New Similarity Measures for Item-based Neighborhood Collaborative Filtering

Eliuth E. López-García, Ildar Batyrshin, Grigori Sidorov

Instituto Politécnico Nacional (IPN), Centro de Investigación en Computación (CIC), Str. Juan de Dios Batiz, P.C. 07320, Mexico City, Mexico
eelopezg1700@alumno.ipn.mx, {batyr1, sidorov}@cic.ipn.mx

Abstract: Similarity measures play an important role in many areas to solve a wide variety of problems. In computer science, these measures are used in decision making, information retrieval, data mining, machine learning, and recommender systems. The recommender systems are tools that have proven their utility in filtering large amounts of information and giving recommendations useful for users. Neighborhood collaborative filtering is the most common recommender system approach implemented by cutting-edge companies. A key element of this approach is the similarity measure, which is used to find neighbors with similar tastes to provide recommendations that satisfy users' needs. A drawback of this approach is the lack of user's information to generate proper recommendations. For this reason, it is important to design new similarity measures that can find the most relevant neighbors to generate more accurate recommendations for users with little information about them. This paper designs two new similarity measures that can generate good recommendations with little information about users. These similarity measures have been tested using MovieLens datasets and different rating prediction methods, and they have shown a good performance in comparison with other similarity measures designed to address the recommendation problem.

Keywords: rating scale; recommender systems; collaborative filtering; neighborhood; item-based; similarity measure; similarity; cold-start

1 Introduction

Recommender systems (RS) are software tools and techniques providing suggestions for items (e.g., movies, songs, books, applications, websites, travel destinations, and e-learning material) to be of use to a user [1]. These systems are created to tackle the need to filter the amount of information generated on the Internet and get the one to meet users' needs. Thus, they have been useful tools for e-commerce companies (such as Amazon, e-bay, Google, Netflix, etc.) to provide automated and personalized suggestions of products to customers.

The recommender systems can be seen as information process systems due to the variated quantity of information processed to afford suggestions of products to users in an automated and personalized manner. The information processed by recommender systems may be explicit, (users' ratings), or implicitly, (users' behavior; applications downloaded; viewed or purchased items) [5]. Thus, data is primary about users and items. However, Ricci [1] refers to the data used by recommender systems as three kinds of objects:

- *Items*. The recommended objects.
- *Users*. The users of the system.
- *Transaction*. A recorded interaction between a user and the RS (ratings, reviews, etc.).

Letters U and I are frequently used to represent the set of users and items in the system, respectively. The set of possible values for a rating is represented by S and R is the set of rates recorded in the system. The interactions of users and items are commonly represented in a user-item rating matrix, see Fig. 1. Grey cells represent items rated by users and blank cells are not rated items. Typically, the number of users and items in the dataset are denoted by n and m respectively. Thus, it is an $n \times m$ matrix.

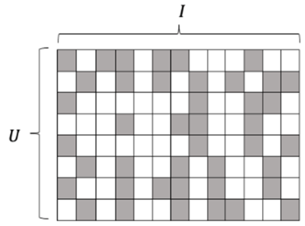


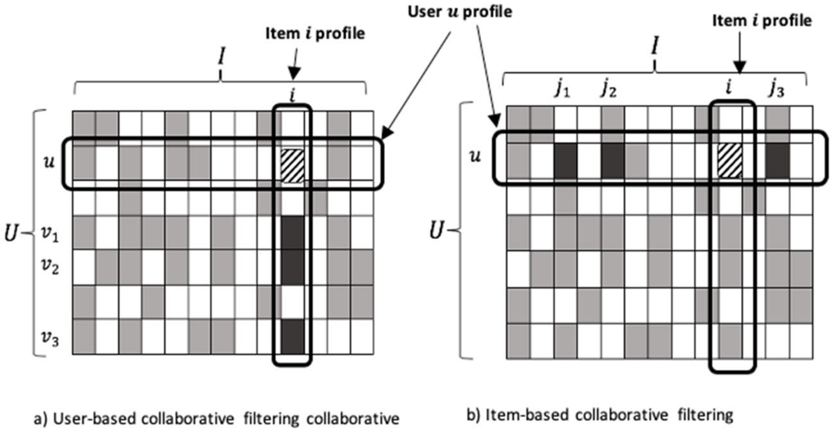
Figure 1
The user-item matrix. Grey cells represent users' ratings *R*

Even when there are many recommender systems approaches, collaborative filtering is widely used in online stores due to its proven success in this field [1, 2, 6, 12]. To recommend products to an active user, this approach considers the opinion of similar users to the active user about these products. Discovering those similar users is a challenging part of these methods. Thus, the selection of the appropriate similarity measure plays an important role in generating good recommendations, which is reflected in the active user's satisfaction.

Neighborhood-Based is a type of collaborative filtering method in which ratings gathered in the system are used right away to predict ratings for new items. There are two flavors for making the predictions: *User-Based* estimates the rating the user $u \in U$ would give to an item $i \in I$ by using the i's ratings given by other users v, best known as neighbors, which have a similar rating taste to user u. *Item-Based* estimate the rating the user u would give to an item i considering the rating user u

give to items $j \in I$ similar to i. In this case, two items are said to be similar if a considerable number of users have rated these items in a similar manner.

Fig. 2 illustrates the idea of Neighborhood-based collaborative filtering approaches; each row is a user's rating vector, and the columns are the items in the dataset. Grey cells represent items rated by users and blank cells are not rated items. Black cells represent the ratings used to predict the rating user u would give to an item i, represented by the striped cell.



a) User-based collaborative filtering collaborative

b) Item-based collaborative filtering

Figure 2
User-based and Item-based collaborative filtering to estimate the rating user u would give to an item i

2 The Cold-Start Problem

Collaborative filtering requires information about the user preferences to recommend or not an item. However, it is very common not to have enough information about users or items to create the recommendation, typically when they are new in the system. This problem is named the cold-start problem. [1, 12]

3 Similarity Measures in Recommender Systems

Similarity measures tell us how similar two objects are and quantify that similarity. Nevertheless, the similarity can also be given by their correlation [1]. The k-NN classifier is the preferred approach to collaborative filtering. This classifier is highly dependent on defining an appropriate similarity or distance measure. Hence, the choice of the appropriate similarity measure is the most critical component in these methods to make good recommendations.

