A Multi-Factor Approach to Measure User Preference Similarity in Neighbor-Based Recommender Systems

Ho Thi Hoang Vy[1,2], Tiet Gia Hong[1,2], Vu Thi My Hang[1,2], Cuong Pham-Nguyen[1,2] and Le Nguyen Hoai Nam[1,2,*]
[1]*Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam*
[2]*VietNam National University, Ho Chi Minh City, Vietnam*

Keywords: Preference Similarity, Collaborative Filtering, Recommender System.

Abstract: Neighbor-based Collaborative filtering is one of the commonly applied techniques in recommender systems. It is highly appreciated for its interpretability and ease of implementation. The effectiveness of neighbor-based collaborative filtering depends on the selection of a user preference similarity measure to identify neighbor users. In this paper, we propose a user preference similarity measure named Multi-Factor Preference Similarity (MFPS). The distinctive feature of our proposed method is its efficient combination of the four key factors in determining user preference similarity: rating commodity, rating usefulness, rating details, and rating time. Our experiments have demonstrated that the combination of these factors in our proposed method has achieved good results on both experimental datasets: Movielens 100K and Personality-2018.

1 INTRODUCTION

The number of online shoppers worldwide is rapidly increasing. It is expected that the online shopping industry will continue to experience rapid growth in the near future. To increase their chances of attracting customers to their online stores, businesses should strive to understand their users' needs and improve their user experience. Recommender systems can be applied to online businesses to provide beneficial recommendations *for* both suppliers and consumers, reducing the time spent searching and selecting items (Schafer et al., 2001; Jannach et al., 2019).

Collaborative filtering is a commonly used type of recommender system that can be classified into two classes: neighbor-based and model-based. Model-based collaborative filtering collects feedback from users and uses a machine learning model to predict user preferences. Neighbor-based collaborative filtering is an easy-to-implement approach that generates interpretable recommendations (Schafer et al., 2007; Shen et al., 2013; Zhang et al., 2014; Ricci et al., 2015). It searches for users with similar preferences to an active user, also known as neighbors of the active user, and suggests items to the active user based on those neighbors. Users with greater similarity exhibit more similar preferences.

The main focus of a neighbor-based collaborative filtering recommender system is to assess the similarity between users to find the neighbor sets. One of the highly effective methods for this task is the Jaccard similarity measure (Ricci et al., 2015; Jain et al., 2020; Fkih et al., 2021). It only relies on the number of items that both related users have rated. However, such an idea is too general. This leads to low performance of neighbor-based collaborative filtering recommendation systems using Jaccard. With the above observation, in this paper, we propose an improved preference similarity measure based on Jaccard, namely Multi-Factor Preference Similarity (MFPS). The contributions of MFPS are as follows:

- To provide recommendations for an active user, it is necessary to predict his/her unknown preferences by aggregating neighbors' observed preferences. Nevertheless, due to sparse data, there are not enough observed preferences of neighbors to achieve an accurate prediction. Hence, in MFPS, a user is considered similar to another user when their observed preferences not only exhibit

* Corresponding author

Vy, H., Hong, T., Hang, V., Pham-Nguyen, C. and Nam, L.
A Multi-Factor Approach to Measure User Preference Similarity in Neighbor-Based Recommender Systems.
DOI: 10.5220/0012135500003541
In *Proceedings of the 12th International Conference on Data Science, Technology and Applications (DATA 2023)*, pages 532-539
ISBN: 978-989-758-664-4; ISSN: 2184-285X

similarity but also significantly contribute to predicting each other's unknown preferences.

- Users' preferences are expressed in detail through ratings corresponding to many different states: very dislike, dislike, neutral, like, and very like. Moreover, time plays a significant role in shaping user preferences. The more recent preferences, the greater significance. Therefore, we take into account the above rating details in MFPS.

