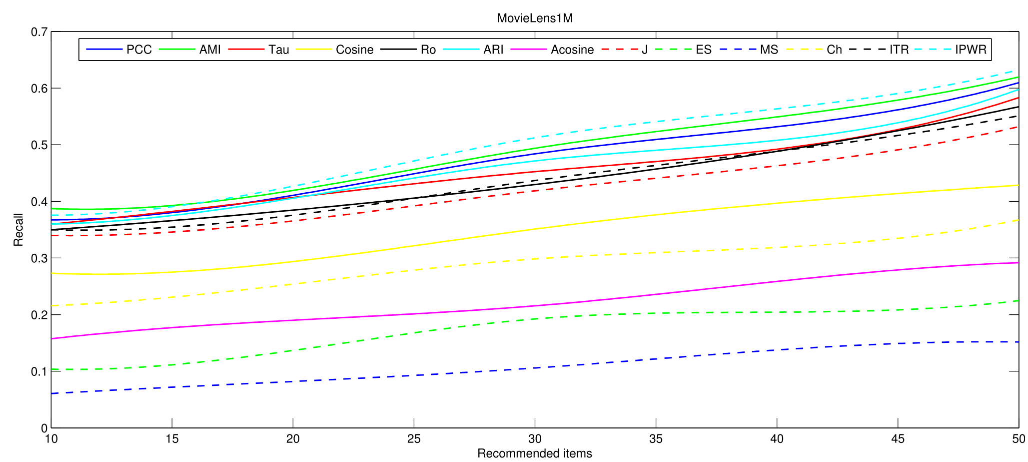


(c)

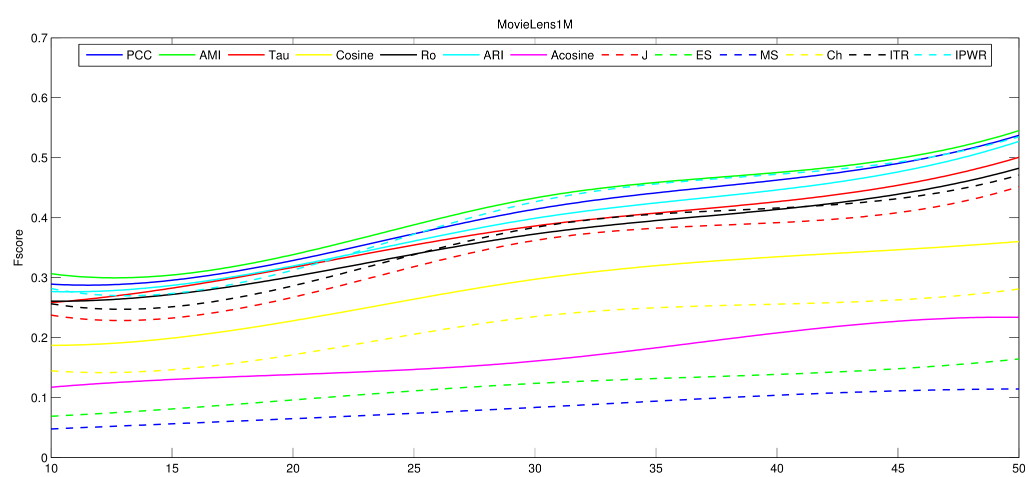
Fig. 11. Evaluation results of the similarity measures using item-based CF algorithm on MovieLens100k. (a) Precision. (b) Recall. (c) F-score.



(a)

