



A Novel Algorithmic Similarity Measure for Collaborative Filtering: A Recommendation System Based on Rating Distances

*¹Şule Öztürk Birim, ²Ayça Tümtürk

¹Manisa Celal Bayar Üniversitesi, Salihli İktisadi ve İdari Bilimler Fakültesi, İşletme, Sayısal Yöntemler ABD, Manisa,

sule.ozturk@cbu.edu.tr 

²Manisa Celal Bayar Üniversitesi, İktisadi ve İdari Bilimler Fakültesi, İşletme, Üretim Yönetimi ve Pazarlama ABD, Manisa

ayca.tumturk@cbu.edu.tr 

Abstract

Internet sources contain a vast amount of information about items that people desire to purchase. It is impossible to evaluate these resources and come to an informed decision. People need automated systems that evaluate previous information and propose item alternatives. Recommending items using a smart system, which is based on the previous user preferences, has growing importance since the available product data is exponentially growing. Additionally, it is difficult to find new and correct things that a user would like among this massive amount of data. To make accurate recommendations with a smart system, researchers and practitioners use collaborative filtering methods with similarity calculation based on user preferences. The crucial point in collaborative filtering is to find a valuable measure that resembles correct similarity between users. The current similarity metrics in the literature have some disadvantages in conducting accurate recommendations. To improve the recommendation performance, this study proposes a novel similarity measure that assesses the distance between the user's ratings and the median score. Considering distance from the median score is essential since some users may prefer to rate close to the median rather than the extremes. Experiments were conducted with a famous collaborative filtering dataset. Results showed that proposed similarity measure demonstrated superior performance regarding the recommendation accuracy. Implications of our results for XYZ are discussed.

Keywords: Collaborative filtering, recommender systems, similarity measure, prediction accuracy, classification accuracy

1. INTRODUCTION

In recent times, people have been exposed to a growing amount of data one-commerce as well as social media websites on several products. When dealing with such data, it is often difficult for individuals to evaluate and compare all of their product options to make an informed purchase decision. During the pandemic period, online shopping remarkably increased. Results of a recent study showed that online shopping has enlarged by almost 20% in 2020 and near 9% in 2021 [1]. Around 70% of customers who shop online do not come to a final decision before looking at the product ratings or reading the reviews [2]. These numbers indicate individuals consider others' opinions when deciding to purchase a product online. However, it is hard to examine all of the views about a product, since there may be too many. To alleviate this difficulty, automated smart systems, which make recommendations based on previous preferences, have been developed. These smart systems are called recommender systems and are widely used by commercial sites like Amazon [3] as well as non-commercial websites that possess the purpose of research such as GroupLens [4]. Recommender systems aim to make the lives of the people more comfortable by recommending the products or services that are similar to their previous preferences.

Rich (1979) provided one of the first references of recommender system modeling. In this research, a librarian known as Grundy grouped users into clusters based on their book type preference, including education, sports and romance. Then using these clusters, Grundy would recommend novels to people [5].

The methods that the recommender systems use to make suggestions can be classified as content-based methods, collaborative filtering and hybrid methods. The content-based recommender systems use content information related to the item and makes recommendations based on the user's previous item preferences [6]. The commonly used collaborative filtering methods suggest items based on the similarities between the items or the different users [7]. The collaborative filtering methods are widely used in practice as well as in academia. Lastly, the hybrid systems combine the content based and collaborative filtering methods of recommender systems [8].

Collaborative filtering systems can be classified as model-based and memory-based methods [9]. A model-based system uses an offline user database to predict a model, which in turn is utilized to make predictions. Unlike the model-based systems, a memory-based system requires the entire user database to calculate the similarities between the users. In memory-based approaches, different metrics may

be used to estimate user or item similarity. Widely used similarity measures in memory-based collaborative filtering are Pearson correlation coefficient and Vector cosine similarity metrics.

This study aims to increase the prediction performance of user-based collaborative filtering in a recommendation system. To reach this goal, we offer a novel algorithmic similarity measure that is to be used in a collaborative filtering system. Traditional similarity methods (i.e., Pearson correlation, Cosine) as well as some recently proposed similarity measures (i.e., in the literature can lead to misleading similarities, such that while ratings show that two users are similar, similarity measure demonstrate the opposite. The traditional methods also have some drawbacks in providing accurate recommendations when the data set is too sparse [10]. Additionally, measures in the literature do not consider the degree of the user tendency of rating close to the median. Some users may avoid giving ratings which are on the extremes. For example, they may prefer to rate the second highest score to an item which they liked a lot. The similarity measure proposed in this study addresses this drawback in the literature by avoiding extreme ratings. Our similarity measure was experimented on one of the most popular datasets used in the recommender systems. We compare our measure with the traditional similarity measures as well as two other similarity measures, which were recently introduced in the literature.

In the next part of this paper, we give a brief review of the literature concerning prominent collaborative filtering studies. Then, the proposed similarity method is explained by comparing it to the existing similarity methods. We then discuss the experiments that were conducted to test the proposed similarity measure. Then the results of the experiments are then discussed to highlight implications for the literatures on recommendation systems.

2. LITERATURE REVIEW

Several studies have used collaborative filtering methods to recommend items to their users. For example, the collaborative filtering term was first introduced in the literature by ... [11]. In their study, an experimental mail system known as Tapestry was introduced as a hybrid approach, such that it supported both the content-based and collaborative filtering methods. In similar vein, Goldberg and Roeder's [12] proposed a collaborative filtering algorithm that was named as Eigentaste. The Eigentaste algorithm employed principal component analysis to solve eigenvalues and eigenvectors matrices with 2500000 ratings, which belong to 57000 users on Jester website, an online joke recommender system. Researchers used normalized mean absolute error (NMAE) [13], [14] to compare the Eigentaste with other selected algorithms, including the algorithm proposed in [9]. Researchers concluded that the algorithm proposed in [9] provides good results by NMAE, but it ignored user differences entirely [12].

In the collaborative filtering methods, numerous similarity metrics were proposed to measure the similarity between the users or the items. When computing the similarity, the most popular traditional approach is the Pearson correlation

coefficient (PCC) [15]. PCC considers all the items or the users in the user-item matrix as equal subjects. It does not consider the commonly rated items or the common users of the identified items. In [16], the authors suggested an algorithm to overcome the disadvantages of PCC and computed the weights for distinct items depending on the scores obtained from training users. They showed that the weighted PCC system gives better performance than the traditional PCC method on two different datasets.

An essential issue in recommender systems is to calculate similarities and perform predictions when the data in the user-item matrix is very sparse. Data sparsity means users rate only a few items at a time?. This problem is known as a *cold start problem*. A cold start problem is a severe issue in collaborative filtering, and many researchers are trying to find solutions to it by offering new approaches. For example, Ahn [17] argued that PCC and Cosine (COS) [18], which are the commonly used similarity measures in collaborative filtering, are not enough when a cold-start problem occurs. To solve this problem, Ahn [17] introduced a similarity measure named PIP (proximity-impact-popularity), which considers the facets of proximity, impact, and popularity of the user ratings. The author demonstrated that in an artificial cold start problem, PIP gives better results than other measures, whereas PCC produced the best results when considering the real data set.

The study of Luo and colleagues [19] proposed a global similarity concept to calculate the similarities between users who have no commonly rated items. The authors used both local and global user similarities when calculating the overall similarity between users. Furthermore, they used the surprisal vector instead of the users' rating vector when computing local similarity. Their measure outperformed PCC based on MAE [20]. Al-Shamri and Bharadwaj [21] integrated demographic or genre information of users into the collaborative filtering. They suggested a hybrid fuzzy-genetic recommender system and their model derived better MAE values when compared to PCC. Jamali and Ester [22] proposed a model that was a mix of item-based and trust-based collaborative filtering methods. The trust-based approach uses Epinions website, which includes trust network and user ratings together. For both the users as well as the cold start users, the suggested model outperformed the traditional similarity methods.