The content of this paper will be presented in the following sections: Section 1 introduces an overview of our research; in Section 2, we review related similarity measures; Section 3 outlines our objectives in this paper; in Section 4, we propose improvements; the proposed method is implemented and experiments are conducted in Section 5; finally, conclusions and future directions are discussed in Section 6. Table 1 presents the symbols that we will use in the next sections.

Table 1: The used symbols.

Symbol	Decription
$u = \{u_1, u_2, ..., u_n\}$	The set of users
$i = \{i_1, i_2, ..., i_m\}$	The set of items
$r_{ui} \neq *$	Observed rating of user u for item i
$r_{ui} = *$	Unknown rating of user u for item i
$sim(u, v)$	Preference similarity between user u and user v
$N_{u,i}$	The neighbor set of user u for item i
I_u	The set of items rated by user u
δ	Liking threshold
t_{ui}	The time when user u perform ratings on item i
k	The size of neighbor set
α	Influence coefficient of the time difference

2 RELATED WORKS

2.1 Problem Definition

The task of recommending items is based on users' previous item preferences, which are represented by $r_{ui} \neq *$ with $u = \{u_1, u_2, ..., u_n\}$ as the set of users and $i = \{i_1, i_2, ..., i_m\}$ as the set of items. The ratings r_{ui} have values from 1 to 5 corresponding to {strongly dislike, dislike, neutral, like, strongly like}. The ratings r_{ui} = * represent the unknown ratings, meaning that the user has not experienced the item yet. For an active user, the recommendation system needs to predict his/her unknown ratings based on the known ratings (Ricci et al., 2015).

It can be seen from the example shown in Fig. 1 that user u_1 has not experienced items i_2 and i_3 (r_{u_1,i_2} = * and r_{u_1,i_3} = *). Therefore, to make recommendations for u_1, the system needs to predict r_{u_1,i_2} and r_{u_1,i_3}. The items with the highest predicted ratings for u_1 will be selected for recommendation.

user	item	rating
0	0	4
0	1	3
0	2	5
0	3	4
1	0	5
1	1	3
1	4	4
2	0	4
2	1	3
2	2	3
2	3	4
3	0	2
3	1	1
3	4	3
4	0	4
4	1	2

	i_0	i_1	i_2	i_3	i_4
u_0	4	3	5	4	-
u_1	5	3	-	-	4
u_2	4	3	3	4	-
u_3	2	1	-	-	3
u_4	4	2	-	-	-

Figure 1: The user-item rating matrix.

2.2 Neighbor-Based Recommender Systems

Neighborhood-based recommender systems operate based on the assumption that when a user u needs to purchase an item i, he/she can consult opinions from other users who have previously experienced i. Therefore, they will search for a set of users with similar preferences to u, called the neighbors of u, to analyze neighbors' opinions and help u make better decisions (Ricci et al., 2015; Fkih et al., 2021).

With such an assumption, the advantage of neighbor-based recommender systems is their high interpretability. For instance, to interpret the recommendation result of item i for user u, the system will visualize the proportion of user u 's neighbors based on rating categories (like and dislike). Using this statistical analysis, if the proportion of neighbors who like item i is high, it will be easy to persuade user u to decide to choose this item. More specifically, the process of predicting an unknown rating of a user u for an item i is implemented as follows:

- Step 1: Measure the similarity between user u and each remaining user in the system, denoted by $sim(u, v)$ where $v = 1 \dots m$
- Step 2: Identify the set of users who have rated item i. Within this set, the users with the

highest similarity to user u will be selected, denoted as the neighbor set $N_{u,i}$.

- Step 3: Calculate the rating of user u for item i by aggregating the observed ratings of the neighbors for item i, as follows:

$$r_{ui} = \mu_u + \frac{\sum_{v \in N_{u,i}} sim(u,v).(r_{v,i} - \mu_v)}{\sum_{v \in N_{u,i}} |sim(u,v)|} \quad (1)$$

where μ_u and μ_v is the average rating of the user u and v.

2.3 User Similarity Measure

As presented in section 2.2, a neighbor-based rating prediction relies on the opinions of neighbors. The accuracy of the neighbor sets depends entirely on the selection of an appropriate similarity measure (Zhang et al., 2014; Fkih et al., 2021).