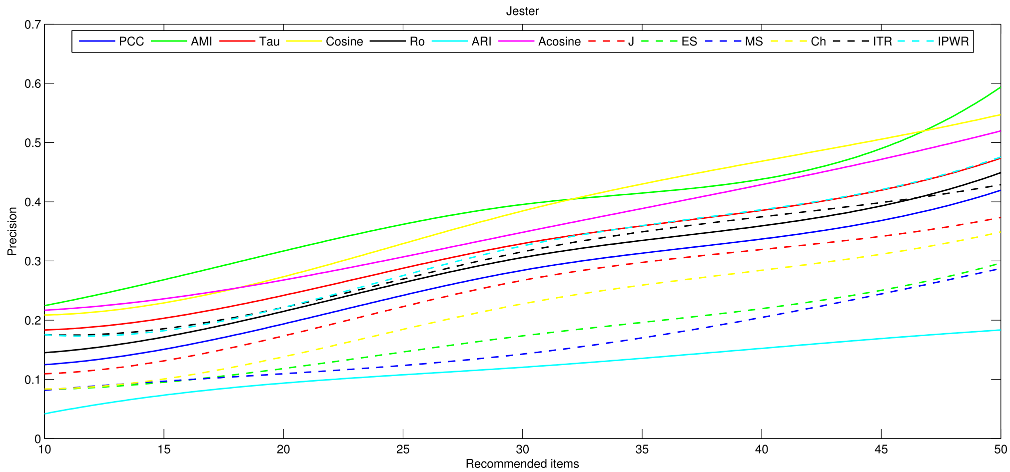


(b)

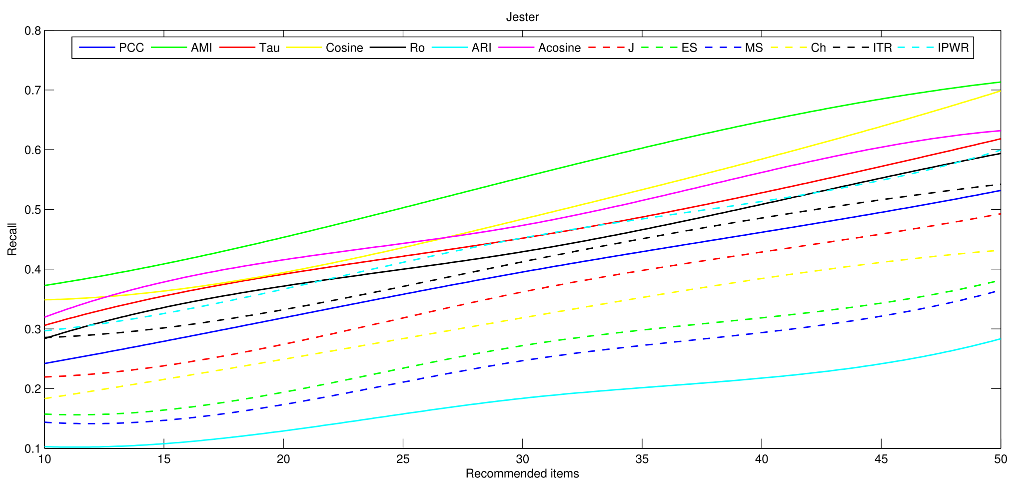


(c)

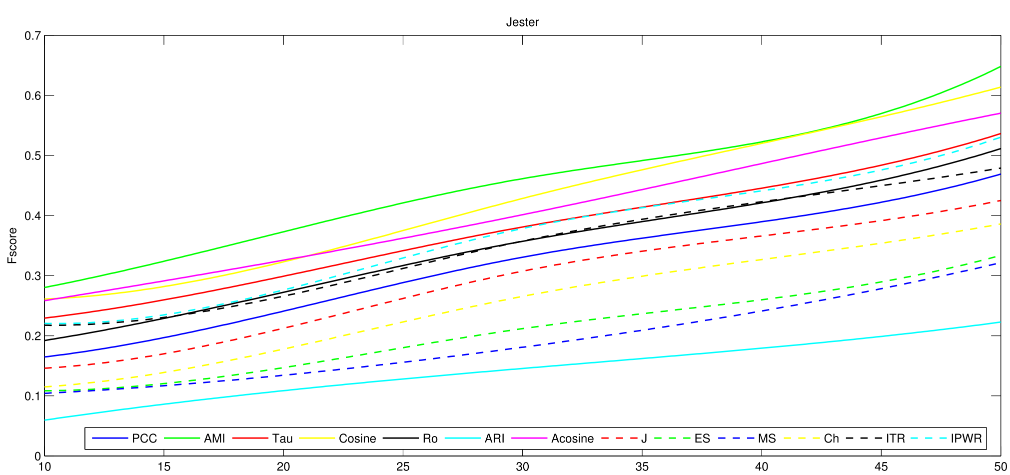
Fig. 12. Evaluation results of the similarity measures using item-based CF algorithm on MovieLens1M. (a) Precision. (b) Recall. (c) F-score.



