IJE TRANSACTIONS B: Applications Vol. 39, No. 08, (August 2026) 2012-2021

International Journal of Engineering

Journal Homepage: www.ije.ir

A Preference-based User Similarity to Construct a Collaborative Recommender System

B. Rahimpour Cami[a], H. Hassanpour*[b], H. Mashayekhi[b]

[a] *Faculty of Industrial and Computer Engineering, Mazandaran University of Science and Technology, Babol, Iran*
[b] *Faculty of Computer Engineering, Shahrood University of Technology, Shahrood, Iran*

P A P E R   I N F O

Paper history:
Received 18 August 2025
Received in revised form 14 September 2025
Accepted 30 September 2025

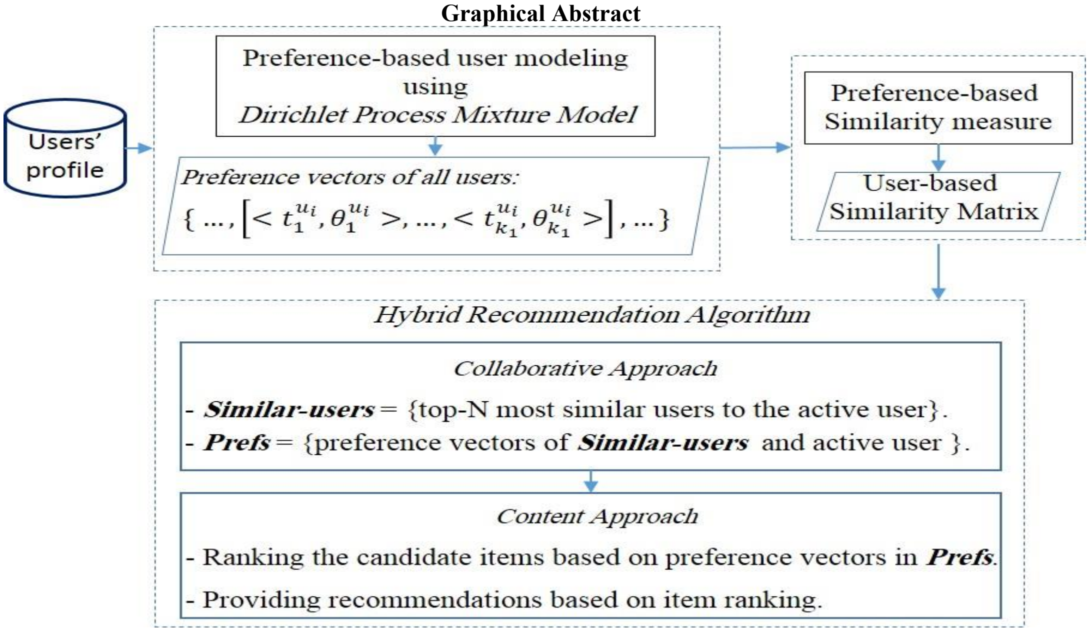
Keywords:
Collaborative Filtering
User Similarity
Similarity Measure
User Preferences

A B S T R A C T

Recommender systems have emerged as indispensable tools to overcome the information overload on the web. These systems provide personalization for users to select favorite resources. Collaborative Filtering is among the popular approaches is widely used to construct recommender systems. The main challenge of this approach is the effectiveness of its similarity measure to find similar users/items for providing recommendations. In this paper, we propose a novel measure to compute the similarity considering users' preferences. The proposed similarity measure improves the performance of recommender systems by incorporating high level latent factors instead of co-rated items by users, or common selection patterns. These latent factors are extracted via preference-based user modeling. It also provides a better recommendation when only a few ratings (sparse dataset) are available for similarity calculation. We implemented the proposed method and evaluated it using the MovieLens dataset. The evaluation results show that the proposed preference-based similarity measure considerably improves recommendation performance compared to other existing approaches.

doi*: 10.5829/ije.2026.39.08b.17*

Graphical Abstract



*Corresponding Author Email: h.hassanpour@shahroodut.ac.ir (H. Hassanpour)

Please cite this article as: Rahimpour Cami B, Hassanpour H, Mashayekhi H. A Preference-based User Similarity to Construct a Collaborative Recommender System. International Journal of Engineering, Transactions B: Applications. 2026;39(08):2012-21.

1. INTRODUCTION

Recommender systems have revolutionized online content discovery, helping users navigate the overwhelming information available on the web. By analyzing user profiles and interests, these systems predict items a user may desire, enhancing personalization and improving user experience. They are used in various online environments such as e-commerce and learning management systems (LMSs) (1, 2).

Three main approaches are available for constructing a recommender system: Collaborative Filtering (CF), content-based filtering, and hybrid methods (3). CF-based systems identify user communities with similar tastes and leverage their collective feedback for recommendations (4). Content-based systems, conversely, construct user profiles from item features and match them to new resources (5). Hybrid systems combine both approaches to improve performance (6).

Among these, CF remains particularly popular and successful (7, 8) employed by major platforms like Amazon (9) and Google News (10). Its core premise is that users with similar behavior share common interests. Modern CF implementations primarily employ neighborhood methods that compute user/item similarities (11, 12) and latent factor models that project users and items into shared feature spaces (13-15).

Despite its widespread adoption, CF's accuracy hinges on the ability of its similarity measure to discern meaningful user afinities, this is a challenge when relying solely on explicit ratings or selection patterns (16, 17). Conventional similarity measures like Cosine, Pearson, Jaccard, and mean squared difference primarily focus on users' selection patterns (16-18). Existing neighborhood-based CF methods often rely on raw co-ratings as numerical vectors for similarity calculation (18, 19), neglecting users' underlying interests and preferences. Furthermore, traditional methods struggle to perform effectively with sparse datasets, where only limited ratings are available (20, 21).

