Switching Hybrid Method Based on User Similarity and Global Statistics for Collaborative Filtering

PATRICIA ORTAL, (Member, IEEE), AND MASATO EDAHIRO, (Member, IEEE)
Graduate School of Information Science, Nagoya University, Aichi 464-8601, Japan
Corresponding author: Patricia Ortal (patricia.ortal@ieee.org)

ABSTRACT Collaborative filtering (CF) is a technique used in recommender systems to provide meaningful suggestions based on known feedback obtained from like-minded users. The measure of similarity plays a critical role in the performance of neighborhood-based CF methods. However, conventional similarity measures suffer from limitations because they only consider the direction of the rating vectors. We propose a novel similarity measure that considers the semantic nuances of the ratings; in particular, it weights the contributions of ratings in proportion to the users’ degree of indifference towards the items. Additionally, to address the sparsity problem that affects the performance of CF techniques, we propose a switching hybrid method that predicts user ratings based on either our custom similarity measure or through user and item biases. We evaluated the proposed method on six different datasets and compared it with other CF methods. The results show that the proposed recommender consistently outperforms those using conventional similarity measures when the sparsity of the dataset is high.

INDEX TERMS Collaborative filtering, hybrid methods, neighborhood-based methods, similarity measure, sparsity, recommender systems.

I. INTRODUCTION
Decision making involves the evaluation and selection of an option between alternatives based on specific criteria. Classical decision theory is associated with the identification of optimal decisions, considering an ideal decision maker who is internally consistent and completely rational. However, such a logical system fails to explain real-world scenarios [1]. Under conditions where available information is incomplete or overly complex, decision makers rely on various simplifying heuristics or efficient rules of thumb rather than extensive analytical processing [2]. However, in highly difficult situations, they may be unable to make a decision; this is known as analysis paralysis. One of the main causes of analysis paralysis is choice overload. Recent research argues that situations in which several options are presented can negatively affect consumer expectations. This leads to choice deferral and lower satisfaction with the selected option [3].

E-commerce websites rapidly grew with the digital revolution. Online stores now provide users with a more comprehensive search than their brick-and-mortar counterparts, delivering information for an immense number of products and services instantaneously. However, this high number of choices has become one of the biggest problems that e-commerce users face [4]. Rather than being a benefit, having several options frequently overwhelms users, leading them to make poor decisions [5]. This choice paradox is an important catalyst for the development of recommender systems technology.

Recommender Systems (RS) are personalized information agents that provide suggestions for products or services likely to interest a particular user [5], [6]. RS assists users in filtering the available alternatives and narrowing down their choices, leading to a reduction in information overload and preventing analysis paralysis. The essence of RS is the assumption that customer interests can be inferred from different sources of data and that significant dependencies exist between user and item interactions. Because user attitudes toward products have been demonstrated to exhibit a certain degree of consistency [7], they are useful indicators of future choices. The purpose of RS is not only to provide users with more relevant choices but also to help the user discover new, particularly
interesting, unexpected, and diverse content. Moreover, this enhances the overall shopping experience and contributes significantly to customer satisfaction [8] and loyalty [9].

A conventional method for classifying recommendation techniques [5], [6] is shown in Table 1. Collaborative filtering (CF) methods are of particular interest because they are considered to be one of the most popular and widely implemented recommendation techniques, often exhibiting high predictive accuracy [10]–[13].

CF makes recommendations from explicit feedback in the form of ratings and reviews, or implicit preferences from historical behavior; such as past purchases or browsing history. The key idea is that the rating of a target user for an item is likely to be similar to that of another user, if both users have rated other items in a similar manner [5]. Thus, CF has been regarded as one of the electronic surrogates of the traditional word-of-mouth [14], where like-minded users influence each other on the consumption of related products or services.