Some commonly used similarity measures include Cosine (COS), Pearson Correlation (COR), Mean Squared Difference (MSD), and Jaccard (Jain et al., 2020; Fkih et al., 2021). Many studies have analyzed their drawbacks and proposed improved versions by incorporating side information. Regarding the Jaccard similarity measure, numerous variations of it have been investigated (Sun et al., 2012; Liu et al., 2014; Liang et al., 2015; Ayub et al., 2018; Bag et al., 2019). Original Jaccard only considers the number of items rated in common by two relevant users ($|I_u \cap I_v|$) as follows:

$$sim(u,v)^{JAC} = \frac{|I_u \cap I_v|}{|I_u \cup I_v|} \quad (2)$$

where $I_{u,}$ and I_v are the set of items rated by user u and user v.

The Sorensen-Dice coefficient (SDC) (Verma et al., 2020) improves the original Jaccard by adding a quantity equal to the number of common ratings to both the numerator and denominator as follows:

$$r_{ui} = \mu_u + \frac{\sum_{v \in N_{u,i}} sim(u,v).(r_{v,i} - \mu_v)}{\sum_{v \in N_{u,i}} |sim(u,v)|} \quad (3)$$

Relevant Jaccard (Bag et al., 2019) incorporates MSD into Jaccard to improve its specificity in the following manner:

$$sim(u,v)^{JMSD} = sim(u,v)^{Jaccard} \times MSD(u,v) \quad (4)$$

Similarly, Proximity-Significance-Singularity (PSS) (Liu et al., 2014) is also integrated into Jaccard as follows:

$$sim(u,v)^{JPSS} = PSS(u,v) \times sim(u,v)^{Jaccard} \quad (5)$$

(Ayub et al., 2020) proposed an improvement to Jaccard by using the ratio of the number of pairs of equal ratings to the total number of common ratings, as follows:

$$sim(u,v) = \frac{|N_T(u,v)|}{|I_u \cap I_v|} \quad (6)$$

3 MOTIVATION

The preference similarity computation step aims to identify the set of users with the most similar preferences to an active user. However, in practice, several users in this set lack the necessary rating information to accurately predict unknown ratings of the active user. Observing the user-item rating matrix in Fig. 2, we can see that u_0 and u_1 have 3 common ratings, while u_0 and u_2 have only 2 common ratings. Therefore, the Jaccard similarity between u_0 and u_1 is higher than the Jaccard similarity between u_0 and u_2. However, u_1 has not experienced i_3 and i_4 yet, so u_1 cannot support u_0 in predicting preferences for items i_3 and i_4. On the contrary, even though u_2 is less similar to u_0, u_2 has experienced i_3 and i_4, so u_0 can rely on this rating information to make decisions on i_3 and i_4.

To address this issue, it is necessary to revise the concept of preference similarity used in similarity measures in general and the Jaccard similarity measures in particular. Specifically, in this paper, we aim to incorporate the usefulness of a user into the similarity formula. The concept of usefulness of a user refers to his/her ability to provide rating information for predicting unknown ratings of the other user. In that case, the more support two users provide for each other's rating prediction, the higher their similarity. Details will be presented in section 4.2.

	i_0	i_1	i_2	i_3	i_4
u_0	5	3	1	-	-
u_1	5	3	4	-	-
u_2	5	3	-	4	3

Figure 2: The rating usefulness.

For better similarity computation, in sections 4.3 -4.4, we delve into the details of the common ratings of the related users rather than just focusing on their quantity as in the original Jaccard formulation. For example, in Fig. 3, two users u_1 and u_2 have provided up to 4 common ratings $\{i_1, i_2, i_3, i_4\}$.