The major contribution of this work lies in using users' preferences as the main component of the similarity measure for building a collaborative filtering–based recommendation system. We incorporate users' preferences modeling (22) to construct the preference-based similarity measure. The preference-based similarity measure is the primary component of the proposed CF-based recommender systems. In this paper, we propose a new collaborative recommender system which employs both users' interests and preferences for clustering users into different groups. Users' preferences not only reflect latent factors but also prioritize them, offering a richer similarity measure for understanding users' behavior. Additionally, the proposed method uses the content information of items to construct the users' interests and to model the users' preferences. Therefore, the proposed method alleviates the sparsity and cold-start issues via using latent features more over numerical/non − numerical users' feedback. We used the MovieLens dataset to evaluate the accuracy of the proposed CF-based recommender systems which is empowered with the preference-based similarity measure. The experiments indicate that the preference-based similarity measure can improve the performance of CF-based recommender systems compared to other existing methods.

This paper is organized as follows: Section 2 provides an overview of previous research on collaborative filtering and similarity metrics. Section 3 introduces the proposed methodology. Section 4 describes the dataset, experimental setup, and comparative analysis. Finally, Section 5 provides conclusions for the paper and suggests directions for future research.

2. RELATED WORKS

In CF-based recommender systems, user profiles are represented as a rating matrix capturing user feedback (e.g., ratings or clicks) on items. Domain independency is the main advantage of collaborative filtering. As previously noted, CF systems primarily adopt two approaches: neighborhood-based methods and latent factor models.

In neighborhood-based CF methods, a similarity measure is applied to the entire rating matrix to identify neighbors of an active user or similar items to a candidate item (11, 21, 23). The most critical component of neighborhood based CF is the similarity measure's effectiveness in finding similar users. In other words, the core of neighborhood-based CF is to calculate similarities between users (user-based) (24) or items (item-based) (9). A list of popular similarity measures applicable for CF-based recommender systems along with their characteristics (advantages and disadvantages) can be found in literature (14, 21).

Bobadilla et al. (13) analyzed the rating matrix to extract significant users and items, and then applied a similarity measures (such as Pearson). Rahimpour et al. (22) presented a meta-heuristic optimization to gather a suitable subset of users' profiles, which is employed in similarity calculation to improve the performance of CF-based recommender systems. There are some modified versions of conventional similarity measures such as the constrained Pearson correlation coefficient (CPCC), weighed Pearson correlation coefficient (WPCC), and adjusted cosine measure (ACOS) which were briefly introduced by Bobadilla et al. (14) and Liu et al. (20). A combination of conventional similarity measures was discussed by Bobadilla et al. (14) and a hybrid method was presented using Jaccard and Mean Squared Differences (MSD) measures. Considering the shortcomings of conventional similarity measures, a contextual similarity measure was presented by Liu et al. (20). This similarity measure is composed of three factors: Proximity, Significance and Singularity (PSS). Similar to conventional similarity measures, proximity is used to calculate the distance between ratings. The second factor determines the significance of two ratings based on their distance from the median rating. The singularity shows how the two ratings are different compared to other ratings.

Wang et al. (25) developed a hybrid method combining user-based and item-based similarities, using item similarity as a weighting factor to enhance user similarity accuracy. Patra et al. (21) calculated the average rate of each item is via a probability distribution. Then the Bhattacharyya distance is used to compute similarity between items. Cai et al. (26) constructed a typicality matrix instead of using rating matrix. The rows and columns of the typicality matrix show the users and cluster indices of items, respectively. Each element reflects the number of items selected by a user from the corresponding cluster. Conventional similarity measures are applied to the matrix, and neighbors are gathered according to their affinity to each cluster of items.

There are several heuristic techniques to alleviate the data sparsity (rating matrix sparsity) in CF-based recommender systems. Gazdar and Hidri (27) introduced a measure that addresses limitations such as data sparsity and cold-start problems. Their approach considers both user-item interactions and contextual information, leading to more accurate and personalized recommendations. Also, Abdalla et al. (19) introduced an innovative similarity metric that incorporated additional contextual information or advanced mathematical approaches to better reflect item similarities and boost the performance of item-based collaborative filtering. A hybrid similarity model was introduced by Guan et al. (28) that combines user-based and item-based CF with additional contextual or content-based information to improve recommendation accuracy in sparse datasets. By leveraging both user-item interactions and auxiliary data, the model mitigates the cold-start problem and enhances prediction performance, even when data is limited. Patra et al. (21) introduced a method based on the Bhattacharyya coefficient, a statistical measure that quantifies the overlap between two probability distributions. This approach captures the underlying statistical relationships between users or items more effectively, even when data is sparse. Anand and Bharadwaj (29) introduced a technique to compute both the direct similarity and transitive similarity (indirect similarity) between the two users who have common ratings. They designed a method to combine the direct and transitive similarities for finding neighbors. Lee et al. (24) introduced a hybrid method in which, the satisfaction of users with each item is embedded into the CF-based recommender systems. They infer the user confidence related to each rating and use it as a coefficient for the similarity computation. Khanian Najafabadi et al. (30) incorporated the rating matrix to cluster the items and constructed a user-clusters matrix. Instead of a user-items matrix, they used users-clusters matrix to calculate user similarity. Also, association rules mining was used to find favorite items (clusters) for the user.

In the latent factor models, the CF-based methods apply machine learning techniques to the rating matrix, and extract a model for prediction (11, 21, 23). Latent factor models are a well-known and successful method to create model-based collaborative recommender systems. This approach derives hidden features from the rating matrix, which reflect implicit associations between users and items in a latent representation space. Chen et al. (31) and Koren (32) utilized matrix factorization to obtain these latent features and apply them for ranking unseen items. The Author Topic Model (ATM) is another approach based on the latent factor model, which is used in collaborative filtering (33). In ATM, topics are extracted from user generated tags for items. Then, the user similarity is calculated via the extracted topics. Although latent factors are able to capture user interests, they cannot capture complex relations from user item ratings matrices. Deep learning has emerged as a powerful method to learn effective representatives from large-scale data. Fu et al. (34), Lei et al. (35) and Zhamg et al. (36) incorporated a deep learning method to construct a collaborative-based recommender system. A deep learning method can also be used to extract items' features. Then the features are integrated into the recommendation methods such as timeSVD++ (37).