(a)



(b)



(c)

Fig. 13. Evaluation results of the similarity measures using item-based CF algorithm on Jester. (a) Precision. (b) Recall. (c) F-score.

dataset compared with ones obtained by MovieLens100k and MovieLens1M can be explained, firstly, by its high density (low sparsity) and secondly by its few number of items (100) which decrease the chance of facing the cold start problem. In the literature of the CF-based Recommender System domain, many researches were focused on improving the existing measures or

Table 9
Comparison between best results provided by each dataset in terms of NMAE and F-score for item-based approach. Best values are highlighted in bold.

Dataset	MovieLens100k	MovieLens1M	Jester
NMAE	0.1864	0.1811	**0.1640**
F-score	48.2%	54.5%	**64.8%**

inventing new ones more efficient without thinking about the data and its impact on the measures' performances.

Furthermore, we think that integrating semantic knowledge (Fkih and Omri, 2012, 2016) in the data will surely improve the results provided by the similarity measures and, consequently, the performance of the Recommender System. For example, a user that expresses in a tweet (or in a picture) his attachment to combat sport and auto racing can be, probably, a fan of 'Fast and Furious' movies. This information might be extracted and analyzed using text mining or image processing techniques and then integrated in the data, especially with the significant evolution in such tools (Fkih and Omri, 2013b). Moreover, many supplementary information (like gender, age, etc.) can be also automatically extracted from users' profiles using adequate techniques (Ouni et al., 2021) which would ameliorate the quality of the recommendation process. As well, many other semantic knowledge can be integrated, such as, social trust (implicit and explicit trust) because people tend to respond positively to recommendations derived from their social trustworthy friends (Ayub et al., 2020).

7. Conclusion

In this work, we conducted an in-depth comparative study of thirteen similarity measures commonly used in the field of CF-based Recommender System. Consequently, we carried out our experiments on three standards datasets: MovieLens100k, MovieLens1M and Jester. Also, we divided the experimental study into two phases: user-based and item-based techniques. For each CF technique, we used a set of well-known evaluation metrics to skill the RS using, each time, a different similarity measure. Practically, similarity measures will be classified and ranked according to the performance of the Recommender System. Therefore, the measure that offers to the system the best performance will be highlighted.

Findings showed that user-based and item-based CF don't share the same similarity measure that offers the best performance to the system. Indeed, Improved Triangle similarity complemented with user rating preferences (ITR) and Improved PCC weighted with RPB (IPWR) are the best choices for a user-based Recommender System, whereas Adjusted Mutual Information (AMI) can offer to an item-based RS the best performance.

Declaration of Competing Interest

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

Acknowledgments

*

References

Abello, J., Pardalos, P.M., Resende, M.G.C. (Eds.), 2002. Handbook of Massive Data Sets. Kluwer Academic Publishers, USA.

Abramowicz, W., 2003. Knowledge-Based Information Retrieval and Filtering from the Web. Springer, Boston, MA. doi:10.1007/978-1-4757-3739-4.

Adomavicius, G., Mobasher, B., Ricci, F., Tuzhilin, A., 2011. Context-aware recommender systems. AI Magazine 32 (3), 67–80. https://doi.org/10.1609/aimag.v32i3.2364.

Aggarwal, C.C., 2016. Recommender Systems: The Textbook. Springer Publishing Company Incorporated.

Ahn, H.J., 2008. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Information Sciences 178 (1), 37–51. https://doi.org/10.1016/j.ins.2007.07.024.

Ayub, R., Ghazanfar, M. a., Mehmood, Z., Saba, T., Alharbey, R., Munshi, A., Alrige, M., 2019. Modeling user rating preference behavior to improve the performance of the collaborative filtering based recommender systems. PLoS ONE 14. doi:10.1371/journal.pone.0220129.

Ayub, M., Ghazanfar, M.A., Mehmood, Z., Alyoubi, K.H., Alfakeeh, A.S., 2020. Unifying user similarity and social trust to generate powerful recommendations for smart cities using collaborating filtering-based recommender systems. Soft Computing 24, 11071–11094.

Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A., 2013. Recommender systems survey. Knowledge-Based Systems 46, 109–132. https://doi.org/10.1016/j.knosys.2013.03.012.