However, u_1 likes items i_1, i_2, i_3 and dislikes i_4 while u_2 is the opposite. On the other hand, although u_1 and u_3 have only 2 common ratings, they both completely like them. It is clear that the similarity between u_1 and u_3 must be greater than the similarity between u_1 and u_2.

	i_1	i_2	i_3	i_4
u_1	4	4	5	2
u_2	2	2	2	5
u_3	4	4	-	-

Figure 3: The rating details.

4 OUR PROPOSED METHOD

The main objective of this section is to propose **Multi-Factor Preference Similarity** (MFPS), an improved Jaccard similarity denoted by $sim(u,v)^{MFPS}$. In the following, we will analyze the important factors defined in the MFPS formula: rating commodity, rating usefulness, rating details, and rating time.

4.1 Rating Commodity

Following the fundamental principle of the original Jaccard, the MFPS similarity between a user u and a user v ($sim(u,v)^{MFPS}$) should be proportional to the number of items that they both have rated ($c_u^v = |I_u \cap I_v|$), as follows:

$$sim(u,v)^{MFPS} \propto c_u^v = |I_u \cap I_v| \tag{7}$$

4.2 Rating Usefullness

As explained in section 3, we consider how user v contributes to the rating prediction of user u. It can be seen that if user v has rated a large number of items that user u has not yet rated ($s_u^v = |I_v - I_u|$) then user v contributes more to the rating prediction of user u. This idea can be expressed as follows:

$$sim(u,v)^{MFPS} \propto s_u^v = |I_v - I_u| \tag{8}$$

4.3 Rating Details

In addition to depending on the number of commonly rated items, the MFPS similarity between user u and user v also directly relates to the number of items that u and v both like or dislike (d_u^v). Specifically, this idea is described as follows:

$$\begin{aligned} sim(u,v)^{MFPS} &\propto d_u^v \\ &= \left| \begin{matrix} i \in (I_u \cap I_v) \\ \wedge\, r_{ui} > \delta \wedge r_{vi} > \delta \end{matrix} \right| \\ &+ \left| \begin{matrix} i \in (I_u \cap I_v) \\ \wedge\, r_{ui} < \delta \wedge r_{vi} < \delta \end{matrix} \right| \end{aligned} \tag{9}$$

where δ is the liking threshold on the rating scale.

4.4 Rating Time

User preferences may change over time. Therefore, the closer the time when user u and user v perform ratings, the more similar their preferences are. To model the similarity based on the rating time, we use the formula proposed in the study (Zhang, 2014). Specifically, it is as follows:

$$sim(u,v)^{MFPS} \propto t_u^v = \sum_{i \in I_u \cap I_v} e^{-\alpha |t_{ui} - t_{vi}|} \tag{10}$$

where α is the influence coefficient of the time difference, which falls within the range [0, 1]; t_{ui} and t_{vi} respectively represent the time when user u and user v perform ratings on item i.

4.5 Multi-Factor Preference Similarity (MFPS)

The similarity between two users in neighbor-based recommender systems is typically defined between 0 and 1. As this value approaches 1, the two users are more similar, and vice versa. To comply with this criterion, similar to the approach in (Bag et al., 2019), we will utilize the sigmoid function in MFPS as follows:

$$sim(u,v)^{MFPS} = \frac{1}{1 + 1/x} \tag{11}$$

where x is a factor proportional to $sim(u,v)^{MFPS}$, i.e., rating commodity, rating usefulness, rating details, and rating time. Therefore, the final formula of MFPS is implemented as follows:

$$sim(u,v)^{MFPS} = \frac{1}{1 + \frac{1}{c_u^v} + \frac{1}{s_u^v} + \frac{1}{d_u^v} + \frac{1}{t_u^v}} \tag{12}$$

5 EXPERIMENT

5.1 Datasets

In this study, we conducted experiments on two datasets, Movielens and Personality-2018. Detailed information on the two datasets is provided in Table 2.

Table 2: Two experimental datasets.