Most CF-based recommender systems rely on low-level user ratings and co-rated items for similarity computation. While some methods identify user interests, they overlook interest prioritization. Our approach leverages preference-based user modeling introduced by Rahimpour et al. (22) and constructs a CF system using a preference similarity measure..

3. PROPOSED PREFERENCE-Based USER SIMILARITY

In this section, we first examine the existence of latent features and their influence on user behavior. These features not only capture user behavior but also provide shared factors for collaborative matching. Subsequently, we present the proposed similarity measure and its mathematical formulation. Finally, we describe the recommendation process.

3. 1. Motovation The **preference-based user similarity** approach addresses three key CF challenges:

sparsity , non − numericalfeedback, and latent feature. We examine each below.

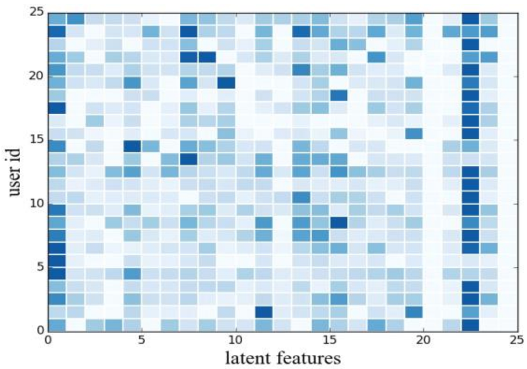
Sparsity. Sparsity. Sparsity is one of the main challenges in CF, and there are several methods to alleviate this issue (38). Conventional similarity measures cannot find similar users to those with few ratings. To illustrate, we selected a subset of the MovieLens dataset (39) depicted in Table 1. The first column shows some randomly selected users. The movies and their genres are listed in the first row. The genre of each movie, in terms of the user's point of view, was brought in the second row. According to Table 1, user5 has few ratings. Therefore, it is impossible to calculate the similarity between user user5 and others. By means of the latent features approach, we can cluster users (or items) to alleviate the sparsity issue. We clustered the users in Table 1 considering the movies' genres. *Action*, *Crime*, and *Drama* are three distinct clusters of the movies. In terms of users clustering, we can find users (such as user1 and user2) who are similar to user5 and have same *Action* cluster. However, in item-based clustering, we can find movies such as *The Matrix* from the *Action* cluster. In other words, by extracting high-level features (instead of item level ratings), we are able to provide suitable features to find neighborhoods.

Non-numerical feedback. We need a numerical rating matrix to employ a similarity measure (or latent factor model) and construct a CF-based recommender system. When dealing with non-numerical feedback (e.g., clicks or views), we must define and extract latent features from the data. To illustrate the applicability of latent features in this context, a subset of the NewsTweet dataset[1] was considered and the selected tweets of all users were aggregated. As a convenient method, we used LDA (40) as a latent factor model to extract latent features. These latent features are the topics which the tweets are discussing. We analyzed the user behavior related to each latent feature. The results of two random samples of the dataset are depicted in Figure 1.

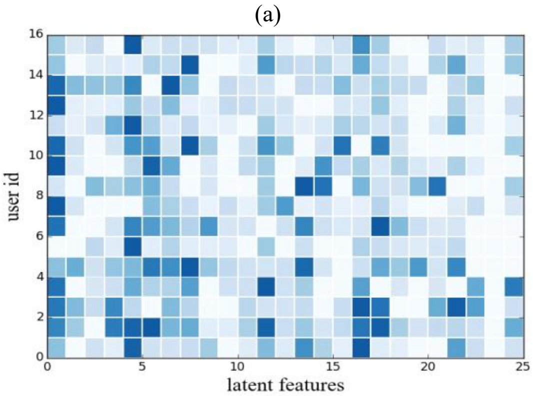
TABLE 1. A random subset of the MovieLens dataset. Movies and their genre are brought in the first row

	Movies				
	The Avengers **(Action)**	***Sherlok*** **(Crime)**	***Transformers*** ***(Action)***	***Matrix*** ***(Action)***	***Titanic*** ***(Drama)***
User1	2		2	4	5
User2	5		4		
User3			5		2
User4	1			5	
User5		2	4		
User6	4		5		1

[1] https://drive.shahroodut.ac.ir/index.php/s/rTvu6cidAiJdTaO



(a)



(b)

Figure 1. The latent features and their priority corresponding to the users for two random samples ((a) and (b)) from the NewsTweet dataset

The horizontal and vertical axes show the indices of latent features and users, respectively. Each row represents the influence of the latent features on the behavior of a particular user. Blocks with deeper shades represent features with greater influence. According to Figure 1, there are some latent factors that capture user behavior and can be considered as user interests. Some factors are more influential and affect users' preferences in selecting items.

Latent features. The main idea of the proposed method is to incorporate high-level latent features (which are extracted with latent factor modeling) instead of low-level item ratings. To further illustrate the latent features and their influence on user behavior, we applied the non-negative matrix factorization (NMF) (41) and timeSVD (32) on a subset of MovieLens dataset. In this experiment, we analyzed the users behavior via latent features. Figures 2(a) and 2(b) show the result of NMF and timedSVD, respectively. The horizontal and the vertical axes show the indices of latent features and users, respectively. The filled circles, depending on their sizes, indicate the priority of each latent feature for the users. As observed, the latent features capture users' item selection behavior across different interests, with certain

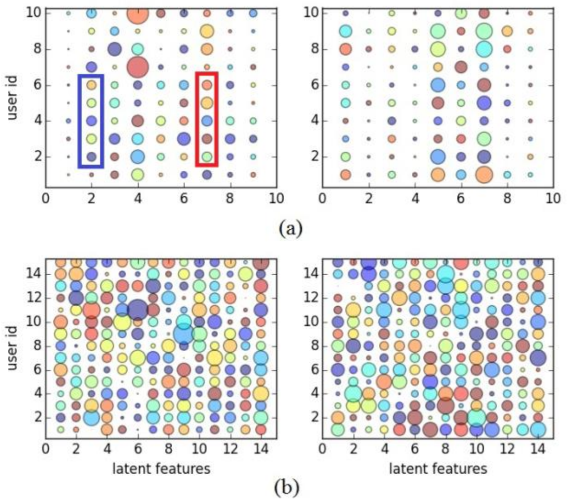


Figure 2. The extracted latent features from two random samples of the MovieLens dataset using (a) timedSVD and (b) NMF. Filled circles, depending on their size, indicate priority of the corresponding feature

features showing stronger user preference. Notably, the marked areas in the figures demonstrate shared latent features between users, enabling neighbor identification through feature similarity.