Bobadilla et al. [23]–[28] conducted several studies about collaborative filtering between 2010–2012. In one study [24] the authors used both the MSD (mean squared difference) [29] as well as the Jaccard [30] metric to overcome the weaknesses of PCC. In Netflix and Movielens datasets, new metric provided acceptably good results, while the FilmAffinity data set did not demonstrate appropriate effects [24]. In another study, [27] the authors proposed a similarity metric considering the significance of items by weighting them while calculating PCC and COS, instead of traditional PCC and COS. They measured prediction performance with MAE, precision [31], and recall [32]. They obtained acceptable results [27]. Bobadilla et al [23] proposed a neural network learning based on the similarity metric, which was found to be faster than other measures when it came to MAE, recall, and precision. In yet another study, a framework for collaborative filtering was proposed

to assess the performance of the recommendations based on the trust on users' neighbors and the novelty of the conducted recommendations [26].

Anand and Bharadwaj [33] focused on the data sparsity problem in collaborative filtering with computing local and global similarities like [19] and used the prediction formula in [34] to conduct recommendations. Cai et al [18] derived the typicality from cognitive psychology to be used in collaborative filtering and observed improvements on MAE. Baltrunas and Ricci [31] used item splitting with matrix factorization and nearest neighbor collaborative filtering algorithms. In another study, Luo et al. [13] introduced an improving non-negative matrix factorization-based collaborative filtering method to recommend items.

In a recent paper, Chen et al. [35] used a special type of k-means clustering to focus on preventing user privacy. Recommendations were formed from the set of neighbors, which belonged to the selected cluster aiming to achieve privacy. The results indicated that the proposed system improved performance compared to the previous ones. Another study [36], aimed to address the sparsity problem in movie recommendation by forming a collaborative system using a singular value decomposition. In this paper, similarities were calculated by using the movie's content information in cosine similarity calculations.

A recent study of Afoudi et al [37] created a hybrid recommender system that combines collaborative filtering, content based similarity and a special type of neural network. The authors used the hybrid system in an unsupervised data. Their results showed that the proposed methodology outperformed the traditional methods, such as...

The study of Liu et al. [38] improved PIP similarity measure, which belongs to Ahn [17]. First, the authors showed existing measures (Pearson, cosine, etc.) and PIP's disadvantages. Then, they upgraded PIP with a more straightforward measure known as the new heuristic similarity measure (NHSM). Authors noted that the PIP measure only considers the absolute value of the rating, repeatedly penalizes on the factors, and sometimes calculates inaccurate results. To increase accuracy, they considered the proportion of the common ratings in NHSM. NHSM showed improved performance than most of the compared methods regarding precision and recall. In a recent study of [39], the authors proposed a new similarity measure based on an adaptive neighbor selection mechanism and outperformed several widely-used methods.

Wang et al. [38, 39] conducted two studies that focused on new approaches to eliminate poor prediction accuracy with collaborative filtering. In the study of [40], the authors used MAE to compare their new approach named -Fuzzy Similarity Measure-User Relevant Aggregation (FSM-URA) with other approaches. They used MovieLens 100k dataset to test the proposed approach and found that the new method performed better than others did. In the research of [41], the same authors suggested a hybrid approach named novel entropy-based similarity measure. This new approach was compared with five different approaches, including item-based Pearson correlation coefficient and user-based Pearson correlation coefficient. Their hybrid approach produced better MAE results than others did in both the

MovieLens 100K dataset as well as the SmartBizSeeker dataset. They also took advantage of the Manhattan distance model to overcome the fat-tail problem that occurs due to ratings that are far from average. We recommend that future researchers examine [41] for further inquiry about the details of the hybrid model.

Thus far, the studies in the literature have proposed improved similarity measures to heighten the recommendation performance of collaborative filtering methods. The current study also suggests a new similarity measure, which can be an alternative to the existing ones. In contrast to those measures already existing in the literature, the proposed new similarity measure is based on how the users tend to rate close or far from the median. This idea originated from the issue that people may avoid giving extreme ratings [42]. In this study, avoiding extreme ratings was called the tendency towards rating close to the median. We chose this definition because when a person avoids giving the highest rating to an item that he or she likes a lot, this also represents that person's tendency towards giving a score close to the median. It is often proposed in the literature that extreme scores would have higher impact in the similarity, such that if two users rate five (the highest score in a 5-point Likert scale) this represents a more similar behavior [17], [38], [43]. In addition to that, the current study suggests that the strength of the rating should be determined when considering the rating behavior by observing all the ratings of that user. If a user tends to give a rating close to the median, and thereby avoids extreme rating, then a four (second highest score in a 5-point Likert scale) can also represent a strong preference. Considering this, in the current study, a novel algorithmic similarity measure that considers people's tendency towards rating close to the median is proposed and the advantages of the proposed similarity measures over the existing ones were analyzed.

3. PROPOSED SIMILARITY METHOD

The similarity method proposed in this study aims to minimize the disadvantages of the existing similarity measures. Several similarity methods that are prominent in the literature are chosen for benchmark. The results of the chosen similarity measures were calculated on a designated user-item matrix. The sample matrix can be seen in Table 1. In this section, firstly, explanation and formalization of the existing similarity measures were demonstrated. Then the existing similarity measures were calculated using the sample matrix. Based on the calculated similarity results, drawbacks of each existing similarity method were discussed. Then, the proposed similarity method calculation steps were explained. For each user combination in the sample matrix, the new similarity measure results were calculated. Based on the obtained results, advantages of the new similarity measure over the existing ones were demonstrated.

User set was described as $U = \{u_1, u_2, u_3, \dots, u_n\}$ and the item set was described as $P = \{p_1, p_2, p_3, \dots, p_m\}$. The user item matrix was represented as $R = \{r_{ij}\}$ where $i = 1, 2, 3, \dots, n$, $j = 1, 2, 3, \dots, m$ and r_{ij} is the rating of user i to the item j .

Table 1: A sample user-item matrix

	Item-1	Item-2	Item-3	Item-4
User-1	2	1	2	2
User-2	5	4	5	5
User-3	4	5	4	4
User-4	1	2	1	1

3.1. Traditional Similarity Measures and Their Drawbacks

To observe the disadvantages of the existing similarity measures, similarities were calculated based on the sample matrix. As seen in Table 1, the sample matrix consists of four users who gave ratings to four different items. Users were resembled in rows, while items were represented in columns. The intersection of the user row and item column resembles the rating that the selected user gave for the selected item.

In collaborative filtering, widely used similarity measures are Pearson Correlation Coefficient (PCC) and Cosine similarity (COS). In addition to these traditional methods, the constrained Pearson correlation [44] [45], was used to eliminate the disadvantages of PCC. In this section, these three similarity measures and their drawbacks are discussed. The notation used in the formalization of the similarity measures are shown below:

C	Commonly rated items by the selected users
u_a, u_b	Users of a and b
$r_{u,j}$	The rating of the user u to the item j
\bar{r}_u	The average rating value of the user u
r_{med}	Median score in the rating scale

3.1.1. Pearson Correlation Coefficient

The Pearson correlation coefficient shows how the two users use similar patterns while giving ratings to the items. The formalization of the Pearson correlation is as follows:

$$PCC(u_a, u_b) = \frac{\sum_{j \in C} (r_{u_a, j} - \bar{r}_{u_a})(r_{u_b, j} - \bar{r}_{u_b})}{\sqrt{\sum_{j \in C} (r_{u_a, j} - \bar{r}_{u_a})^2} \sqrt{\sum_{j \in C} (r_{u_b, j} - \bar{r}_{u_b})^2}} \quad (1)$$

PCC similarity scores of all the pairs in the sample matrix are shown in Table 2. While identifying how the users have similar patterns in ratings is important, this pattern may sometimes be misleading and direct to wrong similarities. When the ratings of User1 and User2 were observed in Table 1, it is seen that User1 and User2 rated the items as (2,1,2,2) and (5,4,5,5) respectively. Based on the scores of the User1 and User2, a low similarity was expected between them because while User1 dislikes all the items, User2 likes all of them. However, Table 2 shows that PCC score of User1 and User2 is +1, representing a perfect correlation. PCC only considers the scoring behavior of the two users and does not consider the meaning of the score. A similar situation was observed for User2 and User3. User2 rated the items as (5,4,5,5), while User3 rated the items as (4,5,4,4). While high similarity is expected between User2 and User3, PCC scores for the two users were calculated as -1. This misleading PCC similarity scores were also observed for other pairs, as seen

in Table 2. Such misleading similarity scores may lead to deficiencies in collaborative filtering.