Datasets	Description	Rating scale
MovieLens	943 users, 1682 movies 100,000 ratings	[1,…,5]
Personality 2018	1819 users 35195 movies 1028752 ratings	[0.5,…,5]

5.2 Measurement

In this study, we use the F1-score to evaluate the recommendation performance. F1-score is a combination of two metrics: precision and recall. Precision is the ratio of accurate recommendations in the recommendation set, while recall is the ratio of accurate recommendations in the truth set. The accurate recommendation set is defined based on items with predicted ratings greater than the liking threshold δ. The truth set includes items with testing ratings greater than the liking threshold δ. Specifically, F1-score is calculated as follows:

$$F1-score = \frac{2 \times precision \times recall}{precision + recall} \quad (13)$$

5.3 Experiment Setup

In this section, we implement the user preferences similarity methods shown in Table 3. The F1-score results of the above methods will be compared with our proposed method, MFPS presented in section 4. These comparisons will be conducted on the testing ratings, which account for 20% of the total ratings in the experimental datasets.

5.4 Experiment Results and Discussion

Figures 4-9 depict the F1-score results on the experimental datasets. We observed the F1-score changes in various liking thresholds δ {3.5, 4, 4.5, and personal - the average rating of each user} and size of the neighbor set k {5, 30, and 50}. It can be seen that our proposed MFPS similarity measure achieves comparable F1-score results, and even performs better than other similarity metrics.

Table 3: Similarity measures in the experiment.

Similarity measures	Denotation
Cosine Similarity (Verma , 2020)	COS
Pearson's Correlation (Verma, 2020)	COR
Constrained Pearson's Correlation (Verma, 2020)	CPC
Jaccard Similarity (Verma, 2020)	JAC
Sorensen–Dice coefficient (Verma, 2020)	SDC
Mean Square Distance (Verma, 2020)	MSD
Jaccard Mean Square Distance (Bobadilla, 2010)	JMSD
Jaccard Proximity-Significance-Singularity (Liu, 2014)	JPSS
Relevant Jaccard (Bag, 2019)	RJ
Relevant Jaccard Mean Square Distance (Bag, 2019)	RJMDS
Jaccard Uniform Operator Distance (Sun HF, 2012)	JUOD
JacLMHUOD (Lee, 2017)	JLMHUOD
Triangle Multiplying Jaccard (Fkih, 2021)	TMJ
JACLMH (Lee, 2017)	JACLMH
Rating Jaccard - Rating Preference Behavior (Ayub, 2020)	RAJRPB
Rating Jaccard (Ayub, 2018)	RAJ
New Heuristic Similarity Model (Liu, 2014)	NHSM

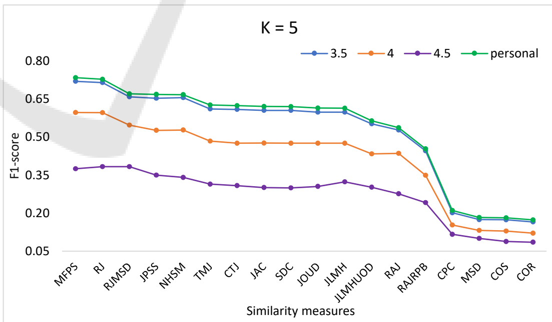

Figure 4: F1-score in the Movielens dataset at the size of the neighbor set k =5 and liking thresholds δ ={3.5, 4, 4.5, and personal - the average rating of each user}.

All methods achieved the highest F1-score results when the liking threshold δ was set to personal, i.e. the average rating of each user. This is because several users tend to rate more critically than others. Therefore, using the fixed liking threshold δ for all users would not be appropriate.

When the size of the neighbor set k increases, the F1-score results also increase because more neighbors are used in the rating prediction process. At the largest value of k, i.e. 50, and the best value of δ, i.e. personal, our method MFPS achieved the best F1-score result of 0.75949 in the Movielens dataset and 0.76912 in the personality-2018 dataset.