These results show that latent features effectively improve collaborative filtering. Our approach extracts them via preference-based user modeling, with the next section detailing the mathematical formulation of our preference-based similarity measure.

3. 2. Formulation of the Preference-Based Similarity Measure

We adapt latent features from user profiles (22)—originally used for content-based recommendation—to develop a novel preference-based similarity measure for collaborative filtering. Our hybrid approach integrates: 1) preference-based user modeling and 2) preference-aware similarity. Table 2 defines all notations used.

(1) Preference-based user modeling. Our preference-based user modeling approach assumes user behaviors are governed by latent factors (features), extracted using established techniques like matrix factorization and Bayesian non-parametric models (42). These latent factors indicate user interests. In the preference-based user modeling, the user preferences can be defined as the priority of interests (22). We use the Dirichlet Process Mixture Model (DPMM) as a Bayesian nonparametric framework to represent user preferences.

The graphical representation of DPMM for preference-based user modeling is depicted in Figure 3. According to this figure, the interest block and preference block are used to infer user interests and to calculate the priority of each interest, respectively. In this figure, the hyperparameter α determines the arrangement of clusters,

TABLE 2. outlines the notations for the proposed method

Notation	Description
r	An item
u	A user
A^u	Containing items selected by user (u) (user's profile)
a	Observed data
i,j,k	Used for indexing
t	The interest (cluster) identifier
θ	The preference distribution (interest priority)
$pref^u$	The preference vector over interests of user (u)
β	Dirichlet distribution parameter for the prior of preference distribution
α	Concentration parameter of Dirichlet Process (ddcrp)
G_0	Base distribution of Dirichlet Process
Ψ	A set of model parameters (α *and* β)

while the parameter G determines this distribution and is sampled from the distance dependent Chinese Restaurant Process (ddCRP) denoted as $G \sim ddCRP(\alpha)$ (43). The model parameter t acts as the cluster index (latent factor) for each data point, corresponding to the destination table of data in the ddCRP representation of the DPMM. Parameter a_i shows the observed data (items selected/rated by the user). The priority of latent clusters (θ_i) is characterized using the hyperparameter β along with the base distribution G_0. More details of the preference-based user modeling can be found in literature (22).

Through preference-based user modeling, each user u_x is represented by a preference vector of the form $(< t_1,\theta_1 >,\ldots,< t_k,\theta_k >)$, where t_i denotes the i^th interest (a cluster of similar selected items) and θ_i represents its priority (quantifying the interest's influence). The generative process of user profile modeling is represented as follows:

- For each item a_i from user profile: draw interest assignment t_i: $t_i \sim ddCRP(.)$ (interest extraction block).
- For each interest k: calculate the distribution of latent clusters (θ) using $\{G_0, \beta\}$: $\theta \sim Dir(.)$ (preference inferring).
- For a new item a_x : $t_x \sim ddCRP(.)$ and $a_x \sim CategoricalDist(\Theta,t_x)$.

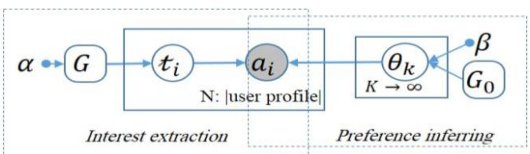


Figure 3. Graphical view of DPMM which is used for the preference-based user modeling.

(2) Preference-aware user similarity. Once the preference model is constructed, each user is represented by a preference vector, with the complete set visualized in Figure 4. Here, θ_i captures user preferences (priority weights), while t_i reflects user interests (feature vectors of selected items). For neighbor identification, we adopt a user-based CF approach that computes similarity between these vectors. Specifically, the similarity between users u_x and u_y, is derived from their preference vectors ($pref^u$) by evaluating two components: interest similarity and preference alignment. The former is calculated via *interestSim(.,.)*, which uses Jaccard similarity to quantify overlap be tween interest vectors t_i and the latter is calculated by *preferenceSim(.,.)*, a confidence term incorporating priority values θ_i of matched interests:

$$\begin{gathered} Similarity(u_x, u_y) = \\ Similarity(pref^{ux}, pref^{uy}) = \\ \sum_{i=1}^{k_{ux}} \sum_{j=1}^{k_{uy}} \left(interestSim\left(t_i^{ux}, t_j^{uy}\right) \times \right. \\ \left. preferenceSim(\theta_i^{ux}, \theta_j^{uy}) \right), \\ preferenceSim\left(\theta_i^{ux}, \theta_j^{uy}\right) = e^{-|\theta_i^{ux}, \theta_j^{uy}|} \end{gathered} \tag{1}$$

Our dual-component measure weights similarity by both shared interests and their relative importance, ensuring high-priority preferences dominate.

3. 3. Recommendation The overall process of the proposed recommender system is depicted in Figure 5. The recommendation process is performed in three steps as follows. First, user profiles are incorporated into the Dirichlet Process Mixture Model (DPMM) to construct preference-based user models. Next, the preference based similarity measure (Equation 1) generates a user-user similarity matrix. Finally, this matrix drives the recommendation algorithm, which identifies the top-N neighbors of an active user *u* and calculates *u*'s preference for candidate items (e.g., item *r*) using Equation 2.