Even when there are many similarity functions, the preferred ones in recommender systems are *Cosine* similarity and *Pearson's Correlation Coefficient* [1, 2]. However, they have been proven low performance in the recommendation problem domain. Therefore, new similarity measures have been proposed by many researchers to tackle the recommendation problem.

Recalling that users u and v are considered vectors of ratings. Let's now define r_{ui} and r_{vi} as the user-item rating given by user $u, v \in U$ to item $i \in I$. Then, I_u and I_v are the set of items rated by user u and v, respectively, and I_{uv} is the set of common rated items by both users. Hence, their *cosine* similarity can be expressed as the cosine angle that they form.

$$sim(u,v)^{COS} = \frac{\sum_{i\in I_{uv}} r_{ui} \cdot r_{vi}}{\sqrt{\sum_{i\in I_u} r^2{}_{ui}} \cdot \sqrt{\sum_{i\in I_v} r^2{}_{vi}}} \tag{1}$$

A drawback of cosine similarity is that it does not consider the differences in the mean and variance of the vectors u and v.

Pearson's Correlation Coefficient measures the linear relationship between the two vectors, users u and v rating vectors in recommender systems. It considers the average rating value of the two vectors u and v defined as $\bar{r}_u$ and $\bar{r}_v$ respectively.

$$sim(u,v)^{PCC} = \frac{\sum_{i\in I_{uv}} (r_{ui} - \bar{r}_u) \cdot (r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i\in I_{uv}} (r_{ui} - \bar{r}_u)^2} \cdot \sqrt{\sum_{i\in I_{uv}} (r_{vi} - \bar{r}_v)^2}} \tag{2}$$

A modified version of equation (2) is the *Constrained Pearson's Correlation Coefficient* which was created to emphasize the effect of positive and negative ratings [4] by using the median of the rating scale. For instance, $\mathrm{r}_{\mathrm{med}} = 3$ on a scale from 1 to 5.

$$sim(u,v)^{CPCC} = \frac{\sum_{i\in I_{uv}} (r_{ui} - r_{med}) \cdot (r_{vi} - r_{med})}{\sqrt{\sum_{i\in I_{uv}} (r_{ui} - r_{med})^2} \cdot \sqrt{\sum_{i\in I_{uv}} (r_{vi} - r_{med})^2}} \tag{3}$$

The *Weighted Pearson's Correlation Coefficient* is another modified version of equation (2) which considers the common items between user u and v [13].

$$sim(u,v)^{WPCC} = \begin{cases} sim(u,v)^{PCC} \cdot \frac{|I_{uv}|}{H}, |I_{uv}| \le H \\ sim(u,v)^{PCC}, otherwise \end{cases} \tag{4}$$

where H is an experimental value, it is set to 50 based on [2].

Another function that also considers the common items between user u and v is *Sigmoid Pearson's Correlation Coefficient* [16].

$$sim(u,v)^{SPCC} = sim(u,v)^{PCC} \cdot \frac{1}{1 + exp\left(-\frac{|I_{uv}|}{2}\right)} \tag{5}$$

It is very common that users tend to give low rates to items they like very much. Thus, the *Adjusted Cosine* measure was presented to consider the preference of the user's rating [14].

$$sim(u,v)^{ACOS} = \frac{\sum_{i\in I}(r_{ui}-\bar{r}_u)\cdot(r_{vi}-\bar{r}_v)}{\sqrt{\sum_{i\in I}(r_{ui}-\bar{r}_u)^2}\cdot\sqrt{\sum_{i\in I}(r_{vi}-\bar{r}_v)^2}} \tag{6}$$

Jaccard is another widely used measure. The main idea is that two users are more similar if they have more common ratings. However, it does not consider absolute ratings value.

$$sim(u,v)^{Jaccard} = \frac{|I_u \cap I_v|}{|I_u \cup I_v|} \tag{7}$$

So far, the state-of-art similarity measures were presented. However, new metrics have been proposed to address the recommender system problem. Shardanand proposed the *Mean Squared Difference* based on the mean squared difference distance [4, 8].

$$distance(u,v)^{MSD} = \frac{\sum_{i\in I_{uv}}(r_{ui}-r_{vi})^2}{|I_{uv}|} \tag{8}$$

Once the MSD distance is calculated, then all users whose $distance(u,v)^{MSD} < L$ are selected, and finally, the similarity is calculated using the following equation.

$$sim(u,v)^{MSD} = \frac{L-distance(u,v)^{MSD}}{L} \tag{9}$$

The problem with MSD is that it only considers the absolute rating, but it does not consider the percentage of common ratings. Nevertheless, Jaccard and MSD can be merged and form a metric that considers both absolute ratings and the percentage of common ratings [12], named *Jaccard Mean Squared Difference*.

$$sim(u,v)^{JMSD} = sim(u,v)^{Jaccard} \cdot sim(u,v)^{MSD} \tag{10}$$

A similarity measure proposed to alleviate the cold-statr problem in recommender systems is the one proposed by Ahn called *PIP* [6]. This measure is made-up of the following three factors of similarity:

1. *Proximity*, given two ratings, calculates the absolute difference between them and considers whether they are in agreement or not, giving a penalization to ratings in disagreement.
2. *Impact*, represents how strongly an item is accepted or refused by users.
3. *Popularity*, tells how common two users' ratings have. Two ratings can provide more information about the similarity of two users if the average rating of both users has an important difference from the average of total users' ratings.

Then, the PIP similarity between user u and v can be calculated using equation (11):

$$sim(u,v)^{PIP} = \sum_{i\in I_{uv}} PIP(r_{ui}, r_{vi}) \tag{11}$$

where $PIP(r_{ui}, r_{vi})$ is the PIP value for the two ratings r_{ui} and r_{vi} on item i by user u and v respectively. PIP can be defined by equation (12):

$$PIP(r_{ui}, r_{vi}) = Proximity(r_{ui}, r_{vi}) \cdot Impact(r_{ui}, r_{vi}) \cdot Popularty(r_{ui}, r_{vi}) \quad (12)$$

Liu proposed the *New Heuristic Similarity Model* (NHSM) which considers the common ratings, context information and it is normalized [2]. Liu improved PIP by taking advantage of the sigmoid function, which is a non-linear function, and it can penalize bad similarity or reward good similarity. The resulting function is named *Proximity Significance Singularity* (PSS):

- *Proximity*, considers the remoteness between two ratings.
- *Significance*, ratings are more significant when two ratings are further away from the median rating.
- *Singularity*, represents how two ratings are different with regard to other ratings.