Table 2. Similarity scores based on sample user-item matrix

Pair	Similarity Scores				
	PCC	COS	CPC	PIP	NHSM
User1 User2	1	0.9885	-3.6689	6	0.0174
User1 User3	-1	0.9414	-3.5714	18.861	0.0078
User1 User4	-1	0.8386	3.6689	3822	0.0216
User2 User3	-1	0.9815	3.6689	3822	0.0216
User2 User4	-1	0.9112	-3.7692	6.5830	0.0087
User3 User4	1	0.9732	-3.6689	6	0.0078

3.1.2. Cosine Similarity

Cosine calculates similarity by observing how magnitudes of the rating vectors differ from each other [46]. Cosine similarity (COS) is calculated as follows:

$$COS(u_a, u_b) = \frac{\sum_{j \in C} r_{u_a, j} * r_{u_b, j}}{\sqrt{\sum_{j \in C} r_{u_a, j}^2} \sqrt{\sum_{j \in C} r_{u_b, j}^2}} \quad (2)$$

COS results for the user pairs in the sample matrix is given in Table 2. COS found the most similar pairs as User1 and User2. Scores of User1 and User2 were (2,1,2,2) and (5,4,5,5) respectively, indicating different preferences. While expecting a low similarity, Cosine found the highest similarity with the score of 0.988 between User1 and User2. Similarly, a high similarity was expected between User1 and User4 with the ratings of (2,1,2,2) and (1,2,1,1) respectively. However, COS found 0.838, which is the lowest score for these two users. These results indicate that COS can lead to misleading similarities in certain conditions. Another drawback of COS is that it proposes a narrow range between the lowest and the highest score, which is 0.8386 – 0.9885. This may lead to problems in differentiating similar and dissimilar users.

3.1.3. Constrained Pearson Correlation

The Constrained Pearson Correlation (CPC) was proposed as an alternative to PCC. It takes into account positive and negative attitudes of the user towards the item by choosing a threshold to decide the direction of the attitude [47]. Therefore, CPC considers the absolute value of the rating. In other words, it considers whether the user liked or disliked the item [47], [48]. In our study, we choose the median as the threshold to decide the direction of the attitudes of the users. Expression of CPC is as follows:

$$CPC(u_a, u_b) = \frac{\sum_{j \in C}(r_{u_a, j} - r_{med})(r_{u_b, j} - r_{med})}{\sqrt{\sum_{j \in C}(r_{u_a, j} - r_{med})^2} \sqrt{\sum_{j \in C}(r_{u_b, j} - r_{med})^2}} \quad (3)$$

As can be noted from Table 2, CPC seems to overcome the drawbacks of PCC. CPC found the lowest similarity score for the pair of User2 and User4. This result is acceptable because User2 and User4 show different preferences. CPC found the highest similarity score for the pairs of (User1, User4) and (User2, User3). These results are consistent with the preferences of these pairs. Prediction performance of CPC should be observed with a broader data to decide how efficient CPC is in collaborative filtering.

3.2. Recently proposed similarity methods

In the real-life user-item preference matrices, users give ratings to only a few items among all the items. This problem is referred to as cold start problem in collaborative filtering literature. The problems with the traditional similarity methods, as stated in Section 3, become even more severe when the data sparsity levels are high [17]. To overcome this shortcoming, the researchers proposed different similarity metrics. Two of the most prominent similarity measures proposed recently are Proximity-Impact-Popularity (PIP) [17] and New Heuristic Similarity Method (NHSM) [38]. The details of the two popular similarity methods are explained below.

3.2.1. Proximity – Impact – Popularity (PIP) Similarity

The PIP similarity measure proposed by Ahn [17] embraces three factors: proximity, impact, and popularity of user ratings. PIP is computed as follows:

$$SIM_{PIP}(u_a, u_b) = \sum_{j \in C} PIP(r_{u_a, j}, r_{u_b, j}) \quad (4)$$

Where for the two ratings $PIP(r_1, r_2)$ is calculated as:

$$PIP(r_1, r_2) = Proximity(r_1, r_2) * Impact(r_1, r_2) * Popularity(r_1 * r_2) \quad (5)$$

The *proximity* of the two ratings demonstrates the mathematical difference between two ratings. It also gives a penalty to ratings in disagreement by increasing the distance. The *impact* factor represents how strong the two users' preferences are about the item. When both users give the highest score or the lowest score to the item, this shows a stronger common preference, explaining a high impact score. Further lowering the impact score when the two ratings disagree also penalizes impact score. *Popularity* measures how the two ratings are far from the mean of all the ratings given to that item. Popularity computes a higher score when the two ratings are both further from the average in the same direction. Further information about three factors and how they are calculated can be found in [17].

PIP measure has several drawbacks. Firstly it repeatedly penalizes ratings on disagreement [38]. This extra penalization can lead to increased deviation in similarity results. PIP similarity scores demonstrate a vast range between the lowest as well as the highest score. When the results of the PIP measure on the sample matrix is observed in Table 2, lowest score is 6 while the highest score is 3822. High deviation in similarity results may lead misleading

predictions. Secondly, the PIP measure does not consider the tendency to rate close to the median. The impact factor of PIP considers how strong the user's preferences is by finding agreement on the extreme values on the rating scale. However, impact does not consider overall rating behavior of the users about how they tend to rate close to the median while deciding stronger preference level.

3.2.2. The New Heuristic Similarity Measure (NHSM)

The new heuristic Similarity Method (NHSM) was proposed by Liu et al. [38] as an improvement to the PIP measure. Proximity-Significance-Singularity (PSS), a revised version of PIP, is used as a factor in NHSM. Proximity only calculates the absolute distance and does not change when there is a disagreement on the item between the two users. Significance measures how the two ratings are different from the median. Singularity measures how the two ratings deviate from the average score of the item. PSS is calculated as follows:

$$PSS(r_1, r_2) = Proximity(r_1, r_2) * Significance(r_1, r_2) * Singularity(r_1 * r_2) \quad (6)$$

After calculating the PSS measure, the proportion of common ratings is added as a factor in the NHSM similarity. In this factor, to penalize the low number of common ratings, the combination of user ratings is calculated by multiplying the number of ratings of the users. It is necessary to indicate that real value of common ratings as proposed in Jaccard measure [30] is modified in NHSM measure.

While significance considers deviation from the median, it does not reflect the effect of the user's tendency towards scoring around the median. NHSM uses another factor named URP to consider rating behavior of the users. However, when the formalization of URP is observed in [38], URP only resembles how the ratings of users differentiate from the mean value. This formalization does not include the effect of the user's tendency about scoring close to the median. For a detailed description of how NHSM is calculated, Liu et al. [38] should be observed.

3.2.3. The reasons for proposing a new similarity measure

As described in the preceding section, traditional similarity measures have some drawbacks. While recently proposed measures of PIP and NHSM have been proven to be effective, alternatives to the traditional methods such as PIP and NHSM have some deficiencies. The deficiencies in these similarity measures, and how the proposed novel algorithmic similarity measure will handle the stated deficiencies as stated in the following part.

PIP measures uses common items while calculating similarity, but it does not consider the effect of the proportion of commonly rated items. This may lead to misleading similarity results such that a pair who have one common item may be treated on par with a pair who has twenty items in common.

While NHSM considers the proportion of common items, the way it calculates the proportion of these items has some drawbacks. Let User A and User B be the pair for similarity calculation. NHSM uses a revised version of Jaccard and divides the number of common items by the product of the

number of rated items of User A and User B, as shown in this equation:

$$Jaccard_{nhsm} = \frac{|n(A \cap B)|}{n(A) * n(B)} \quad (7)$$

Where $A \cap B$ is the set of commonly rated items of User A and User B. $n(A)$ and $n(B)$ represent the number of items rated by User A and User B respectively. NHSM version of Jaccard calculation not only penalizes users who have less common items, but also is disadvantageous to users who have a high amount of common items. The proportion of common items for the users who have a high number of common items will lower the value of proportion since the denominator will be high. In our study, to protect the accuracy of the proportion of common ratings original Jaccard formula were used.