This equation integrates:
• $A^u = \{r_1, r_2, ..., r_{|A^u|}\}$, the set of items selected by u.
• Ψ, hyper-parameters (e.g. α and β in Figure 3).
• *prefs*, concatenated preference vectors of *u*'s neighbors.
• *interestAssignment*, the probability of assigning item *r* to interest *k*.

$$pref^{u_1} = \left[< t_1^{u_1}, \theta_1^{u_1} >, \ldots, < t_{k_1}^{u_1}, \theta_{k_1}^{u_1} >\right]$$
$$\vdots$$
$$pref^{u_n} = [< t_1^{u_n}, \theta_1^{u_n} >, \ldots, < t_{k_1}^{u_n}, \theta_{k_1}^{u_n} >]$$

Figure 4. The preference vectors of all users

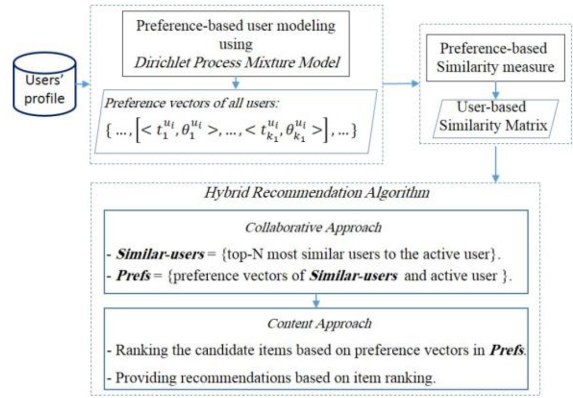


Figure 5. Components of the recommender system

• *interestPriority*, the priority of interest *k* from *prefs*.

Equation 2 ranks candidate items into u's recommendation list. Algorithm 1 implements this process, applying Equation 2 to all of u's neighbors to generate the ordered list..

$$\begin{gathered} p(r|\, A^u, prefs, \Psi) = \frac{1}{p(A^u|\Psi)} \sum_{k=1}^{|prefs|} p(t_r = \\ k|\Psi) * \mathrm{p}(\theta_k|\Psi) \times p(r, A^u|\, \theta_k, \Psi) \end{gathered} \tag{2}$$

4. EXPERIMENTAL RESULTS

We integrated our preference-based similarity measure into a hybrid CF system and validated it through extensive experiments against state-of-the-art algorithms. Below we detail the dataset, implementation, evaluation metrics, baselines, and experimental protocol, followed by a performance analysis showing our method's accuracy and robustness advantages.

We used the MovieLens[1] dataset for evaluation. To provide descriptive details about movies, we extracted storylines and genres from IMDb[2]. The movie information was then processed using text processing tools to construct feature representations corresponding to every movie.

Statistical parameters of the dataset are represented in Table 3.

We evaluate performance using standard accuracy metrics: precision, recall, and F1−measure (44), and employ the DCG@N[3] metric (4) to assess recommended resources according to their ranked positions. These metrics are calculated for varying recommendation list lengths (top@N = {3, 6, 9, 12}).

4. 1. Baselines We evaluated our approach against several existing techniques used to build recommender

[1] - https://grouplens.org/datasets/movielens
[2] - http://www.imdb.com
[3] - Discounted Cumulative Gain

Algorithm 1: the recommendation algorithm

```
Input: preference vectors, similarity matrix, active user u, and candidate_items
Output: An ordered list for recommendation
Initialization:
Similar_users = select top-N most similar users to u
Prefs = collect preference vectors of Similar_users and u
priority_list = {}
for each item in candidate_items, do: Iterates for each candidate item
p = Calculate the likelihood of item using Eq 2
priority_list = priority_list ∪ {item, p}
end for
Return priority_list in descending order of likelihood.
```

TABLE 3. Specifications of the dataset from MovieLens

Dataset	MovieLens
Number of users	4743
Number of movies	8787
Number of feedbacks	1 million

systems, utilizing the dataset presented in Table 3. These strategies involve item-based collaborative filtering (45), matrix factorization (SVD) (15), Funk-SVD (46), timeSVD (47), NMF (41), OS (27), AMI (48), and content-based filtering (5). SVD, timeSVD, Funk-SVD and NMF use latent factors to derive the user interests but do not prioritize them. OS (27) introduced a mathematically derived similarity metric. AMI (48) employed ratings to compute the statistical dependence between two users or two items.

The item-based filtering, fun-kSVD and content-based filtering were carried out using the LensKit[1] (49). The SVD and NMF were carried out using the Surprise[2] (50). Also, the timeSVD was carried out using the LibRec[3] (51-53).

4. 2. Experiments and Results

We carried out several experiments to assess the proposed recommender system (denoted as RS) equipped with our novel similarity measure.

Recommendation accuracy. We assessed our recommender system's accuracy on MovieLens using precision and DCG@N metrics across multiple trials. Compared to baselines under identical conditions (Table 4), our preference-based similarity measure achieved superior performance by effectively modeling user preferences. The results confirm that explicit preference modeling significantly improves recommendation quality over state-of-the-art approaches.

Hybrid vs. non-hybrid. The proposed method is, in fact, a hybrid approach that integrates collaborative filtering and content-based components, as illustrated in Figure 5. To illustrate the impact of the hybrid approach in providing more accurate recommendations, we constructed user models and performed several experiments on two methods, hybrid and content − based (non − hybrid) (21). First, we employed each user model separately and performed a content-based experiment. Our method recommends items matching the active user's preferences by employing a preference-based model in collaborative filtering to identify users with similar preferences. The results are depicted in Figure 6. According to this figure, the hybrid method (shown with the dotted bar) outperforms the content-based recommender system.

Sparsity. To test sparsity robustness, we randomly removed 35–40% of user profiles and compared results with baselines (see Table 5). Our method achieves higher precision and DCG@N for users with sparse histories by leveraging preference-based similarity, which relies on latent preferences rather than explicit ratings. It also addresses item cold-start by ranking new items via feature-vector-to-preference comparisons.

As reported in Tables 4 and 5, the baseline methods suffered a performance degradation of approximately 33%, whereas the proposed method showed only an 18% degradation. These results confirm that the proposed similarity measure is more resilient to sparsity and cold-start scenarios, demonstrating its effectiveness in maintaining recommendation quality compared to conventional approaches.