The similarity measure of PSS is given by equations (13) and (14).

$$sim(u,v)^{PSS} = \sum_{i \in I_{uv}} PSS(r_{ui}, r_{vi}) \quad (13)$$

$$PSS(r_{ui}, r_{vi}) = Proximity(r_{ui}, r_{vi}) \cdot Significance(r_{ui}, r_{vi}) \cdot Singularity(r_{ui}, r_{vi}) \quad (14)$$

In addition, Liu also made use of a modified version Jaccard formula to penalize the small proportion of common ratings. The resulting modified Jaccard is given by equation (15):

$$sim(u,v)^{Jaccard\prime} = \frac{|I_u \cap I_v|}{|I_u| \times |I_v|} \quad (15)$$

Then equations (14) and (15) are combined as follows:

$$sim(u,v)^{JPSS} = sim(u,v)^{Jaccard\prime} \cdot sim(u,v)^{PSS} \quad (16)$$

Liu also considers each user's preference using equation (17). The idea behind is that some users might unwillingly give high scores to items they like, or vice versa.

$$sim(u,v)^{URP} = 1 - \frac{1}{1 + exp(-|\mu_u - \mu_v| \cdot |\sigma_u - \sigma_v|)} \quad (17)$$

where μ_u and μ_v represent the mean rating of user u and v respectively. The σ_u and σ_v are the standard variance of user u and v.

Finally, NHSM is the combination of equations (16) and (17):

$$sim(u,v)^{NHSM} = sim(u,v)^{JPSS} \cdot sim(u,v)^{URP} \quad (18)$$

4 Proposed Similarity Measures

The proposed similarity is inspired by the *Constrained Pearson's Correlation Coefficient* (CPCC), which emphasizes the positive and negative rates of the two users to calculate their correlation. As observed in equation (3), CPCC uses the center value of the scale, r_{med}, to distinguish when a rating is positive or negative.

Nevertheless, it has two disadvantages: it does not consider the proportion of common rated items of two users, and it may return low similarity even when there are many common rate values. For example, consider the data in Fig. 3 a). When calculating the users' CPCC similarity, presented in Fig. 3 b), the similarities seem not to be correct. As an example, $u1$ should have a high similarity with $u3$, instead, it has a high similarity with $u2$.

User	$i1$	$i2$	$i3$	$i4$
$u1$	4	3	5	4
$u2$	5	3		
$u3$	4	3	3	4
$u4$	2	1		
$u5$	4	2		

a) Users-Items-Ratings matrix taken from[2]

User	$u2$	$u3$	$u4$	$u5$
$u1$	1	0.577	-0.447	0.707
$u2$		1	-0.447	0.707
$u3$			-0.447	0.707
$u4$				0.316

b) CPCC

User	$u2$	$u3$	$u4$	$u5$
$u1$	0.5	0.577	-0.223	0.353
$u2$		0.5	-0.447	0.707
$u3$			-0.223	0.353
$u4$				0.316

c) JCPCC

User	$u2$	$u3$	$u4$	$u5$
$u1$	0.333	0.384	0	0.235
$u2$		0.5	0	0.707
$u3$			0	0.353
$u4$				0

d) PJCPCC

Figure 3
Users' similarities matrix for CPCC, JCPCC and PJCPCC

4.1 Jaccard Constrained Pearson's Correlation Coefficient

CPCC can be multiplied by *Jaccard* similarity measure, equation (7), to give it support. The resulting similarity measure is named *Jaccard Constrained Pearson's Correlation Coefficient*, JCPCC.

$$sim(u,v)^{JCPCC} = sim(u,v)^{Jaccard} \cdot sim(u,v)^{CPCC} \quad (19)$$

4.2 Positive Constrained Pearson's Correlation Coefficient

Continuing with the example, Fig. 3 c) displays the similarities using JCPCC. Now, $u1$ has the highest similarity with $u3$.

However, suggesting items to the active user u for which it has a positive response is the aim of recommender systems. Therefore, the Jaccard similarity can only consider the common rates whose value is positive. A rating r_{ui} is positive if $r_{ui} \geq \theta$, for instance, $\theta = 4$ on a scale from 1 to 5 [17, 18, 19]. This similarity is named *Positive Jaccard Constrained Pearson's Correlation Coefficient*, PJCPCC. In equation (20), the parameter θ is used to filter the positive rates in u and v.

$$sim(u,v)^{PJCPCC} = sim(u,v,\theta = 4)^{Jaccard} \cdot sim(u,v)^{CPCC} \quad (20)$$

The resulting similarities using PJCPCC are then displayed in Fig. 3 d). The similarities now are adjusted based on the common items rated positively and the correlation given by their rates.

5 Datasets

For the experiments, two of the most used datasets from Movie Lens were selected (https://grouplens.org/): ML-100K and ML-Latest-Small. Table 1 illustrates the datasets information and Fig. 4 illustrates the rating distribution of the datasets.

Table 1

Datasets information

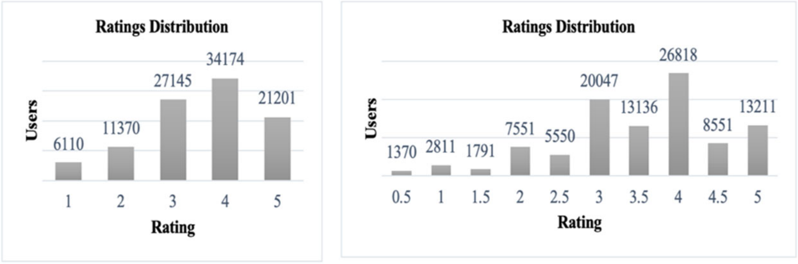
Dataset	Ratings	Users	Items	Density $\frac{\#\text{ Rates} * 100}{\#\text{Users} * \#\text{Items}}$	Rate Scale
ML-100K	100,000	944	1,682	6.3%	[1, 2, 3, 4, 5]
ML-Latest-Small	100,836	610	9724	1.7%	[0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5]

The test dataset is created by following the next steps:

1. The users' set U is randomly split: 80% for training (U_{train}), and 20% for testing (U_{test}), the users to which the system creates recommendations.
2. To simulate a cold-start behavior, only ten ratings are randomly selected as training ratings (R_{train}) for each user in testing (U_{test}). The remaining rates are the testing ratings (R_{test}), rates to be predicted and evaluated.