Adopting the idea that some people tend to avoid scoring extreme values [42], our study argues that some people do not prefer to give high ratings, although they liked the item a lot. For example, a person who loves a movie may give four to the item. They may think that the film should have extraordinary features to get a score of five. This situation may be correct in the reverse direction. A person who extremely dislikes a movie may rate the item as two instead of one. We refer to this behavior as the *tendency to rate close to the median*. We consider this tendency of the users in the novel algorithmic similarity measure calculation.

Neither PIP nor NHSM consider the tendency to rate close to the median. In NHSM, a factor named user-rating preference (URP) is considered to reflect the preference to rate high or low. However, URP does not measure users' tendency to rate close to the median. It represents how the users' ratings are different from each other. It is a product of the mean difference as well as the standard deviation difference between users, calculated as such:

$$URP_{nhsm}(u_a, u_b) = 1 - \text{sigmoid}(|\mu_a - \mu_b| * |\sigma_a * \sigma_b|) \quad (8)$$

where the Sigmoid function is calculated as follows:

$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)} \quad (9)$$

While URP calculates how the mean and the standard deviation of two users differentiate from each other, URP may not be sufficient in considering how the users avoid rating the highest/lowest score or tend to rate close to the median.

In this study, while calculating proximity between the users' ratings, the *tendency to rate close to the median* is considered to reflect proximity value more accurately. *Singularity* is calculated as how each user's rating is different from the mean ratings of that item. In addition to proximity and singularity, Vector Space Similarity (VSS) is adapted to calculate user similarity. This algorithm is chosen because, in VSS, values that represent user ratings can be calculated to reflect how each user's score is different from the median. With adopting VSS in the new similarity measure, not only do the proximity value represent the *tendency to rate close to the median*, but the overall similarity measure considers how user's ratings are different from the median. Details about calculating the novel algorithmic similarity measure are explained in the next section.

3.3. Expression of the Novel Algorithmic Similarity Measure

3.3.1. The Notation

In this section, we give a mathematical formula to calculate the new similarity method, which we named **the Novel Algorithmic Similarity (NAS)**. We implemented the sigmoid function while calculating first part of the similarity known as the Proximity –Singularity (PS). The sigmoid function not only normalizes the results, but it also rewards good similarity and punishes the bad one [38]. Initial similarity measure PS was calculated as follows:

$$PS(r_1, r_2) = \text{Proximity}(r_1, r_2) * \text{Singularity}(r_1, r_2) \quad (10)$$

PS for each rating pair is calculated by multiplying proximity and singularity of the two ratings. Proximity in PS is different from proximity calculation in PIP and NHSM. Before we calculate proximity for PS, we calculate a user factor that resembles the tendency to rate close to the median for each user. Small average absolute deviation from the median means the user has a higher tendency to rate close to the mean. Reversely, high deviation from the median shows the user has lower tendency to rate close to the mean. Therefore, to compute user factor, average absolute deviation from the median is subtracted from 1 after it is normalized by the sigmoid function. User factor (UF) is calculated as follows:

$$UF_u = 1 - \text{sigmoid}\left(\frac{\sum_{r \in A} |r - r_{med}|}{n(A)}\right) \quad (11)$$

where A is the set of user u's ratings.

Since UF resembles tendency to rate close to the median, when the user rated above the median, UF should be added to the score to reflect possible rating preference of the user. This sum will resemble the score of the user if the user does not have any tendency of rating close to the median. Similarly, if the user rated below the median, UF should be subtracted from the user's rating to make the rating free from the tendency towards the median. Calculation of the scores with adopting UF is as follows:

$$\begin{aligned} s_{u,j} &= r_{u,j} + UF_u, & \text{if } r_{u,j} > r_{med} \\ s_{u,j} &= r_{u,j} - UF_u, & \text{if } r_{u,j} < r_{med} \\ s_{u,j} &= r_{u,j}, & \text{if } r_{u,j} = r_{med} \end{aligned} \quad (12)$$

where $s_{u,j}$ is the updated score of the user, u for the item, and j after adopting UF_u .

The proximity is calculated as the absolute value of the distance between the updated scores as follows:

$$\text{Proximity}(s_{u_a,j}, s_{u_b,j}) = 1 - \text{sigmoid}(|s_{u_a,j} - s_{u_b,j}|) \quad (13)$$

Singularity resembles how each users' rating is different from the average rating for that item. Different from [38], we consider deviation from the mean separately for each score and then multiply them. This makes the singularity value stronger by empowering the scores, which are far from the mean rating of the item. Singularity of two ratings is calculated as follows:

$$\text{Singularity}(r_{u_a,j}, r_{u_b,j}) = \text{sigmoid}(|r_{u_a,j} - \bar{r}_j| * |r_{u_b,j} - \bar{r}_j|) \quad (14)$$

If the two users rated an item as five, where the average rating on the Likert scale is three, the singularity value of the

two ratings will have a high impact on the similarity score. Similarly, if a user rates an item five, and another user rates it as one, the singularity of these two scores will have a high impact on the similarity in a negative direction because the VSS considers the direction of the relationship (as will be seen in the following section).

When a rating is far from the median and close to the extreme, this reflects a stronger preference. In a similarity measure, it is necessary to consider that if two scores are far from the median and close to the extremes, these scores should have a stronger impact on the similarity between users [17], [19]. This issue resembles the significance of the scores [27]. For two users, let the first pair of scores be (5,5) and the second pair of scores be (2,3). The first pair indicates a higher similarity between the users than the second pair. To reflect the significance of the scores, we can use the Surprisal-based Vector Similarity method (SVS). Vector similarity method uses a maximum likelihood estimator, which refers to the average attitude of the scores [19]. The score that is far from the maximum likelihood estimator has more significance on the similarity.

In Luo et al. [19], the mean score of the item was used as the maximum likelihood estimator. In our study, to calculate SVS, we use the median as the maximum likelihood estimator. With this estimation, users who rate on the extremes will represent a strong preference and as well as have a high similarity between them. SVS similarity is calculated using the surprisal vector of the user rather than the actual rating vector. Surprisal vector of a user is calculated as follows [19]:

$$S_a = [s_{a,1}, s_{a,2}, s_{a,3}, \dots, s_{a,z}]$$

$$s_a = [\text{sign}(r_{u_{a,1}} - r_{med}) * I(r_{u_{a,1}}), \dots, \text{sign}(r_{u_{a,z}} - r_{med}) * I(r_{u_{a,z}})] \quad (15)$$

where Z is the set of items user a has rated, $\text{sign}(r_{a,1} - r_{med})$ means the sign of the notation, $r_{a,1} - r_{med}$ is positive or negative, and $I(r_{a,1})$ is defined as the quantity of information and calculated as follows:

$$I(r_{u_{a,j}}) = \ln(2\hat{b}_j) + \frac{|r_{u_{a,i}} - r_{med}|}{\hat{b}_j} \quad (16)$$

where \hat{b}_j is defined as a scale parameter and calculated as follows:

$$\hat{b}_j = \frac{1}{N} \sum_{j=1}^N |r_{u_{i,j}} - r_{med}| \quad (17)$$

where N is the set of all users.

After calculating the surprisal vector of users, Surprisal-based Vector Similarity (SVS) coefficient is computed to calculate the similarity between two users as shown in follows:

$$SVS(u_a, u_b) = \frac{\sum_{j \in C} s_{a,j} * s_{b,j}}{\sqrt{\sum_{j \in C} s_{a,j}^2} \sqrt{\sum_{j \in C} s_{b,j}^2}} \quad (18)$$

To calculate the proportion of common ratings we used the Jaccard formula. It is important to calculate the exact

proportion of common ratings to see how they affect the similarity score. Jaccard similarity is calculated as follows:

$$Jaccard(u_a, u_b) = \frac{n(A \cap B)}{n(A \cup B)} \quad (19)$$

Finally, all the factors in the similarity measure are represented in the Novel Algorithmic Similarity (NAS) measure, which is calculated by multiplying the equations, (10), (18) and (19) as follows:

$$NAS(u_a, u_b) = PS(u_a, u_b) * SVS(u_a, u_b) * Jaccard(u_a, u_b) \quad (20)$$

3.3.2. Discussion on the Novel Algorithmic Similarity Measure

The Novel algorithmic similarity measure was applied to the sample matrix on Table 1 to observe how it overcame the drawbacks of the existing similarity measures, as discussed in Section 3.1 and 3.2. As Table 3 shows, the highest similarity based on NAS was observed between users 1-4 and users 2-3, as expected based on the ratings. However, PCC and COS did not accurately identify the most similar users. According to PCC, similarity of users 1-4 and users 2-3 were negative. According to COS, users 1-4 were the least similar pair. These results indicate that NAS overcame the drawbacks of PCC and COS when it came to misleading similarity score regardless of the similar preferences.