4. 3. Discussion

The primary focus of our work is to introduce a novel similarity measure that enhances collaborative recommender systems. It emphasizes the development of a preference-based similarity measure that improves the identification of user neighborhoods by incorporating latent factors and user preferences. As a result, the proposed method incorporating a preference-based similarity measure contributes to alleviating cold-start and sparsity as a natural by-product.

Our experiments (Tables 4 and 5) demonstrate clear improvements: for instance, our method improves precision and NDCG scores compared to the baseline methods (Section 4.1) that using conventional similarity measures. Moreover, Figure 6 shows improvement in F1 and NDCG over the non-hybrid approach (21).

[1] - http://lenskit.grouplens.org/

[2] - http://surprise.readthedocs.io/en/stable

[3] - https://www.librec.net/

TABLE 4. Comparison between the proposed RS and other methods in terms of the precision and NDCG metrics

top@N	Method	Precision	nDCG
	Lenskit: item-item	0.212	0.219
	Lenskit: Funk-SVD	0.168	0.214
	Lenskit: Content-based	0.167	0.172
3	Surprise: SVD	0.083	0.110
	Surprise: NMF	0.083	0.210
	LibRec: TimeSVD	0.342	0.360
	proposed RS	0.746	0.521
	Lenskit: item-item	0.342	0.360
	Lenskit: Funk-SVD	0.331	0.348
	Lenskit: Content-based	0.267	0.253
6	Surprise: SVD	0.217	0.233
	Surprise: NMF	0.256	0.253
	LibRec: TimeSVD	0.213	0.230
	proposed RS	0.747	0.670
	Lenskit: item-item	0.436	0.464
	Lenskit: Funk-SVD	0.427	0.450
	Lenskit: Content-based	0.331	0.346
	Surprise: SVD	0.317	0.377
9	Surprise: NMF	0.317	0.377
	LibRec: TimeSVD	0.280	0.300
	OS (27)	0.3	-
	AMI (48)*	0.25	-
	proposed RS	0.751	0.775

TABLE 5. Comparison of the precision and NDCG metrics for the proposed RS and baselines via a sparse dataset

top@N	Method	Precision	nDCG
	Lenskit: item-item	0.149	0.154
3	Lenskit: Funk-SVD	0.142	0.141
	Surprise: NMF	0.046	0.071
	proposed RS	0.615	0.398
	Lenskit: item-item	0.234	0.247
6	Lenskit: Funk-SVD	0.230	0.239
	Surprise: NMF	0.120	0.151
	proposed RS	0.607	0.503
	Lenskit: item-item	0.298	0.320
9	Lenskit: Funk-SVD	0.292	0.311
	Surprise: NMF	0.177	0.211
	proposed RS	0.616	0.586
	Lenskit: item-item	0.352	0.380
12	Lenskit: Funk-SVD	0.345	0.372
	Surprise: NMF	0.226	0.261
	proposed RS	0.617	0.652

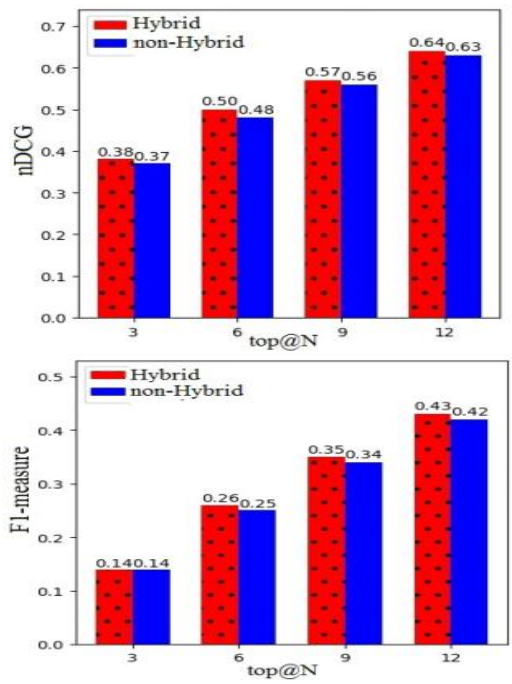


Figure 6. Comparison between the hybrid (the proposed RS) and non-hybrid (21) approaches in terms of the F1− measure and NDCG metrics

5. CONCLUSIONS

In this paper, we proposed a preference-based user similarity measure to construct a collaborative filtering recommender system. In addition to user interests, we used user preferences to calculate the similarity and find neighbors of the active user.

We implemented the preference-based user similarity and constructed a collaborative filtering system to provide recommendations.

The findings indicate that our method enhamces the performance of recommendation systems. Aaaitionally, the results indicate that, the preference-based similarity measure can alleviate the sparsity of users' profiles (rating matrix). For future research, incorporating preferences to construct item-item collaborative filtering could also be considered.

The proposed hybrid method can also be extended by incorporating contextual data (e.g., location, temporal events) for more adaptive user modeling, and by leveraging advanced deep learning models such as attention-based transformers to enrich user and item profiles.

6. REFERENCES

1. Lu J, Wu D, Mao M, Wang W, Zhang G. Recommender system application developments: a survey. Decision support systems. 2015;74:12-32. https://doi.org/10.1016/j.dss.2015.03.008