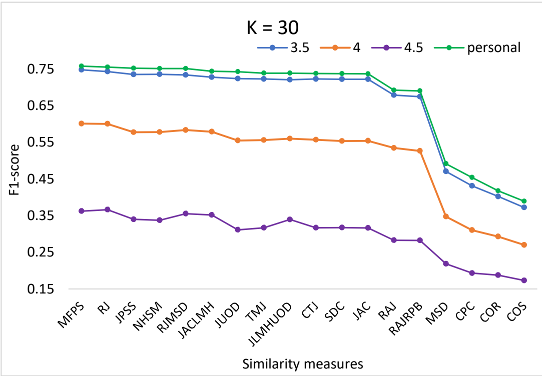

Figure 5: F1-score in the Movielens dataset at the size of the neighbor set k =30 and liking thresholds δ ={3.5, 4, 4.5, and personal - the average rating of each user}.

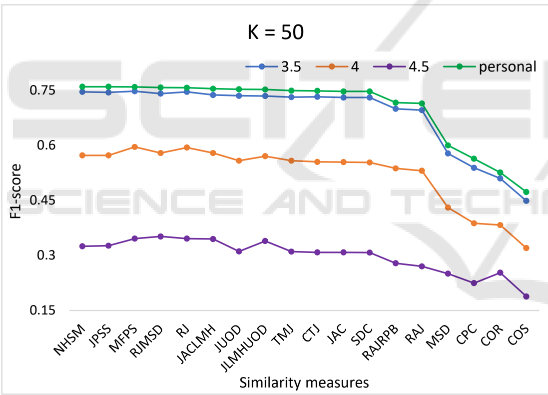

Figure 6: F1-score in the Movielens dataset at the size of the neighbor set k =50 and liking thresholds δ ={3.5, 4, 4.5, and personal - the average rating of each user}.

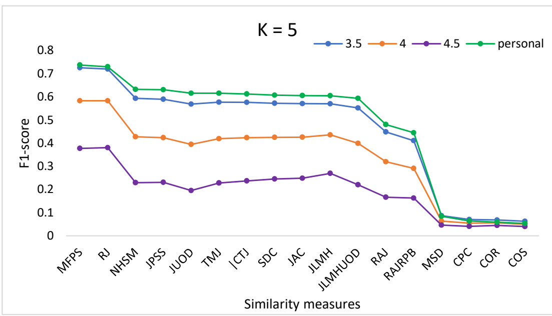

Figure 7: F1-score in the Personality-2018 dataset at the size of the neighbor set k=5 and liking thresholds δ ={3.5, 4, 4.5, and personal - the average rating of each user}.

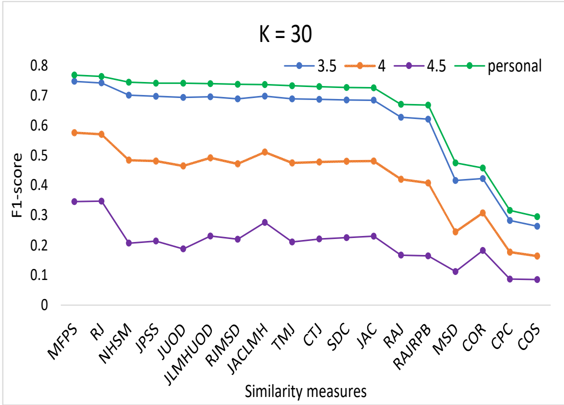

Figure 8: F1-score in the Personality-2018 dataset at the size of the neighbor set k=30 and liking thresholds δ ={3.5, 4, 4.5, and personal - average rating of each user}.

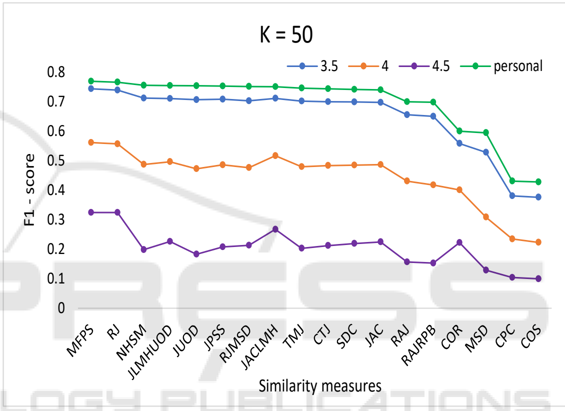

Figure 9: F1-score in the Personality-2018 dataset at the size of the neighbor set k=50 and liking thresholds δ ={3.5, 4, 4.5, and personal - average rating of each user}.