The whole process is done using k-folds cross-validation with $k = 5$. Thus, each fold contains a disjointed user test set, U_{test}, with the training rates R_{train} of each user $u \in U_{test}$. The set of items rated in R_{train} by user u is denoted as I_{train}.

Similarly, the set of items rated in R_{test} by user u is denoted as I_{test}. Hence, the set of all items rated by user u is denoted as $I_u = I_{train} + I_{test}$.



a) b)

Figure 4

a) ML-100K and b) ML-Latest-Small datasets ratings distribution

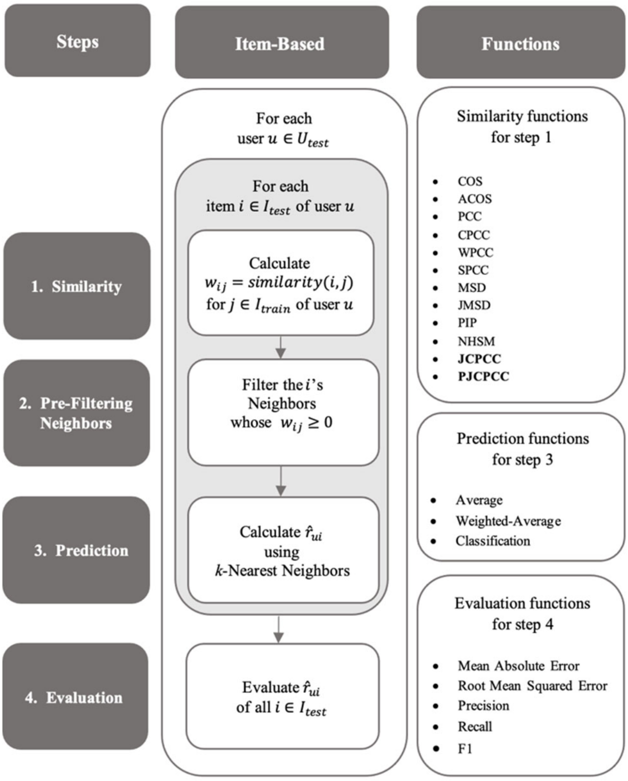

Steps
Item-Based
Functions
For each user $u \in U_{test}$
For each item $i \in I_{test}$ of user u
1. Similarity
Calculate $w_{ij} = similarity(i,j)$ for $j \in I_{train}$ of user u
2. Pre-Filtering Neighbors
Filter the i's Neighbors whose $w_{ij} \geq 0$
3. Prediction
Calculate $\hat{r}_{ui}$ using k-Nearest Neighbors
4. Evaluation
Evaluate $\hat{r}_{ui}$ of all $i \in I_{test}$
Similarity functions for step 1
COS
ACOS
PCC
CPCC
WPCC
SPCC
MSD
JMSD
PIP
NHSM
JCPCC
PJCPCC
Prediction functions for step 3
Average
Weighted-Average
Classification
Evaluation functions for step 4
Mean Absolute Error
Root Mean Squared Error
Precision
Recall
F1

Figure 5

Collaborative filtering Prediction-Evaluation framework

6 Collaborative Filtering Prediction-Evaluation Framework

The collaborative filtering prediction-evaluation framework has four steps in general: similarity, pre-filter neighbors, prediction, and evaluation. The collaborative filtering prediction-evaluation framework used in this research is illustrated in Fig. 5. The Functions column displays the evaluated similarity functions for step 1, the prediction for step 3, and the evaluation functions for step 4. The neighbors are pre-filtered by their similarity weight $w_{ij} \geq 0$ for step 2.

7 Results

The results are dived into two sections for ML-100K and ML-Latest-Small datasets. Each section contains the graphs to illustrate the similarity measures performance using *MAE*, *RMSE*, *Precision*, *Recall*, and *F1* evaluation metrics for each rating prediction function: *Average*, *Weighted-Average*, and *Classification*. The rating predictions were calculated using the k-Nearest-Neighbors' ratings, for k = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] because users in testing have up to ten ratings for training.

7.1 ML-100K Results

7.1.1 Average Rating Prediction

Fig. 6 illustrates the MAE, RMSE, precision, and recall results for *Average* rating prediction. The similarity measures with the lowest MAE are the proposed JCPCC and PJCPCC. Their MAE value is close to 0.94 when $k = 1$ and it decreases as k increases until they reach the lowest error, close to 0.84 when $k = 6$ and $k = 7$. For RMSE, JCPCC and PJCPCC are in the group of similarity measures with the lowest RMSE value, around 1.26 when $k = 1$. This value also decreases as k increases, which is about 1.05 when $k = 6$. Regarding precision, JCPCC has the highest precision value, which is about 0.64 when $k = 1$, and it is followed by PJCPCC whose value is close to 0.63. In general, the precision decreases as long as k increases, and these two similarities are affected. Regarding recall, JCPCC and PJCPCC have the highest values, which is about 0.58 when $k = 1$.

In general, recall decreases as long as k increases, but JCPCC and PJCPCC keep the highest value. Finally, Fig. 7 illustrates the F1 metric. It can be observed that the proposed JCPCC and PJCPCC have the highest value, about 0.6 when $k = 1$. This value also decreases as long as k increases. However, JCPCC and PJCPCC are in the group of similarity measures with the highest F1 value.