2. Zhao WX, Li S, He Y, Chang EY, Wen J-R, Li X. Connecting social media to e-commerce: Cold-start product recommendation using microblogging information. IEEE Transactions on Knowledge and Data Engineering. 2015;28(5):1147-59. https://doi.org/10.1109/TKDE.2015.2508816
3. Aljunid MF, Manjaiah D, Hooshmand MK, Ali WA, Shetty AM, Alzoubah SQ. A collaborative filtering recommender systems: Survey. Neurocomputing. 2025;617:128718. https://doi.org/10.1016/j.neucom.2024.128718
4. Wu H, Pei Y, Li B, Kang Z, Liu X, Li H. Item recommendation in collaborative tagging systems via heuristic data fusion. Knowledge-Based Systems. 2015;75:124-40. https://doi.org/10.1016/j.knosys.2014.11.026
5. Lops P, De Gemmis M, Semeraro G. Content-based recommender systems: State of the art and trends. Recommender systems handbook. 2010:73-105. https://doi.org/10.1007/978-0-387-85820-3_3
6. Burke R. Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction. 2002;12(4):331-70. https://doi.org/10.1023/A:1021240730564
7. Cunha T, Soares C, de Carvalho AC. Metalearning and Recommender Systems: A literature review and empirical study on the algorithm selection problem for Collaborative Filtering. Information Sciences. 2018;423:128-44. https://doi.org/10.1016/j.ins.2017.09.050
8. Elahi M, Ricci F, Rubens N. A survey of active learning in collaborative filtering recommender systems. Computer Science Review. 2016;20:29-50. https://doi.org/10.1016/j.cosrev.2016.05.002
9. Linden G, Smith B, York J. Amazon. com recommendations: Item-to-item collaborative filtering. IEEE Internet computing. 2003;7(1):76-80. https://doi.org/10.1109/MIC.2003.1167344
10. Ricci F, Rokach L, Shapira B. Introduction to recommender systems handbook. Recommender systems handbook: Springer; 2010. p. 1-35.
11. Adomavicius G, Kwon Y. Improving aggregate recommendation diversity using ranking-based techniques. IEEE Transactions on Knowledge and Data Engineering. 2011;24(5):896-911. https://doi.org/10.1109/TKDE.2011.15
12. Adomavicius G, Tuzhilin A. 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. 2005;17(6):734-49. https://doi.org/10.1109/TKDE.2005.99
13. Bobadilla J, Hernando A, Ortega F, Gutiérrez A. Collaborative filtering based on significances. Information Sciences. 2012;185(1):1-17. https://doi.org/10.1016/j.ins.2011.09.014
14. Bobadilla J, Serradilla F, Bernal J. A new collaborative filtering metric that improves the behavior of recommender systems. Knowledge-Based Systems. 2010;23(6):520-8. https://doi.org/10.1016/j.knosys.2010.03.009
15. Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. Computer. 2009;42(8):30-7. https://doi.org/10.1109/MC.2009.263
16. Bag S, Kumar SK, Tiwari MK. An efficient recommendation generation using relevant Jaccard similarity. Information Sciences. 2019;483:53-64. https://doi.org/10.1016/j.ins.2019.01.023
17. He P, Shi J, Ma W, Zheng X. Broad collaborative filtering with adjusted cosine similarity by fusing matrix completion. Applied Soft Computing. 2024;165:112075. https://doi.org/10.1016/j.asoc.2024.112075
18. Abdalla HI, Amer YA, Nguyen L, Amer AA, Al-Maqaleh BM, editors. Numerical similarity measures versus Jaccard for collaborative filtering. International Conference on Advanced Intelligent Systems and Informatics; 2023: Springer.
19. Abdalla HI, Amer AA, Amer YA, Nguyen L, Al-Maqaleh B. Boosting the item-based collaborative filtering model with novel similarity measures. International Journal of Computational Intelligence Systems. 2023;16(1):123. https://doi.org/10.1007/s44196-023-00299-2
20. Liu H, Hu Z, Mian A, Tian H, Zhu X. A new user similarity model to improve the accuracy of collaborative filtering. Knowledge-based systems. 2014;56:156-66. https://doi.org/10.1016/j.knosys.2013.11.006
21. Patra BK, Launonen R, Ollikainen V, Nandi S. A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data. Knowledge-Based Systems. 2015;82:163-77. https://doi.org/10.1016/j.knosys.2015.03.001
22. Cami BR, Hassanpour H, Mashayekhi H. User preferences modeling using dirichlet process mixture model for a content-based recommender system. Knowledge-Based Systems. 2019;163:644-55. https://doi.org/10.1016/j.knosys.2018.09.028
23. Karabadji NEI, Beldjoudi S, Seridi H, Aridhi S, Dhifli W. Improving memory-based user collaborative filtering with evolutionary multi-objective optimization. Expert Systems with Applications. 2018;98:153-65. https://doi.org/10.1016/j.eswa.2018.01.015
24. Lee J, Lee D, Lee Y-C, Hwang W-S, Kim S-W. Improving the accuracy of top-N recommendation using a preference model. Information Sciences. 2016;348:290-304. https://doi.org/10.1016/j.ins.2016.02.005
25. Wang Y, Deng J, Gao J, Zhang P. A hybrid user similarity model for collaborative filtering. Information Sciences. 2017;418:102-18. https://doi.org/10.1016/j.ins.2017.08.008
26. Cai Y, Leung H-f, Li Q, Min H, Tang J, Li J. Typicality-based collaborative filtering recommendation. IEEE Transactions on Knowledge and Data Engineering. 2013;26(3):766-79. https://doi.org/10.1109/TKDE.2013.7
27. Gazdar A, Hidri L. A new similarity measure for collaborative filtering based recommender systems. Knowledge-Based Systems. 2020;188:105058. https://doi.org/10.1016/j.knosys.2019.105058
28. Guan J, Chen B, Yu S. A hybrid similarity model for mitigating the cold-start problem of collaborative filtering in sparse data. Expert Systems with Applications. 2024;249:123700. https://doi.org/10.1016/j.eswa.2024.123700
29. Anand D, Bharadwaj KK. Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities. Expert systems with applications. 2011;38(5):5101-9. https://doi.org/10.1016/j.eswa.2010.09.141
30. Najafabadi MK, Mahrin MNr, Chuprat S, Sarkan HM. Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data. Computers in Human Behavior. 2017;67:113-28. https://doi.org/10.1016/j.chb.2016.11.010
31. Chen B, Yang Z, Yang Z. An algorithm for low-rank matrix factorization and its applications. Neurocomputing. 2018;275:1012-20. https://doi.org/10.1016/j.neucom.2017.09.052
32. Koren Y, editor Collaborative filtering with temporal dynamics. Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining; 2009.
33. Jiang S, Qian X, Shen J, Fu Y, Mei T. Author topic model-based collaborative filtering for personalized POI recommendations. IEEE transactions on multimedia. 2015;17(6):907-18. https://doi.org/10.1109/TMM.2015.2417506
34. Fu M, Qu H, Yi Z, Lu L, Liu Y. A novel deep learning-based collaborative filtering model for recommendation system. IEEE transactions on cybernetics. 2018;49(3):1084-96. https://doi.org/10.1109/TCYB.2018.2795041