Breese, J.S., Heckerman, D., Kadie, C., 1998. Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, UAI'98. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 43–52.

Brun, A., Castagnos, S., Boyer, A., 2009. A positively directed mutual information measure for collaborative filtering. In: Ghenima, M., Sidhom, S., Ouksel, A. (Eds.), 2nd International Conference on Information Systems and Economic Intelligence - SIIE 2009, Malek Ghenima (ESCE Université la Manouba - Tunisie) and Sahbi Sidhom (Nancy Université - France). IHE éditions Tunis - Tunisie, Hammamet, Tunisia, pp. 943–958.

Cacheda, F., Carneiro, V., Fernández, D., Formoso, V. Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems, ACM Trans. Web 5 (1). doi:10.1145/1921591.1921593.

Chen, R., Hua, Q., Chang, Y., Wang, B., Zhang, L., Kong, X., 2018. A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks. IEEE Access 6, 64301–64320. https://doi.org/10.1109/ACCESS.2018.2877208.

Colin Cameron, A., Windmeijer, F.A., 1997. An r-squared measure of goodness of fit for some common nonlinear regression models. Journal of Econometrics 77 (2), 329–342. https://doi.org/10.1016/S0304-4076(96)01818-0.

Conover, W., 1971. Practical Nonparametric Statistics. Wiley.

de Gemmis, M., Lops, P., Musto, C., Narducci, F., Semeraro, G., 2015. Semantics-Aware Content-Based Recommender Systems. Springer US, Boston, MA, pp. 119–159. https://doi.org/10.1007/978-1-4899-7637-6_4.

Ekstrand, M.D., Riedl, J.T., Konstan, J.A., 2011. Collaborative filtering recommender systems. Found. Trends Hum.-Comput. Interact. 4 (2), 81–173. doi:10.1561/1100000009.

Fkih, F., Omri, M.N., 2012. Information retrieval from unstructured web text document based on automatic learning of the threshold. Int. J. Inf. Retr. Res. 2 (4), 12–30. https://doi.org/10.4018/ijirr.2012100102.

Fkih, F., Omri, M.N., 2013a. Estimation of a priori decision threshold for collocations extraction: An empirical study. Int. J. Inf. Technol. Web Eng. 8 (3), 34–49. https://doi.org/10.4018/ijitwe.2013070103.

Fkih, F., Omri, M.N., 2013. A statistical classifier based Markov chain for complex terms filtration, in: Proceedings of the International Conference on Web Informations and Technologies, ICWIT 2013, Hammamet, Tunisia, pp. 175–184..

Fkih, F., Omri, M.N., 2016. Hybridization of an index based on concept lattice with a terminology extraction model for semantic information retrieval guided by wordnet. In: Abraham, A., Haqiq, A., Alimi, A.M., Mezzour, G., Rokbani, N., Muda, A.K. (Eds.), Proceedings of the 16th International Conference on Hybrid Intelligent Systems (HIS 2016), Marrakech, Morocco, November 21–23, 2016, Vol. 552 of Advances in Intelligent Systems and Computing, Springer, pp. 144–152. doi:10.1007/978-3-319-52941-7_15.

Fkih, F., Omri, M.N., 2018. Fca_retrieval: A multi-operator algorithm for information retrieval from binary concept lattice. In: S. Politzer-Ahles, Y. Hsu, C. Huang, Y. Yao (Eds.), Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation, PACLIC 2018, Hong Kong, December 1–3, 2018, Association for Computational Linguistics.

Fkih, F., Omri, M.N., 2020. Hidden data states-based complex terminology extraction from textual web data model. Appl. Intell. 50 (6), 1813–1831. https://doi.org/10.1007/s10489-019-01568-4.

Goldberg, K., Roeder, T., Gupta, D., Perkins, C., 2001. Eigentaste: A constant time collaborative filtering algorithm. Inf. Retr. 4 (2), 133–151. https://doi.org/10.1023/A:1011419012209.

Guo, G., 2012. Resolving data sparsity and cold start in recommender systems. In: Masthoff, J., Mobasher, B., Desmarais, M.C., Nkambou, R. (Eds.), User Modeling, Adaptation, and Personalization. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 361–364.