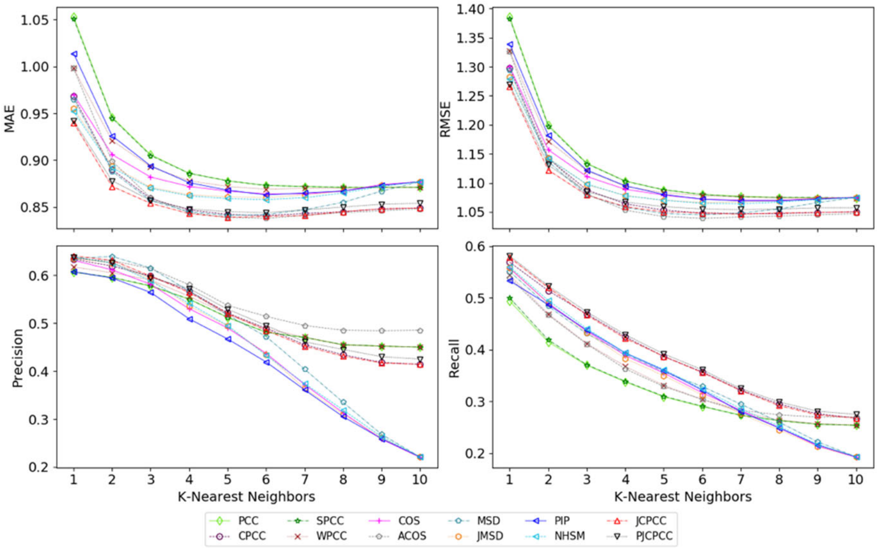


Figure 6

MAE, RMSE, Precision, and Recall vs K-Nearest Neighbors using Average rating prediction on ML-100K dataset

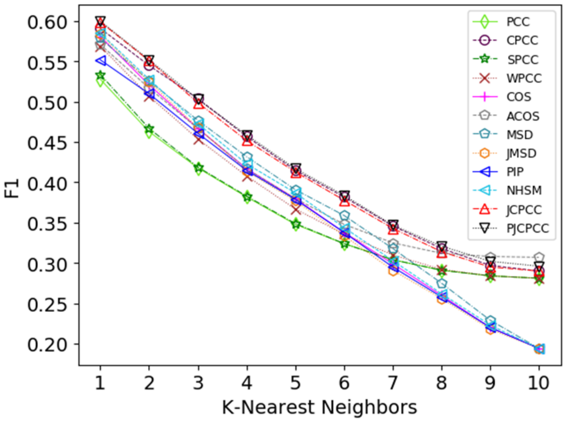


Figure 7

F1 vs K-Nearest Neighbors using Average rating prediction on ML-100K dataset

7.1.2 Weighted Average Rating Prediction

Fig. 8 illustrates the MAE, RMSE, precision, and recall results for *Weighted-Average* rating prediction. The similarity measures with the lowest MAE is the proposed JCPCC and PJCPCC similarities. Their MAE value is close to 0.94 when $k = 1$ and it decreases until they reach the lowest error, close to 0.82 when $k \geq 7$.

With respect to RMSE, JCPCC has the lowest error, around 1.26 when $k = 1$. This value decreases below 1.05 when $k \geq 5$. Regarding precision, the two similarities are in the group of the highest values, about 0.64 when $k = 1$. As observed, the precision decreases as long as k increases. However, PJCPCC keeps the highest precision value. With regard to recall, JCPCC and PJCPCC have also the highest values, which is about 0.58 when $k = 1$. Even when recall decreases as long as k increases, PJCPCC keeps the highest value.

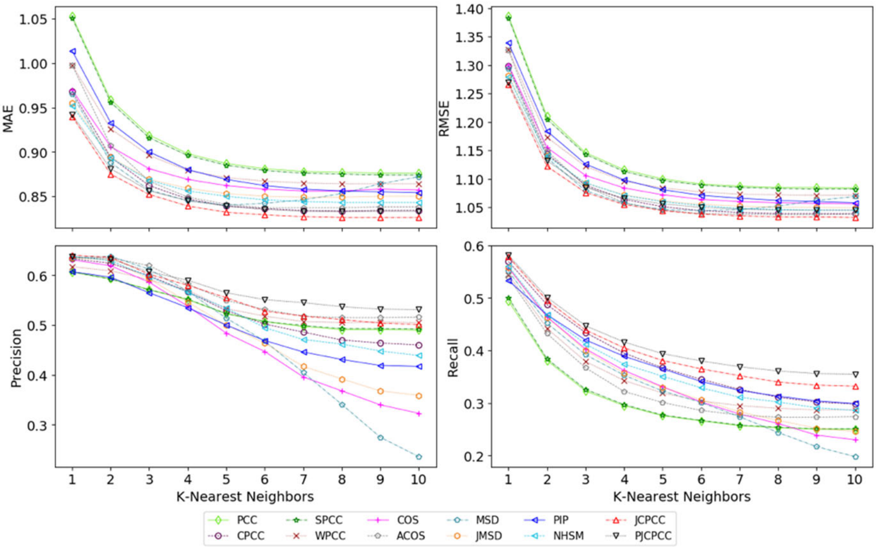


Figure 8

MAE, RMSE, Precision, and Recall vs K-Nearest Neighbors using Weighted Average rating prediction on ML-100K dataset

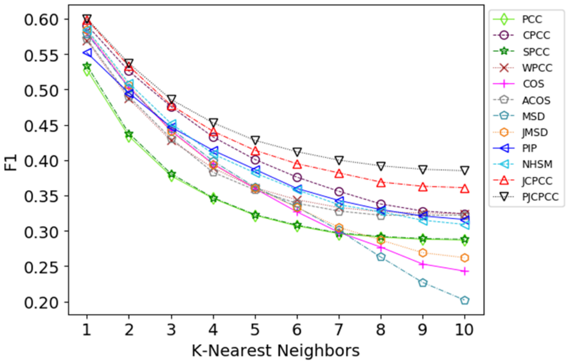


Figure 9

F1 vs K-Nearest Neighbors using Weighted Average rating prediction on ML-100K dataset

Finally, Fig. 9 illustrates the F1 metric. It can be observed that the proposed JCPCC and PJCPCC have the best performance for all k values. F1 value is about 0.6 when $k = 1$ and it decreases as long as k increases. However, JCPCC always has the highest value.

7.1.3 Classification Rating Prediction

Fig. 10 illustrates the MAE, RMSE, precision, and recall results for *Classification* rating prediction. As observed, both similarity measures, PJCPCC and JCPCC, have the best performance. The lowest MAE and RMSE errors, and the highest precision and recall values.