Table 4 presents the average F1-score of each similarity measure across both experimental datasets at the optimal parameters (the size of the neighbor set k is 50 and the liking threshold δ is personal). According to this table, the top 3 best methods are MFPS, RJ, and NHSM. Our proposed similarity measure MFPS achieves the highest average F1-score. It can be seen that combined methods consistently produce better results compared to traditional methods. This finding further reinforces the idea of combining multiple factors in proposing similarity measures.

Figure 10 illustrates the F1-score results of our proposed method MFPS when fixing k at 5, the liking threshold δ at personal, and decreasing gradually the influence coefficient of time difference α from 10^{-1} to 10^{-6}. As α decreases, the F1-score results of experimental methods increase. This can be explained as follows: In the movie recommendation domain, the time factor plays a significant role in determining user preferences. Therefore, reducing α implies placing more importance on the time factor in the process of computing user preference similarity.

Table 4: The average F1-score of each similarity measure across both experimental datasets at the optimal parameters (the size of the neighbor set k is 50 and the liking threshold δ is personal). Underline methods are the top 3 best methods.

Similarity measures	Movielens	Personality	Average F1-score
MFPS	0.75949	0.76912	**0.76431**
RJ	0.75741	0.76562	**0.76152**
NHSM	0.76002	0.75510	**0.75756**
JPSS	0.75975	0.75246	0.75610
RJMSD	0.75797	0.75119	0.75458
JLMHUOD	0.75259	0.75415	0.75337
JUOD	0.75278	0.75372	0.75325
JACLMH	0.75452	0.75021	0.75237
TMJ	0.74888	0.74558	0.74723
CTJ	0.74847	0.74307	0.74577
SDC	0.74701	0.74070	0.74386
JAC	0.74717	0.73974	0.74346
RAJRPB	0.71652	0.69756	0.70704
RAJ	0.71420	0.69962	0.70691
MSD	0.59985	0.59398	0.59691
COR	0.52590	0.59984	0.56287
CPC	0.56375	0.43058	0.49716
COS	0.47280	0.42764	0.45022

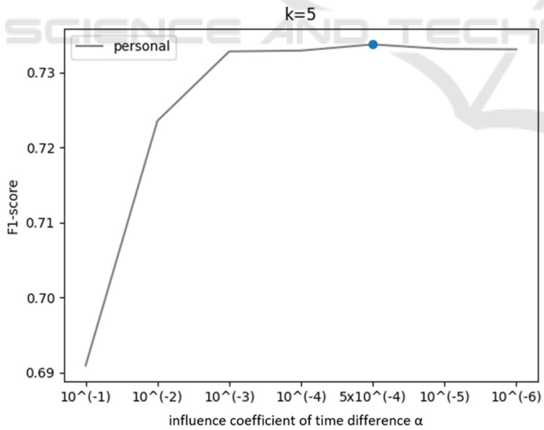

Figure 10: F1-score with the influence coefficient of time difference α from 10^{-1} to 10^{-6}.

6 CONCLUSIONS

In this paper, we have proposed a similarity measure named MFPS using the Jaccard principle The distinctive feature of MFPS is an effective combination of four key factors in determining the preference similarity between two users: rating commodity, rating usefulness, rating details, and rating time. We conducted experiments on two datasets, Movielens 100K and personality-2018. The experimental results showed that MFPS produced better results than other methods in both datasets.

In reality, user preferences are expressed through not only ratings but also reviews, user actions, and item descriptions that they are interested in. Therefore, in the future, we will aim to combine these factors into MFPS to enhance its effectiveness. However, incorporating too much information may increase the computational cost of calculating user similarity. Therefore, it is necessary to design an efficient implementation approach for the proposed similarity measure.