As observed in Table 3, NAS correctly identified the least similar users as users 1-2, users 1-3, users 2-4 and users 3-4. NAS accurately determined the direction of the relationship between the least similar users as negative. However, in COS, PCC, PIP, and NHSM, direction identification was either wrong or not applied. PCC incorrectly identified the direction of the relationship between users 1-2 and users 3-4. The other similarity measures, with the exception of CPC, did not show the direction of the relationship. NHSM similarity between users 1-3 was found relatively high and it is difficult to say whether users 1-3 had a high or low similarity based on the overall scores of NHSM.

As seen in Table 3, NAS correctly identified the most and the least similar users based on their ratings. NAS also correctly determined the direction of the similarity between the users. These features make NAS an appropriate similarity measure that could be used as an alternative to other similarity measures in collaborative filtering.

Table 3. NAS results for the sample matrix

Pair	NAS Similarity
User1 – User2	-0.1126
User1 – User3	-0.1863
User1 – User4	0.9376
User2 – User3	0.9376
User2 – User4	-0.1064
User3 – User4	-0.1126

To summarize, the steps taken to calculate the algorithm for the NAS measure for a user pair is as follows:

- 1- The user factor, based on the tendency of scoring close to the mean, is calculated for each user.

- 2- The proximity between scores is calculated with adapting the user factor, based on equation (13).
- 3- The singularity between the scores is calculated based on equation (14).
- 4- The PS score for the user pair is calculated by summing the multiplication of the proximity and the singularity scores of the commonly rated items.
- 5- The SVS similarity score for the user pair is calculated based on the equation (18) where maximum likelihood estimator is the median.
- 6- The commonly rated items for the user pair is calculated based on the Jaccard formula in equation (19).
- 7- The NAS measure for the user pair is calculated by multiplying the PS, SVS and Jaccard scores.

4. EXPERIMENTS

4.1. Data Set

The MovieLens Dataset of ML-100K and FilmTrust datasets were used in our experiments. The MovieLens Dataset of ML-100K was prepared by the GroupLens Research at the University of Minnesota [4]. In ML-100K, there were 943 users and 1682 movies with a total of 100 000 ratings. Each user rated at least 20 movies out of 1682 movies. Ratings changed between 1 and 5, where 1 represented the lowest score for the preference and 5 depicted the highest score. This made the ML-100K dataset very sparse with %6.3 density. FilmTrust is a movie-rating website, wherein the users give ratings for the selected movies. This dataset was prepared by Guo et al. [49]. In the FilmTrust dataset, there were 1508 users and 2071 movies with a total of 35500 ratings. Ratings change between 0.5 and 4 and the sparsity level of FilmTrust was % 1.14.

To observe the performance of the proposed novel algorithmic similarity (NAS) measure, we compared the proposed measure with the traditional methods of COS, PCC, CPC and the recent methods of PIP and NHSM. Recommendations were conducted based on each similarity method, and the performance of each similarity method was compared based on selected evaluation metrics of MAERMSE, recall, and precision.

For each experiment, 70% of users in the dataset were selected as training users for similarity calculation. The remaining 30% of users were testing users for whom the recommendations were conducted. For each testing user, 70% of their rated items were selected as training ratings for similarity calculation, while 30% of the rated items were used for conducting the actual recommendations.

4.2. Performance metrics

After calculating the user based similarities, the predicted ratings of an item for a specific user was calculated using the the following formula [19]

$$\hat{r}_{u_t,p} = \frac{\sum_{i \in K} sim_{u_t, u_i} r_{u_i,p}}{\sum_{i \in K} |sim_{u_t, u_i}|} \quad (21)$$

where sim_{u_t, u_i} resembles the similarity between the test user u_t and his neighbor u_i . K denotes the set of most similar neighbors of the test user u_t .

With the above formula, predicted ratings were calculated for each similarity measure framework. To compare the performance of each similarity measure, prediction accuracy was calculated with MAE. As stated before, the MAE is a commonly used metric to calculate how the predicted values differ from the real values [14]. MAE is calculated as follows:

$$MAE = \frac{\sum_{j=1}^N |r_j - \hat{r}_j|}{N} \quad (22)$$

Another commonly used metric to measure how the predicted values are distant from the real values is RMSE. RMSE calculates the averaged squared distances between the real ratings and the predicted ratings[14]. RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{\sum_{j=1}^N (r_j - \hat{r}_j)^2}{N}} \quad (23)$$

Besides measuring prediction accuracy of the selected algorithms, we also wanted to measure the classification accuracy of the proposed frameworks. Classification accuracy metrics tolerate deviations from the actual ratings when they measure how the items are truly classified as recommended [50]. Recall and precision are the classification accuracy metrics, which were also used in our research. . Recall is defined and calculated as follows [51]:

$$Recall = \frac{\text{number of testing items liked by the testing user and assigned to recommended list}}{\text{number of testing items actually liked by the testing user}} \quad (24)$$

Precision is defined and calculated as follows [51], [52] :

$$Precision = \frac{\text{number of testing items liked by the testing user and assigned to recommended list}}{\text{number of testing items assigned to recommended list}} \quad (25)$$

For performance measures, smaller MAE values indicate better prediction accuracy while higher recall and precision scores indicate better classification accuracies.

4.3. Experimental Design

To compare the proposed NAS algorithm with the other selected similarity algorithms, we build several configurations of the chosen parameters. We altered the number of nearest neighbors in our experiments. The number of nearest neighbors (K) is a fundamental parameter in collaborative filtering, and it affects the prediction performance [53]. We identified eight levels for the number of nearest neighbors and calculated the MAE, RMSE, recall and precision values for each level using the proposed similarity measure NAS as well as the other selected similarity measures of COS, CPC, PCC, PIP, and NHSM.

5. Results and Discussion

In this section to compare the performance of the proposed NAS measure, several experiments were conducted, using two datasets that were the MovieLens ML-100k and FilmTrust. Performance comparison were based on prediction accuracy and classification accuracy. Prediction accuracy was measured using the metrics of MAE and RMSE, while classification accuracy was measured using Recall and Precision. We compared the results of each of the

performance metric based on the pre-determined levels of the numbers of the nearest neighbor. First, we presented the results for prediction accuracy. Then, we presented the classification accuracy performance of predicted values.

5.1. MAE and RMSE results for prediction accuracy

As K denotes different levels for the number of nearest neighbors, we analyzed the impacts of the different number of nearest neighbors on prediction accuracy measured with MAE and RMSE. Figure 1 shows the MAE results of the similarity measures used with six levels of the number of nearest neighbors for the MovieLens 100k (ml-100k) dataset. Figure 1a shows that except Cosine, for all similarity measures MAE decrease as K increase from 30 to 80. We observed that the novel algorithmic similarity (NAS) measure reveals lower MAE values than the most of the other similarity measures in all levels of the number of nearest neighbors. When K is 20, 30 and 40 NAS obtains the lowest MAE among all of the similarity measures. When K is 10, 60 and 80, NAS gets the second lowest MAE after NHSM measure.

Figure 1b shows the MAE results for the FilmTrust dataset. Figure 1b shows that for all similarity measures MAE decrease as K increase from 10 to 20. NAS measure produces lower MAE values than the other similarity measures except for the levels when the number of nearest neighbors are 80 and 60. When K is 10, 20, 30 and 40 NAS obtains the lowest MAE among all of the similarity measures. When K is 60 and 80, NAS gets the second lowest MAE after NHSM measure after PIP.