35. Dong X, Yu L, Wu Z, Sun Y, Yuan L, Zhang F, editors. A hybrid collaborative filtering model with deep structure for recommender systems. Proceedings of the AAAI Conference on artificial intelligence; 2017.
36. Zhang S, Yao L, Sun A, Tay Y. Deep learning based recommender system: A survey and new perspectives. ACM computing surveys (CSUR). 2019;52(1):1-38. https://doi.org/10.1145/3285029
37. Wei J, He J, Chen K, Zhou Y, Tang Z. Collaborative filtering and deep learning based recommendation system for cold start items. Expert systems with applications. 2017;69:29-39. https://doi.org/10.1016/j.eswa.2016.09.040
38. Hwangbo H, Kim Y. An empirical study on the effect of data sparsity and data overlap on cross domain collaborative filtering performance. Expert Systems with Applications. 2017;89:254-65. https://doi.org/10.1016/j.eswa.2017.07.041
39. Harper FM, Konstan JA. The movielens datasets: History and context. Acm transactions on interactive intelligent systems (tiis). 2015;5(4):1-19. https://doi.org/10.1145/2827872
40. Jelodar H, Wang Y, Yuan C, Feng X, Jiang X, Li Y, et al. Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey. Multimedia tools and applications. 2019;78(11):15169-211. . https://doi.org/10.1007/s11042-018-6894-4
41. Luo X, Zhou M, Xia Y, Zhu Q. An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. IEEE Transactions on Industrial informatics. 2014;10(2):1273-84. https://doi.org/10.1109/TII.2014.2308433
42. Gershman SJ, Blei DM. A tutorial on Bayesian nonparametric models. Journal of Mathematical Psychology. 2012;56(1):1-12. https://doi.org/10.1016/j.jmp.2011.08.004
43. Blei DM, Frazier PI. Distance dependent Chinese restaurant processes. Journal of Machine Learning Research. 2011;12(8). https://dl.acm.org/10.1007/978-3-031-40498-6_23
44. Avazpour I, Pitakrat T, Grunske L, Grundy J. Dimensions and metrics for evaluating recommendation systems. Recommendation systems in software engineering: Springer; 2013. p. 245-73.
45. Deshpande M, Karypis G. Item-based top-n recommendation algorithms. ACM Transactions on Information Systems (TOIS). 2004;22(1):143-77. https://doi.org/10.1145/963770.963776
46. Çoba L, Zanker M, editors. Replication and reproduction in recommender systems research-evidence from a case-study with the rrecsys library. International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems; 2017: Springer.
47. Koren Y, Rendle S, Bell R. Advances in collaborative filtering. Recommender systems handbook. 2021:91-142. https://doi.org/10.1007/978-1-0716-2197-4_3
48. Fkih F. Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison. Journal of King Saud University-Computer and Information Sciences. 2022;34(9):7645-69. https://doi.org/10.1016/j.jksuci.2021.09.014
49. Ekstrand MD, Ludwig M, Konstan JA, Riedl JT, editors. Rethinking the recommender research ecosystem: reproducibility, openness, and lenskit. Proceedings of the fifth ACM conference on Recommender systems; 2011.
50. Hug N. Surprise: A Python library for recommender systems. Journal of Open Source Software. 2020;5(52):2174. https://doi.org/10.21105/joss.02174
51. Guo G, Zhang J, Sun Z, Yorke-Smith N, editors. Librec: A java library for recommender systems. CEUR Workshop Proceedings; 2015: RWTH Aachen University.
52. Spoorthy G, Sanjeevi S. Multi-criteria–recommendations using autoencoder and deep neural networks with weight optimization using firefly algorithm. International Journal of Engineering Transactions A: Basics. 2023;36(1):130-8. https://doi.org/10.5829/ije.2023.36.01a.15
53. Moeini M, Broumandnia A, Moradi M. A Personalized Web Recommendation System Based on a Weighted user Behavior Profile by Applying Extended Learning Method. 2025. https://doi.org/10.5829/ije.2025.38.04a.17

Persian Abstract

چکیده

سیستم‌های توصیه‌گر به عنوان ابزارهای ضروری برای غلبه بر سربار اطلاعات در بستر وب توسعه یافته‌اند. این سیستم‌ها با فراهم کردن یک محیط شخصی‌سازی به کاربران امکان می‌دهند تا منابع مورد علاقه خود را انتخاب کنند. فیلترینگ مشارکتی از جمله رویکردهای موفق است که به طور گسترده برای ساخت سیستم‌های توصیه‌گر استفاده می‌شود. چالش اصلی این رویکرد، اثربخشی معیار شباهت آن برای یافتن کاربر/کالای مشابه برای ارائه توصیه است. در این مقاله، یک معیار جدید برای محاسبه شباهت با در نظر گرفتن تمایلات کاربران پیشنهاد می‌شود. معیار شباهت پیشنهادی، به جای استفاده از ماتریس امتیاز *کاربر×کالا*، با ترکیب فاکتورهای پنهان سطح بالا، عملکرد سیستم‌های توصیه‌گر مبتنی بر فیلترینگ مشارکتی را بهبود می‌بخشد. این فاکتورهای پنهان از طریق مدل‌سازی مبتنی بر تمایلات کاربر، استخراج می‌شوند. همچنین، زمانی که ماتریس بازخورد خلوت باشد (مجموعه داده‌های پراکنده) توصیه بهتری ارائه می‌شود. روش پیشنهادی با استفاده از مجموعه داده‌های MovieLens ارزیابی شده‌است. نتایج ارزیابی نشان می‌دهد که معیار شباهت پیشنهادی مبتنی بر تمایلات کاربر، عملکرد توصیه را در مقایسه با سایر رویکردهای موجود، بهبود می‌بخشد.