ACKNOWLEDGEMENTS

This research is funded by University of Science, VNUHCM under grant number CNTT 2022-06.

REFERENCES

Ayub, M., Ghazanfar, M. A., Khan, T., & Saleem, A. (2020). An effective model for Jaccard coefficient to increase the performance of collaborative filtering. *Arabian Journal for Science and Engineering*, *45*(12), 9997-10017.

Ayub, M., Ghazanfar, M. A., Maqsood, M., & Saleem, A. (2018, January). A Jaccard base similarity measure to improve performance of CF based recommender systems. In *2018 International conference on information networking (ICOIN)* (pp. 1-6). IEEE.

Bag, S., Kumar, S. K., & Tiwari, M. K. (2019). An efficient recommendation generation using relevant Jaccard similarity. *Information Sciences*, *483*, 53-64.

Bobadilla, J., Serradilla, F., & Bernal, J. (2010). A new collaborative filtering metric that improves the behavior of recommender systems. *Knowledge-Based Systems*, *23*(6), 520-528.

Fkih, F. (2022). Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison. *Journal of King Saud University-Computer and Information Sciences*, *34*(9), 7645-7669.

Jain, G., Mahara, T., & Tripathi, K. N. (2020). A survey of similarity measures for collaborative filtering-based recommender system. In *Soft Computing: Theories and Applications: Proceedings of SoCTA 2018* (pp. 343-352). Springer Singapore.

Jannach, D., & Jugovac, M. (2019). Measuring the business value of recommender systems. *ACM Transactions on Management Information Systems (TMIS)*, *10*(4), 1-23.

Lee, S. (2017). Improving jaccard index for measuring similarity in collaborative filtering. In *Information Science and Applications 2017: ICISA 2017 8* (pp. 799-806). Springer Singapore.

Liang, S., Ma, L., & Yuan, F. (2015). A singularity-based user similarity measure for recommender systems. *International journal of innovative computing information and control*, *11*(5), 1629-1638.

Liu, H., Hu, Z., Mian, A., Tian, H., & Zhu, X. (2014). A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-based systems*, 56, 156-166.

Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender systems: introduction and challenges. Recommender systems handbook, 1-34.

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. *The adaptive web: methods and strategies of web personalization*, 291-324.

Schafer, J. B., Konstan, J. A., & Riedl, J. (2001). E-commerce recommendation applications. *Data mining and knowledge discovery*, 5, 115-153.

Sharma, R., Gopalani, D., & Meena, Y. (2017, February). Collaborative filtering-based recommender system: Approaches and research challenges. *In 2017 3rd international conference on computational intelligence & communication technology* (CICT) (pp. 1-6). IEEE.

Shen, J., Wei, Y., & Yang, Y. (2013). Collaborative filtering recommendation algorithm based on two stages of similarity learning and its optimization. *IFAC Proceedings Volumes*, 46(13), 335-340.

Sun, H. F., Chen, J. L., Yu, G., Liu, C. C., Peng, Y., Chen, G., & Cheng, B. (2012). JacUOD: a new similarity measurement for collaborative filtering. *Journal of Computer Science and Technology*, 27(6), 1252.

Verma, V., & Aggarwal, R. K. (2020). A comparative analysis of similarity measures akin to the Jaccard index in collaborative recommendations: empirical and theoretical perspective. *Social Network Analysis and Mining*, 10, 1-16.

Zhang, R., Liu, Q. D., & Wei, J. X. (2014, November). Collaborative filtering for recommender systems. *In 2014 Second International Conference on Advanced Cloud and Big Data* (pp. 301-308). IEEE.

Zhang, X., He, K., Wang, J., Wang, C., Tian, G., & Liu, J. (2014, June). Web service recommendation based on watchlist via temporal and tag preference fusion. *In 2014 IEEE International Conference on Web Services* (pp. 281-288). IEEE.