Harper, F.M., Konstan, J.A. The movielens datasets: History and context, ACM Trans. Interact. Intell. Syst. 5 (4). doi:10.1145/2827872.

Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T., 2004. Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. 22 (1), 5–53. https://doi.org/10.1145/963770.963772.

Iftikhar, A., Ghazanfar, M.A., Ayub, M., Mehmood, Z., Maqsood, M., 2020. An improved product recommendation method for collaborative filtering. IEEE Access 8, 123841–123857. https://doi.org/10.1109/ACCESS.2020.3005953.

Iovine, A., Narducci, F., Semeraro, G., 2020. Conversational recommender systems and natural language: A study through the converse framework. Decision Support Systems 131,. https://doi.org/10.1016/j.dss.2020.113250 113250.

Isinkaye, F., Folajimi, Y., Ojokoh, B., 2015. Recommendation systems: Principles, methods and evaluation. Egyptian Informatics Journal 16 (3), 261–273. https://doi.org/10.1016/j.eij.2015.06.005.

Jaccard, P., 1912. The distribution of the flora in the alpine zone. New Phytologist 11 (2), 37–50.

Jain, D.K., Kumar, A., Sharma, V., 2020. Tweet recommender model using adaptive neuro-fuzzy inference system. Future Generation Computer Systems 112, 996–1009. https://doi.org/10.1016/j.future.2020.04.001.

Jalili, M., Ahmadian, S., Izadi, M., Moradi, P., Salehi, M., 2018. Evaluating collaborative filtering recommender algorithms: A survey. IEEE Access 6, 74003–74024. https://doi.org/10.1109/ACCESS.2018.2883742.

Kendall, M.G., 1938. A new measure of rank correlation. Biometrika 30 (1–2), 81–93. https://doi.org/10.1093/biomet/30.1-2.81.

Kendall, M., Gibbons, J.D., 1990. Rank Correlation Methods, 5th Edition, A Charles Griffin Title.

Koh, E.T., Owen, W.L., 2000. Nonparametric Statistics, Springer US, Boston, MA, pp. 155–168. doi:10.1007/978-1-4615-1401-5_9.

McCarey, F., Cinneide, M.O., Kushmerick, N., 2006. A recommender agent for software libraries: An evaluation of memory-based and model-based collaborative filtering. In: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT '06, IEEE Computer Society, USA, pp. 154–162. doi:10.1109/IAT.2006.23.

Najafabadi, M.K., Mahrin, M.N., Chuprat, S., Sarkan, H.M., 2017. Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data. Computers in Human Behavior 67, 113–128. https://doi.org/10.1016/j.chb.2016.11.010.

Natarajan, S., Vairavasundaram, S., Natarajan, S., Gandomi, A.H., 2020. Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data. Expert Systems with Applications 149,. https://doi.org/10.1016/j.eswa.2020.113248 113248.

Neapolitan, R.E., Jiang, X., 2007. Chapter 11 - collaborative filtering. In: Neapolitan, R.E., Jiang, X. (Eds.), Probabilistic Methods for Financial and Marketing Informatics. Morgan Kaufmann, Burlington, pp. 373–385. https://doi.org/10.1016/B978-012370477-1.50028-1.

Ning, X., Desrosiers, C., Karypis, G., 2015. A Comprehensive Survey of Neighborhood-Based Recommendation Methods. Springer US, Boston, MA, pp. 37–76. https://doi.org/10.1007/978-1-4899-7637-6_2.

O'Neill, B., 2006. Chapter 2 - frame fields, in: B. O'Neill (Ed.), Elementary Differential Geometry (Second Edition), second edition Edition, Academic Press, Boston, pp. 43–99. doi:https://doi.org/10.1016/B978-0-12-088735-4.50006-7.

Ouni, S., Fkih, F., Omri, M.N., 2021. Toward a new approach to author profiling based on the extraction of statistical features. Soc. Netw. Anal. Min. 59 (11). doi:10.1007/s13278-021-00768-6.

Pearson, K., 1895. Note on Regression and Inheritance in the Case of Two Parents. Proceedings of the Royal Society of London Series I (58), 240–242.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.