RMSE results for MovieLens 100k for all the measures with six levels of K can be observed in Figure 2a. NAS is the best similarity measure among the six similarity measures based on RMSE because NAS is the only measure that provided the best performance in the four levels out of six levels of K . When K is 20, 30, 40, and 60 NAS provides the lowest RMSE score. When K is 10, NAS shows the second lowest score with PIP after NHSM. When K is 80 NAS comes third after NHSM and PIP. RMSE results for FilmTrust can be observed in Figure 2b. NAS provided the best performance in the four levels out of six levels of K . When K is 10, 20, 30, 40 NAS produces the lowest RMSE score. When K is 60 and 80, NAS shows the second lowest score after PIP.

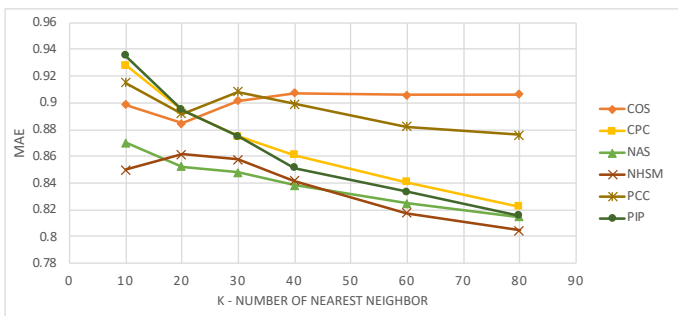


Figure 1a. MAE results of Similarity measures in different K nearest neighbors for MovieLens 100k

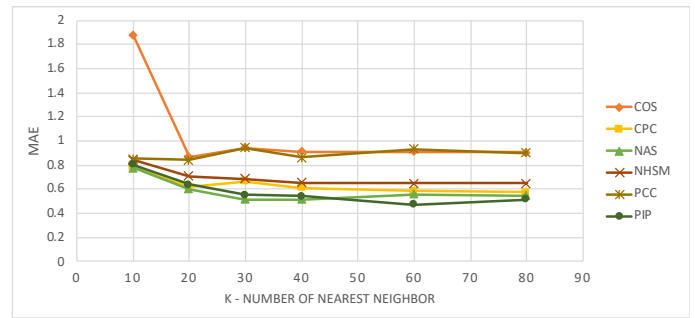


Figure 1b. MAE results of Similarity measures in different K nearest neighbors for FilmTrust

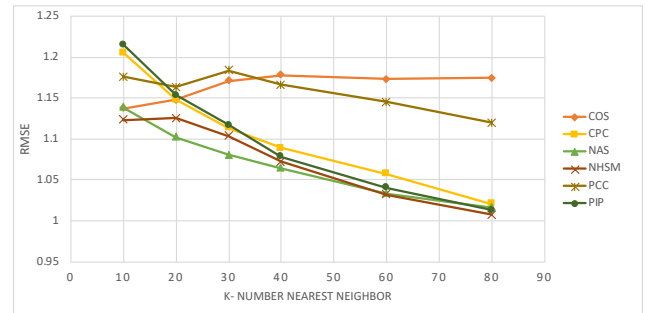


Figure 2a. RMSE results in different K nearest neighbors for MovieLens 100k

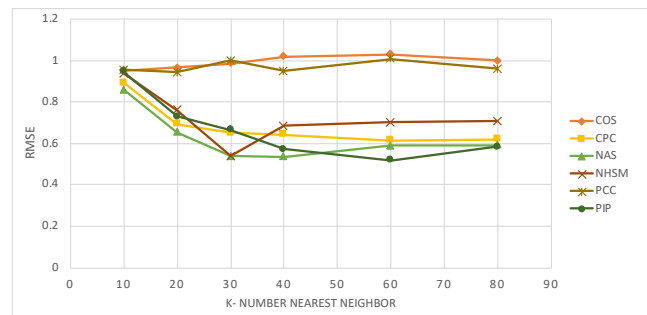


Figure 2b. RMSE results in different K nearest neighbors for FilmTrust

Concerning prediction accuracy, our newly proposed similarity measure NAS has outstanding performance when compared to the other similarity measures. It showed lower MAE score than all of the traditional similarity measures of COS, CPC, and PCC in all of the levels of K for both of the two datasets. For the ml-100k dataset, NAS obtained lower MAE from the recently proposed popular similarity measure PIP in five out of six levels of K . For three levels of K , NAS comes first meaning showing the best prediction accuracy while providing better results than the NHSM and PIP. In the lowest number and the two highest numbers of nearest neighbors(K), NAS comes second best after NHSM. Regarding RMSE for ml-100k, in the lowest and the highest

K, NAS comes second after NHSM. For the FilmTrust dataset, NAS brought lower MAE and RMSE levels for all number of nearest neighbors but the two highest level of number of nearest neighbors. In the two highest number of nearest neighbors PIP is the only measure that showed lower MAE and RMSE results. This means NAS showed better performance than the NHSM for all the levels.

The reason for NAS coming second in the lowest and highest K is related with the similarity calculation based on the number of nearest neighbors. When K is small like 10, some similar users may be left out to calculate accurate similarity scores between the users. When K is high like 80, some false neighbors may be identified. False neighbor is a user that is identified as a neighbor but in fact he or she is not similar to the interested user. To summarize, using low number of neighbors may result in leaving some similar users out of the group and using high number of neighbors may result in bringing false neighbors inside the group. In both situations, neighbor group may not accurately represent the similar users. This may be the reason for NAS not being the best similarity measure regarding prediction accuracy when K is the lowest and the highest. When K is between 20, 3 and 40, a more representative similar neighbor group can be established to make healthier predictions and this may lead the NAS being the best measure in the given K interval.

5.1. Recall and Precision results for classification accuracy

We measured classification accuracy with the metrics of recall and precision. We firstly analyzed the impacts of the different K on classification accuracy. The results of the analysis are given in Figures 3 and 4 for precision and recall respectively. As Figure 3a shows, for ml-100k regarding recall scores the newly proposed similarity measure NAS showed remarkable results among the six similarity measures. Recall of NAS was higher than the three of the metrics which are Cosine, PCC, and NHSM. The highest recall is observed when K is 20 and 30 for CPC, PIP, NAS, and NHSM. For these similarity measures recall gets lower as K increases. Only in Cosine and PCC recall increased as K increased. The two mostly used traditional similarity measures Cosine, and PCC shows the lowest recall score among all measures in all K. CPC revealed the highest recall values in all the K while PIP comes second and NAS comes third.

For the FilmTrust dataset, as Figure 3b shows, number of levels has a huge impact on the recall results. As number of nearest neighbor increases Recall decreases for all the measures. While K is 10 NAS comes third after PIP and CPC. When K is 20, NAS is the second after CPC. When K is 30, NAS comes second after NHSM. When K is 40 NAS is has the highest recall results. When K is 60 NAS is the third after NAS and NHSM. Finally when K is 80 NAS is the second after COS.

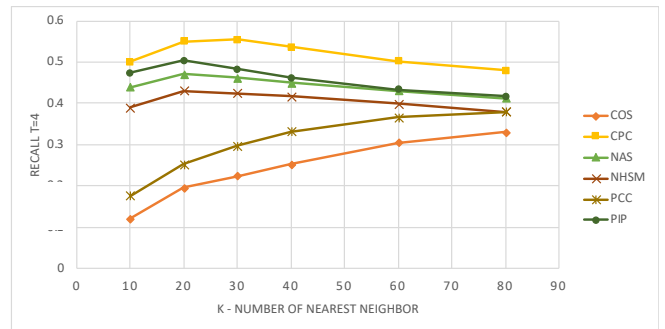


Figure 3a. Recall results for ml-100k

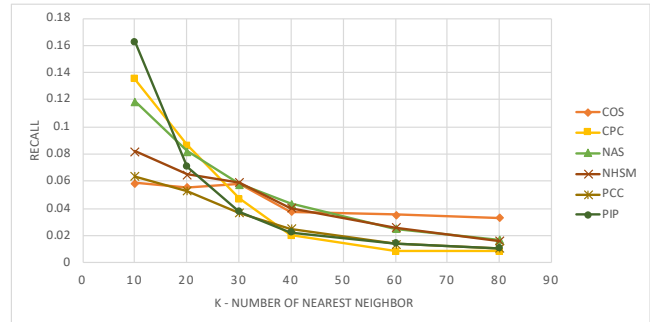


Figure 3b. Recall results for FilmTrust

Precision values for ml-100k for all K values when the score threshold is 4 can be observed in Figure 4a and 4b. Precision of NAS shows remarkable results. As shown in Figure 4a, for ml-100k, in the three lowest K which are 10, 20 and 30 NAS showed the second-best precision score. In all levels of K, NHSM revealed the best precision score. Except for Cosine and PCC, precision values are decreased as K decreased. Cosine and PCC showed stable values in all K, while Cosine revealed the lowest precision scores in the first four levels of K and PIP obtained the lowest precision score when K is 60 and 80. Precision results also show that as K increases, precision score of all values come close to each other. Especially after 40 precision values of the similarity measures are very close to each other. This result indicates that as the number of nearest neighbors increase false neighbors increase. Increasing number of false neighbors decrease the ability of precision measure to differentiate similarity measures from each other.