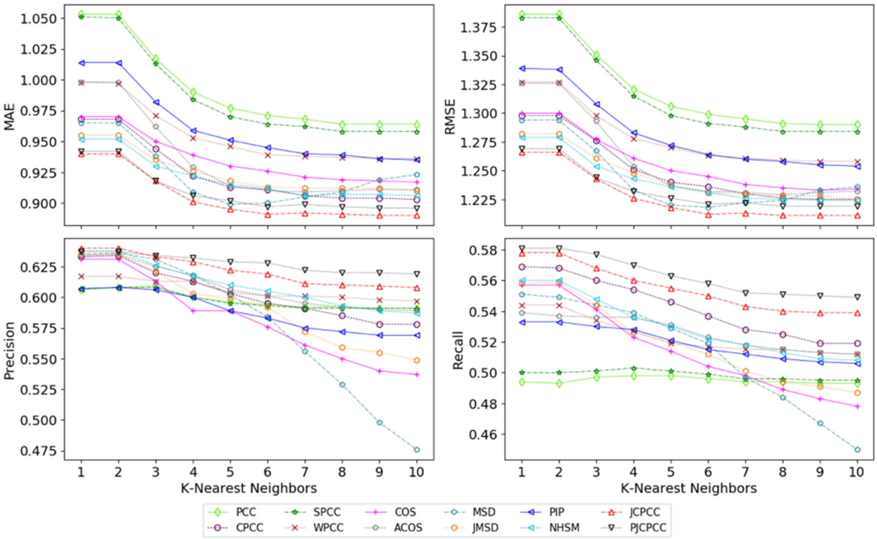


Figure 10
MAE, RMSE, Precision, and Recall vs K-Nearest Neighbors using Classification rating prediction on ML-100K dataset

MAE value is close to 0.94 when $k = 1$, and $k = 2$, then it decreases until it reaches the lowest error close to 0.89 when $k \geq 6$. Similar behavior is observed for RMSE, whose value is about 1.26 when $k = 1$ and $k = 2$. Then, it decreases until it reaches the lowest error close to 1.211 when $k \geq 8$. With regard to precision, PJCPCC and JCPCC have the highest value, about 0.625, which reminds stable regardless of the k value. Regarding recall, PJCPCC and JCPCC also have the best performance. However, PJCPCC has the highest value, around 0.58 when $k \leq 2$ and it decreases to about 0.55 when $k \geq 7$.

Finally, Fig. 11 illustrates the F1 metric. In this case, JCPCC and PJCPCC have the best performance whose highest value is about 0.6 when $k \leq 2$. Even when F1 decreases while k increases, PJCPCC holds the highest F1 values and it is followed by JCPCC with the second-highest F1 value.

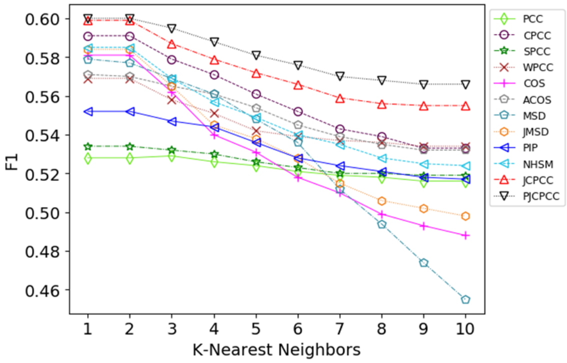


Figure 11
F1 vs K-Nearest Neighbors using Classification rating prediction on ML-100K dataset

7.2 ML-Latest-Small Results

7.2.1 Average Rating Prediction

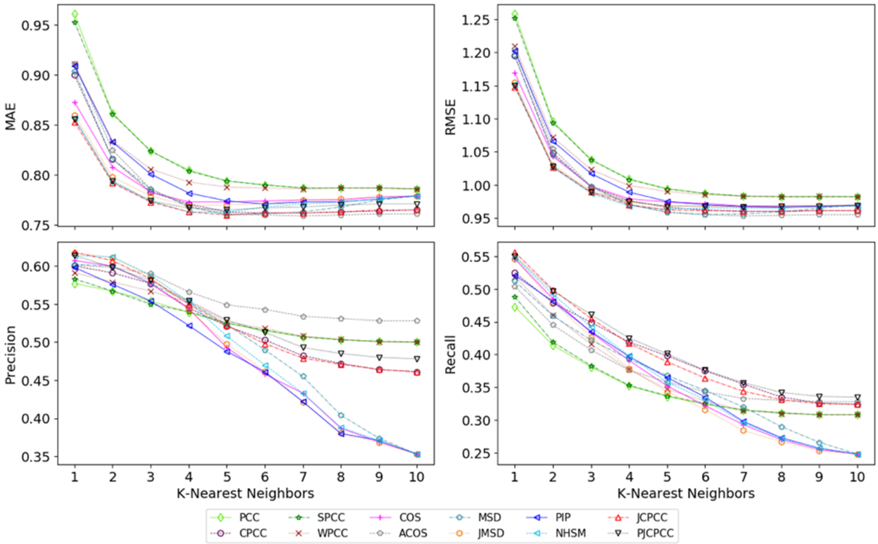


Figure 12
MAE, RMSE, Precision, and Recall vs K-Nearest Neighbors using Average rating prediction on ML-Latest-Small dataset

Fig. 12 illustrates the MAE, RMSE, precision, and recall results for *Average* rating prediction. In general, JCPCC and PJCPCC are in the group of similarities with the lowest MAE. Their MAE value is close to 0.85 when $k = 1$ and it decreases until they reach the lowest error, which is close to 0.76 when $k \geq 4$. For RMSE, JCPCC and PJCPCC also have the lowest error, around 1.15 when $k = 1$. They are also in the group of similarities with the lowest RMSE as k increases. Regarding precision, JCPCC has the highest precision value, about 0.61 when $k \leq 2$. In general, the precision decreases as long as k increases. With regard to recall, JCPCC and PJCPCC have the highest values, which is about 0.55 when $k \leq 2$, even when recall decreases while k increases, both similarity measures hold the highest recall values. Finally, Fig. 13 illustrates the F1 metric. The similarities JCPCC and PJCPCC have an F1 close to 0.57 when $k = 1$. When k increases F1 decreases, but the two similarities are in the group of similarity measures with the highest F1 values.