Polatidis, N., Georgiadis, C.K., 2017. A dynamic multi-level collaborative filtering method for improved recommendations. Computer Standards & Interfaces 51, 14–21. https://doi.org/10.1016/j.csi.2016.10.014.

Quijano-Sánchez, L., Cantador, I., Cortés-Cediel, M.E., Gil, O., 2020. Recommender systems for smart cities. Information Systems 92,. https://doi.org/10.1016/j.is.2020.101545 101545.

Rand, W., 1971. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66 (336), 846–850.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J., 1994. Grouplens: An open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW '94, Association for Computing Machinery, New York, NY, USA, pp. 175–186. doi:10.1145/192844.192905.

Ricci, F., Rokach, L., Shapira, B., Kantor, P.B., 2010. Recommender Systems Handbook. Springer-Verlag, Berlin, Heidelberg.

Ritter, A., Muñoz-Carpena, R., 2013. Performance evaluation of hydrological models: Statistical significance for reducing subjectivity in goodness-of-fit assessments. Journal of Hydrology 480, 33–45. https://doi.org/10.1016/j.jhydrol.2012.12.004.

Sarwar, B., Karypis, G., Konstan, J., Riedl, J., 2001. Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, WWW '01, Association for Computing Machinery, New York, NY, USA, pp. 285–295. doi:10.1145/371920.372071.

Shannon, C.E., 2001. A mathematical theory of communication. SIGMOBILE Mob. Comput. Commun. Rev. 5 (1), 3–55. https://doi.org/10.1145/584091.584093.

Shardanand, U., Maes, P., 1995. Social information filtering: Algorithms for automating "word of mouth, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '95, ACM Press/Addison-Wesley Publishing Co., USA, pp. 210–217. doi:10.1145/223904.223931.

Silveira, T., Zhang, M., Lin, X., Liu, Y., Ma, S., 2019. How good your recommender system is? a survey on evaluations in recommendation. International Journal of Machine Learning and Cybernetics 10, 813–831.

Sinnott, R., Duan, H., Sun, Y., 2016. Chapter 15 - a case study in big data analytics: Exploring twitter sentiment analysis and the weather. In: R. Buyya, R.N. Calheiros, A.V. Dastjerdi (Eds.), Big Data, Morgan Kaufmann, pp. 357–388. doi: https://doi.org/10.1016/B978-0-12-805394-2.00015-5.

Spearman, C., 2010. The proof and measurement of association between two things. International Journal of Epidemiology 39 (5), 1137–1150. https://doi.org/10.1093/ije/dyq191.

Sun, S.-B., Zhang, Z.-H., Dong, X.-L., Zhang, H.-R., Li, T.-J., Zhang, L., Min, F. Integrating triangle and jaccard similarities for recommendation. PLoS ONE 12 (8).

Szabo, F.E., 2015. M, in: F.E. Szabo (Ed.), The Linear Algebra Survival Guide, Academic Press, Boston, pp. 219–233. doi:https://doi.org/10.1016/B978-0-12-409520-5.50020-5.

Szczepanska, A., 2011. Research design and statistical analysis, third edition by jerome l. myers, arnold d. well, robert f. lorch, jr. International Statistical Review 79 (3), 491–492. doi: 10.1111/j.1751-5823.2011.00159_12.x.

Turner, J., Baker, R., Kellner, F., 2018. Theoretical literature review: Tracing the life cycle of a theory and its verified and falsified statements. Human Resource Development Review 17, 34–61.

Valcarce, D., Landin, A., Parapar, J., Barreiro, Álvaro, 2019. Collaborative filtering embeddings for memory-based recommender systems. Engineering Applications of Artificial Intelligence 85, 347–356. https://doi.org/10.1016/j.engappai.2019.06.020.

Vinh, N.X., Epps, J., Bailey, J., 2009. Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In: Proceedings of the 26th Annual International Conference on Machine Learning. Association for Computing Machinery, New York, NY, USA, pp. 1073–1080. https://doi.org/10.1145/1553374.1553511.

Wilson, J., Chaudhury, S., Lall, B., 2014. Improving collaborative filtering based recommenders using topic modelling. In: Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 01, WI-IAT '14, IEEE Computer Society, USA, pp. 340–346. doi:10.1109/WI-IAT.2014.54.