For FilmTrust dataset, precision score decreases as K increases for all similarity measures. When K is 10, NAS showed highest precision results. When K is 20 and 30 NAS is the second after CPC. When K is 40 NAS is second after NHSM. When K is 60 NAS is the third after COS and NHSM. When K is 80 NAS is the second after COS. Although NAS was not the best measure for all the levels of K, it showed higher precision scores than the majority of the similarity measures for all the levels of K.

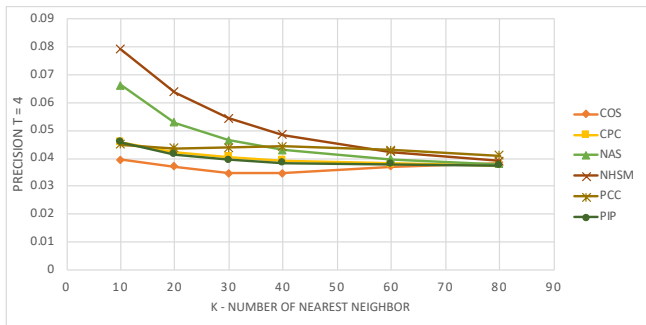


Figure 4a. Precision results different K nearest neighbors for ml-100k

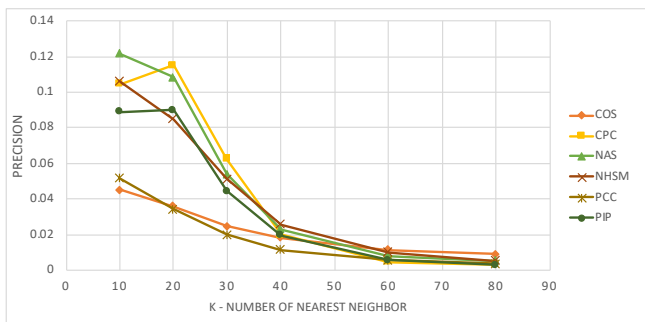


Figure 4b. Precision results different K nearest neighbors for FilmTrust

To summarize the results, we can conclude that proposed Novel Algorithmic Similarity (NAS) measure can provide better prediction and classification accuracy performance than most other tested similarity measures as can be seen from Figure 1a to Figure 4b. NAS provided best results when the number of nearest neighbors is average like 20, 30 and 40. Average number of nearest neighbors result in a more accurate similar neighbor set. Regarding prediction accuracy, NAS can outdo all of the tested similarity measures in the all but the two extreme levels of the number of nearest neighbors. Concerning classification accuracy, NAS comes second or third best among all the proposed measures in the whole number of nearest neighbors. The above results demonstrate that the newly proposed NAS is an effective similarity measure in collaborative filtering and can be an excellent alternative to the similarity measures used in the literature. NAS demonstrated a remarkable performance because it distinguishes users not only how they score different from each other, it also considers how the users are different based on their tendencies towards scoring close to the median.

6. CONCLUSIONS

This research proposes a new similarity model that can compete with the popular similarity measures in the literature. For comparison with the proposed measure, five similarity measures were chosen from the literature. Firstly, the main drawbacks of the five similarity measures were stated using a sample matrix. The stated similarity measures had problems in genuinely identifying similar users. To overcome the drawbacks of the existing similarity measures, the novel algorithmic similarity (NAS) measure was used. NAS distinguishes similar users by considering the tendency towards scoring close to the median, which is then used as

the maximum likelihood estimator for the suppose vector similarity. To demonstrate the effectiveness of the proposed NAS measure, several configurations of the nearest neighbors and score threshold for classification accuracy were used in our experiments. Results reveal that the novel algorithmic measure yields better prediction accuracy performance than other similarity measures in almost every level of the nearest neighbors. The novel algorithmic similarity measure also showed better performance in terms of classification accuracy than most of the other similarity measures when the threshold is the score of four. These findings indicate that the proposed NAS measure can overcome the shortcomings of the existing measures and be a strong competitor to the similarity measures used for collaborative filtering.

REFERENCES

- [1] S. Davis and L. Toney, "How Coronavirus (COVID-19) Is Impacting Ecommerce," *Roi Revolution*, 2021. .
- [2] M. Z. Fisher, "Why Product Reviews are Important for Buyers and Sellers | ShipStation," *ShipStation*, 2018. [Online]. Available: <https://www.shipstation.com/blog/product-reviews-important-buyers-sellers/>. [Accessed: 12-Nov-2021].
- [3] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003, doi: 10.1109/MIC.2003.1167344.
- [4] F. M. Harper and J. A. Konstan, "The MovieLens Datasets," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, pp. 1–19, 2016, doi: 10.1145/2827872.
- [5] E. Rich, "User modeling via stereotypes," *Cognitive Science*, vol. 3, no. 4, pp. 329–354, 1979, doi: 10.1016/S0364-0213(79)80012-9.
- [6] P. Lops, M. De Gemmis, and G. Semeraro, "Content-based Recommender Systems: State of the Art and Trends," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Boston: Springer, 2011.
- [7] W. W. Cohen and W. Fan, "Web-collaborative filtering: recommending music by crawling the Web," *Computer Networks*, vol. 33, no. 1, pp. 685–698, 2000, doi: 10.1016/S1389-1286(00)00057-8.
- [8] A. B. Barragáns-Martínez, E. Costa-Montenegro, J. C. Burguillo, M. Rey-López, F. A. Mikic-Fonte, and A. Peleteiro, "A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition," *Information Sciences*, vol. 180, no. 22, pp. 4290–4311, 2010, doi: 10.1016/j.ins.2010.07.024.
- [9] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 43–52, 1998, doi: 10.1111/j.1553-2712.2011.01172.x.
- [10] H. Ma, I. King, and M. R. Lyu, "Effective missing data prediction for collaborative filtering," in *SIGIR '07 Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007, pp. 39–46, doi: 10.1145/1277741.1277751.