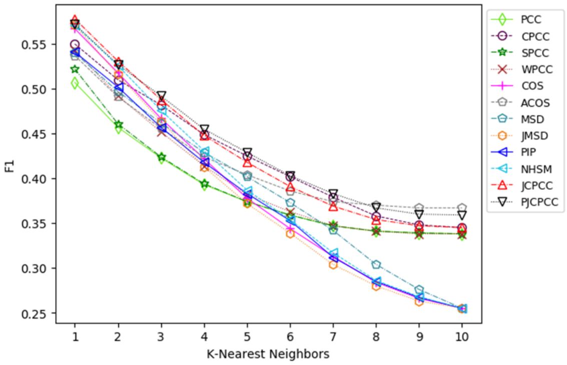


Figure 13
F1 vs K-Nearest Neighbors using Average rating prediction on ML-Latest-Small dataset

7.2.2 Weighted Average Rating Prediction

Fig. 14 illustrates the MAE, RMSE, precision, and recall results for *Weighted-Average* rating prediction. JCPCC, PJCPCC are in the group of similarity measures with the lowest MAE. Their MAE value is close to 0.85 when $k = 1$ and it decreases until they reach the lowest error, close to 0.75 when $k \geq 6$. Similarly, JCPCC and PJCPCC are also in the group of similarity measures with the lowest RMSE error, around 1.15 when $k = 1$. This value also decreases as k increases, but JCPCC and PJCPCC hold a good performance, about 0.95 when $k \geq 6$. Regarding precision, JCPCC and PJCPCC are in the group of similarity measures with the highest precision value, about 0.61 when $k \leq 3$. In general, the precision decreases as long as k increases. Similarly, JCPCC and PJCPCC have also in the group of similarity measures with the highest recall, which is about 0.55 when $k = 1$. Even when recall decreases as long as k increases PJCPCC reminds as the similarity with the highest value.

Finally, Fig. 15 illustrates the F1 metric. JCPCC has the highest value, about 0.58 when $k = 1$. It is followed by PJCPCC with a value close to 0.57. As long as k increases, the F1 value decreases for all similarity measures, but PJCPCC keeps the highest value.

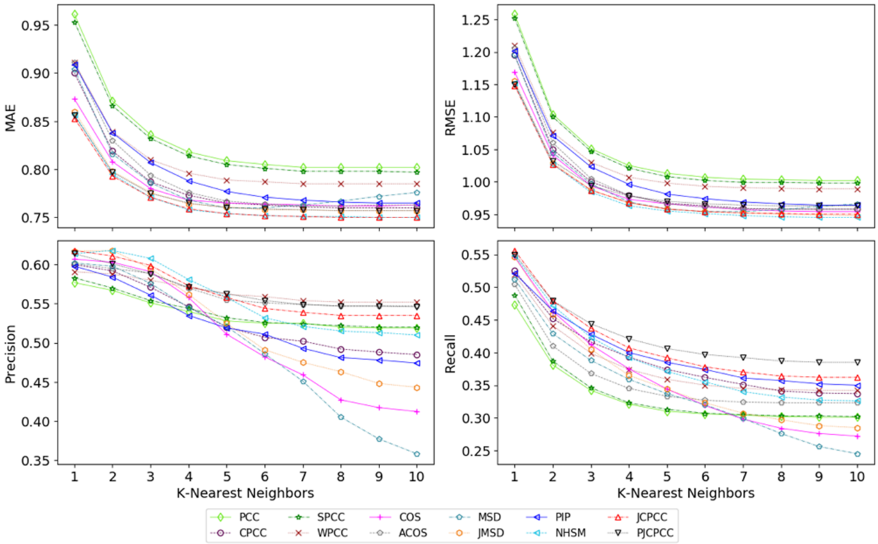


Figure 14

MAE, RMSE, Precision, and Recall vs K-Nearest Neighbors using Weighted Average rating prediction on ML-Latest-Small dataset

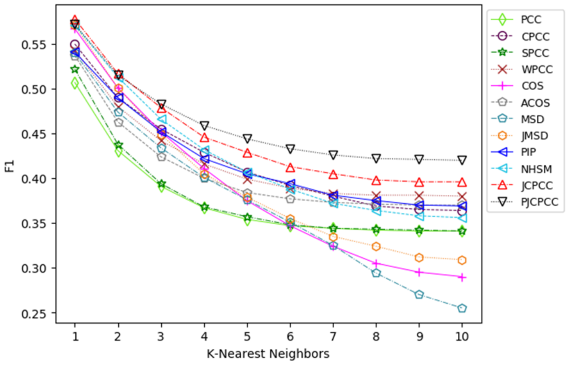


Figure 15

F1 vs K-Nearest Neighbors using Weighted Average rating prediction on ML-Latest-Small dataset

7.2.3 Classification Rating Prediction

Fig. 16 illustrates the MAE, RMSE, precision, and recall results for *Classification* rating prediction. The results are similar to those for this scenario when using the ML-100K dataset. In general, both similarity measures, PJCPCC and JCPCC, have the best performance. The lowest MAE and RMSE errors, and the highest precision and recall values. MAE value is close to 0.86 when $k = 1$ and $k = 2$, then it decreases until it reaches the lowest error close to 0.81 when $k \geq 7$.

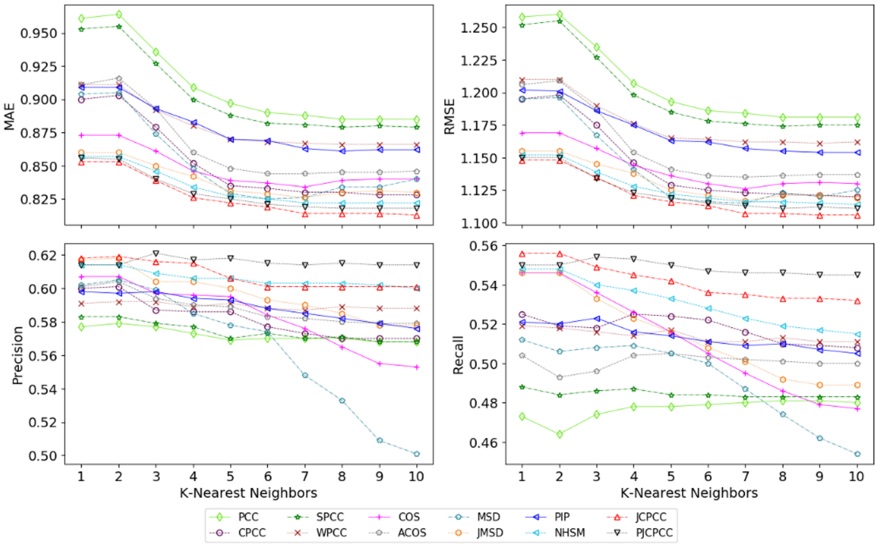


Figure 16

MAE, RMSE, Precision, and Recall vs K-Nearest Neighbors using Classification rating prediction on ML-Latest-Small dataset