- [11] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992, doi: 10.1145/138859.138867.
- [12] K. Y. Goldberg and T. M. Roeder, "Eigentaste: A Constant Time Collaborative Filtering Algorithm," *CEUR Workshop Proceedings*, vol. 1225, no. July, pp. 41–42, 2001, doi: 10.1023/A.
- [13] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1273–1284, 2014, doi: 10.1109/TII.2014.2308433.
- [14] M. D. Ekstrand, "Collaborative Filtering Recommender Systems," *Foundations and Trends® in Human-Computer Interaction*, vol. 4, no. 2, pp. 81–173, 2011, doi: 10.1561/1100000009.
- [15] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005, doi: 10.1109/TKDE.2005.99.
- [16] R. Jin, J. Y. Chai, and L. Si, "An automatic weighting scheme for collaborative filtering," *Proceedings of the 27th annual international conference on Research and development in information retrieval - SIGIR '04*, pp. 337–344, 2004, doi: 10.1145/1008992.1009051.
- [17] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Information Sciences*, vol. 178, no. 1, pp. 37–51, 2008, doi: 10.1016/j.ins.2007.07.024.
- [18] Y. C. Cai, H. Leung, Q. Li, H. Min, J. Tang, and J. Li, "Typicality-Based Collaborative Filtering Recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 3, pp. 766–779, 2013, doi: 10.1109/TKDE.2013.7.
- [19] H. Luo, C. Niu, R. Shen, and C. Ullrich, "A collaborative filtering framework based on both local user similarity and global user similarity," *Machine Learning*, vol. 72, no. 3, pp. 231–245, 2008, doi: 10.1007/s10994-008-5068-4.
- [20] B. Zhang and B. Yuan, "Improved collaborative filtering recommendation algorithm of similarity measure," *AIP Conference Proceedings*, vol. 1839, 2017, doi: 10.1063/1.4982532.
- [21] M. Y. H. Al-Shamri and K. K. Bharadwaj, "Fuzzy-genetic approach to recommender systems based on a novel hybrid user model," *Expert Systems with Applications*, vol. 35, no. 3, pp. 1386–1399, 2008, doi: 10.1016/j.eswa.2007.08.016.
- [22] M. Jamali and M. Ester, "TrustWalker: a random walk model for combining trust-based and item-based recommendation," *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 397–406, 2009, doi: citeulike-article-id:5151320.
- [23] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, "A collaborative filtering approach to mitigate the new user cold start problem," *Knowledge-Based Systems*, vol. 26, pp. 225–238, 2012, doi: 10.1016/j.knosys.2011.07.021.
- [24] J. Bobadilla, F. Serradilla, and J. Bernal, "A new collaborative filtering metric that improves the behavior of recommender systems," *Knowledge-Based Systems*, vol. 23, no. 6, pp. 520–528, 2010, doi: 10.1016/j.knosys.2010.03.009.
- [25] J. Bobadilla, F. Ortega, A. Hernando, and J. Alcalá, "Improving collaborative filtering recommender system results and performance using genetic algorithms," *Knowledge-Based Systems*, vol. 24, no. 8, pp. 1310–1316, 2011, doi: 10.1016/j.knosys.2011.06.005.
- [26] J. Bobadilla, A. Hernando, F. Ortega, and J. Bernal, "A framework for collaborative filtering recommender systems," *Expert Systems with Applications*, vol. 38, no. 12, pp. 14609–14623, 2011, doi: 10.1016/j.eswa.2011.05.021.
- [27] J. Bobadilla, A. Hernando, F. Ortega, and A. Gutiérrez, "Collaborative filtering based on significances," *Information Sciences*, vol. 185, no. 1, pp. 1–17, 2012, doi: 10.1016/j.ins.2011.09.014.
- [28] J. Bobadilla, F. Ortega, and A. Hernando, "A collaborative filtering similarity measure based on singularities," *Information Processing and Management*, vol. 48, no. 2, pp. 204–217, 2012, doi: 10.1016/j.ipm.2011.03.007.
- [29] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, *Recommender Systems Handbook*, 2nd editio., vol. 53, no. 9, 2011.
- [30] G. Koutrika, B. Bercovitz, and H. Garcia-Molina, "FlexRecs: expressing and combining flexible recommendations," *Proceedings of the 35th SIGMOD international conference on Management of data*, pp. 745–758, 2009, doi: 10.1145/1559845.1559923.
- [31] L. Baltrunas and F. Ricci, "Experimental evaluation of context-dependent collaborative filtering using item splitting," *User Modeling and User-Adapted Interaction*, vol. 24, no. 1–2, pp. 7–34, 2014, doi: 10.1007/s11257-012-9137-9.
- [32] S. Jiang, X. Qian, J. Shen, Y. Fu, and T. Mei, "Author topic model-based collaborative filtering for personalized POI recommendations," *IEEE Transactions on Multimedia*, vol. 17, no. 6, pp. 907–918, 2015, doi: 10.1109/TMM.2015.2417506.
- [33] D. Anand and K. K. Bharadwaj, "Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5101–5109, 2011, doi: 10.1016/j.eswa.2010.09.141.
- [34] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pp. 175–186, 1994, doi: 10.1145/192844.192905.
- [35] Z. Chen, Y. Wang, S. Zhang, H. Zhong, and L. Chen, "Differentially private user-based collaborative filtering recommendation based on k-means clustering," *Expert Systems with Applications*, vol. 168, no. April 2019, 2021, doi: 10.1016/j.eswa.2020.114366.
- [36] N. Bhalse and R. Thakur, "Algorithm for movie recommendation system using collaborative filtering,"

- Materials Today: Proceedings, no. xxxx, pp. 1–6, 2021, doi: 10.1016/j.matpr.2021.01.235.
- [37] Y. Afoudi, M. Lazaar, and M. Al Achhab, “Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network,” *Simulation Modelling Practice and Theory*, vol. 113, no. July, p. 102375, 2021, doi: 10.1016/j.simpat.2021.102375.
- [38] 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 Systems*, vol. 56, pp. 156–166, 2014, doi: 10.1016/j.knsys.2013.11.006.
- [39] S. Ahmadian, M. Meghdadi, and M. Afsharchi, “A social recommendation method based on an adaptive neighbor selection mechanism,” *Information Processing and Management*, vol. 0, pp. 1–19, 2017, doi: 10.1016/j.ipm.2017.03.002.
- [40] W. Wang, J. Lu, and G. Zhang, “A new similarity measure-based collaborative filtering approach for recommender systems,” in *Foundations of Intelligent Systems*, 2014, pp. 443–452.
- [41] W. Wang, G. Zhang, and J. Lu, “Collaborative Filtering with Entropy-Driven User Similarity in Recommender Systems,” *International Journal of intelligent Systems*, vol. 30, no. 8, pp. 854–870, 2015, doi: 10.1002/int.
- [42] S. Lee, “Improving Jaccard Index Using Genetic Algorithms for Collaborative Filtering,” in *Information Science and Applications 2017: ICISA 2017*, Springer, 2017, pp. 378–385.
- [43] X. Amatriain, J. M. Pujol, and N. Oliver, “I like it... i like it not: Evaluating user ratings noise in recommender systems,” in *International Conference on User Modeling, Adaptation, and Personalization*, 2009, vol. 5535 LNCS, pp. 247–258, doi: 10.1007/978-3-642-02247-0_24.
- [44] A. Agarwal and M. Chauhan, “Similarity Measures used in Recommender Systems: A Study,” *International Journal of Engineering Technology Science and Research*, vol. 4, no. 6, pp. 2394–3386, 2017.
- [45] B. Yapriady and A. Uitdenbogerd, “Combining demographic data with collaborative filtering for automatic music recommendation,” in *Knowledge-Based Intelligent Information and Engineering Systems: 9th International Conference*, 2005, pp. 201–207, doi: 10.5772/38338.
- [46] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, “Item-based collaborative filtering recommendation algorithms,” in *10th International Conference on World Wide Web - WWW '01*, 2001, pp. 285–295, doi: 10.1145/371920.372071.
- [47] U. Shardanand and P. Maes, “Social information filtering algorithms for automating ‘word of mouth,’” *Conference proceedings on Human factors in computing systems*, pp. 210–217, 1995.
- [48] Y. Wang, J. Deng, J. Gao, and P. Zhang, “A hybrid user similarity model for collaborative filtering,” *Information Sciences*, vol. 418–419, pp. 102–118, 2017, doi: 10.1016/j.ins.2017.08.008.
- [49] G. Guo, J. Zhang, and N. Yorke-Smith, “A Novel Evidence-Based Bayesian Similarity Measure for Recommender Systems,” *ACM Transactions on the Web*, vol. 10, no. 2, pp. 1–30, 2013, doi: 10.1145/2856037.
- [50] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004, doi: 10.1145/963770.963772.
- [51] K. Miyahara and M. J. Pazzani, “Collaborative Filtering with the Simple Bayesian Classifier,” *IPSI Journal*, vol. 43, no. 11, pp. 679–689, 2002, doi: 10.1007/3-540-44533-1_68.
- [52] S. J. Yu, “The dynamic competitive recommendation algorithm in social network services,” *Information Sciences*, vol. 187, no. 1, pp. 1–14, 2012, doi: 10.1016/j.ins.2011.10.020.
- [53] C. C. Aggarwal, “Neighborhood-Based Collaborative Filtering,” in *Recommender Systems*, 2016, pp. 29–70.