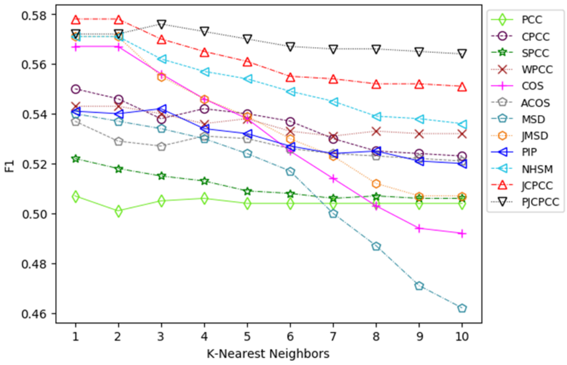


Figure 17

F1 vs K-Nearest Neighbors using Classification rating prediction on ML-Latest-Small dataset

Similar behavior is observed for RMSE, whose value is about 1.15 when $k = 1$ and $k = 2$, and then it decreases until it reaches the lowest error close to 1.1 when $k \geq 7$. It is also observable that PJCPCC and JCPCC precision value reminds stable regardless of the k value, which is about 0.61. Regarding recall, PJCPCC and JCPCC have the best performance. However, JCPCC has the highest value, around 0.55 when $k \leq 2$, and then PJCPCC is the one with the highest value about 0.54 $k \geq 3$ and this value reminds stable regardless of the k value.

Finally, Fig. 17 illustrates the F1 metric. In this case, JCPCC and PJCPCC have again the best performance. When $k \leq 2$ JCPCC has the highest F1 value, about 0.58. In this k range, PJCPCC's F1 value is about 0.57. It can be observed that when $k \geq 3$, PJCPCC keeps the highest value, which is about 0.57. In this k range, JCPCC now has the second-highest F1 value, which is between 0.55 and 0.54.

Conclusions

This paper analyses the similarity measures used to address the cold-start problem in Recommender Systems; when there is a small amount of information about users' preferences. In addition, it proposes two similarity measures to address this problem, JCPCC, and PJCPCC. The new similarity measures were analyzed and compared with state-of-art and other similarity measures using the *Item-Based* collaborative filtering approach, and different rating prediction functions (*Average*, *Weighted-Average,* and *Classification* rating predictions). The experiments were performed using two different MovieLens datasets, and the evaluation metrics used were MAE, RMSE, precision, recall, and F1. The results produced by the experiments proved that these new similarity measures could yield better performance than other measures used in cold-start scenarios in *Item-Based* collaborative filtering approaches.

Acknowledgment

The work is partially supported by projects 20220857 and 20211874 of SIP IPN, Mexico. The authors thank the CONACYT for the computing resources brought to them through the Deep Learning Platform for Language Technologies of the Supercomputing Laboratory of the INAOE, Mexico, and acknowledge the support of Microsoft through the Microsoft Latin America Ph.D. Award.

References

[1] F. Ricci, L. Rokach, B. Shapira, and K. P. B., Recommender Systems Handbook, First. Boston, MA: Springer US, 2011

[2] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering," Knowledge-Based Syst., Vol. 56, pp. 156-166, 2014

[3] Y. Wang, J. Deng, J. Gao, and P. Zhang, "A hybrid user similarity model for collaborative filtering," Inf. Sci. (Ny), Vol. 418-419, pp. 102-118, 2017

[4] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating 'word of mouth,'" Conf. Hum. Factors Comput. Syst. - Proc., Vol. 1, pp. 210-217, 1995

[5] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," Knowledge-Based Syst., Vol. 46, pp. 109-132, 2013

[6] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," Inf. Sci. (Ny), Vol. 178, No. 1, pp. 37-51, 2008

[7] J. Bobadilla, A. Hernando, F. Ortega, and J. Bernal, "A framework for collaborative filtering recommender systems," Expert Syst. Appl., Vol. 38, No. 12, pp. 14609-14623, 2011

[8] P. Maes and F. R. Morgenthaler, "Social Information Filtering for Music Recommendation," Master Thesis, 1994

[9] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-N recommendation tasks," RecSys'10 - Proc. 4th ACM Conf. Recomm. Syst., No. January 2010, pp. 39-46, 2010

[10] B. K. Patra, R. Launonen, V. Ollikainen, and S. Nandi, "A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data," Knowledge-Based Syst., Vol. 82, pp. 163-177, 2015

[11] I. Batyrshin, "Towards a general theory of similarity and association measures: Similarity, dissimilarity and correlation functions," J. Intell. Fuzzy Syst., Vol. 36, No. 4, pp. 2977-3004, 2019

[12] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, "A collaborative filtering approach to mitigate the new user cold start problem," Knowledge-Based Syst., Vol. 26, pp. 225-238, 2012

[13] J. L. Herlocker, Joseph A. Konstan, A. Borchers, and J. Riedl, "An Algorithmic Framework for Performing Collaborative Filtering," ABA J., Vol. 102, No. 4, pp. 24-25, 2017

[14] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," Proc. 10th Int. Conf. World Wide Web, WWW 2001, pp. 285-295, 2001

[15] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative filtering recommender systems," Found. Trends Human-Computer Interact., Vol. 4, No. 2, pp. 81-173, 2010

[16] M. Jamali and M. Ester, "TrustWalker: A random walk model for combining trust-based and item-based recommendation," in Proceeding of the ACM SIGKDD Int. Conference on Knowledge Discovery and Data Mining, 2009, pp. 397-405

[17] I. Z. Batyrshin, "Constructing correlation coefficients from similarity and dissimilarity functions," Acta Polytechnica Hungarica, Vol. 16, No. 10, pp. 191-204, 2019

[18] I. Batyrshin, F. Monroy-Tenorio, A. Gelbukh, L. A. Villa-Vargas, V. Solovyev, and N. Kubysheva, "Bipolar rating scales: A survey and novel correlation measures based on nonlinear bipolar scoring functions," Acta Polytechnica Hungarica, Vol. 14, No. 3, pp. 33-57, 2017

[19] F. Monroy-Tenorio, I. Batyrshin, A. Gelbukh, V. Solovyev, N. Kubysheva, and I. Rudas, "Correlation measures for bipolar rating profiles," in Fuzzy Logic in Intelligent System Design, Springer, pp. 22-32, 2018