In the CF neighborhood-based approach, the preference ratings of a target user are estimated as a weighted average of the known ratings by the target user’s closest neighbors. The weights are determined by computing a similarity measure between users (or items in the case of item-based models). CF neighborhood-based methods are often preferred in practice because they are intuitive, relatively simple to implement, stable, and highly explainable [15]–[17]. However, the performance of these methods is known to be highly sensitive to the number of observed ratings. When the rating sparsity is high, it becomes difficult to identify suitable matching neighbors, leading to less accurate recommendations. This is closely related to the cold-start problem, where it is not possible to provide reliable recommendations to new users or about new items because of the initial lack of ratings [18], [19].

Another critical aspect that has a significant impact on both the accuracy and performance of neighborhood-based recommenders is the choice of the similarity metric. The similarity metric identifies reliable neighbors and weights their relevance in the prediction. Its definition must capture the aspects of the relationships between users and items that are most representative of user preferences.

The present research proposes a novel CF approach for predicting user rating values. It is designed to avoid common problems of conventional similarity-based methods and mitigate the adverse effects of highly sparse data. In particular, the contributions of our study are as follows:

- A new custom similarity measure that considers the semantic meaning of the ratings, with a score bounded between 0 and 1 for higher interpretability.
- The use of the global average, item average, and user average to compute user and item biases that in turn are used to compute ratings.
- The proposal of a switching hybrid method that chooses between two predicted ratings based on a sparsity criterion.

The main difference between our proposed method and existing methods lies in the novel similarity measure introduced in Section III-A. In particular, this was designed to consider the user’s sentiment toward the items, interpreted in terms of how extreme their ratings are. In addition, we proposed an RS architecture based on this similarity. The recommender, described in Section III-C, uses a switching hybrid method to handle sparsity and complements its predictions with those of a recommender using Jaccard similarity. This novel architecture is shown to outperform conventional ones, as described in Section V.

Section II offers an overview of neighborhood-based CF methods, with an emphasis on conventional similarity measures. We also outline the main problems associated with these similarity measures as well as the difficulty in making meaningful predictions for highly sparse datasets. Section III proposes the Switching Hybrid with Biases (SHB) recommender comprising a CF method with a custom similarity measure and a ratings prediction approach based on global statistics. This is used in combination with a CF method using Jaccard similarity. The custom similarity measure of the SHB recommender was designed to consider the semantic information in the ratings to generate more meaningful similarity scores, leading to better performance even if the dataset sparsity is high. Section IV outlines the experimental settings used in the evaluation of our proposal, including the details of the six datasets used. Section V analyzes the performance of the proposed recommender in terms of root-mean-square and mean absolute errors and compares it to those of conventional similarity-based CF methods. Section VI summarizes our findings and conclusions.

II. PREVIOUS WORK

One of the earliest applications of collaborative filters emerged owing to the need to aid overwhelmed email users in sorting through their huge stream of incoming content. The Tapestry system [20] allowed users to collaboratively annotate documents with tags that are later used to build filters. The filters ensured that users received only potentially interesting information, allowing them to manage emails more efficiently. Another early example of a platform that implemented CF techniques is the GroupLens system [21], which was designed to recommend news articles to readers based on the ratings of other users.

A. COLLABORATIVE FILTERING METHODS

CF techniques are generally classified into two categories: neighborhood-based and model-based techniques.
Neighborhood-based methods (also referred to as memory-based methods) compute recommendations for a target user from the known scores of other users with similar rating patterns. These methods are further classified as 'user-based' when they employ users’ ratings to cluster like-minded users together and 'item-based' if they attempt to find the items that are most similar to a target one. Model-based approaches, conversely, use the ratings to build predictive models. Some examples include Bayesian clustering, Boltzmann machines [22], and latent factor models such as singular value decomposition [23] or matrix factorization [24], [25].

Matrix factorization (MF) algorithms aim to represent the user-item matrix as a product of matrices, with user vectors reduced to a lower-dimensional latent space. Once the user-item matrix has been factorized in this manner, the lower-dimensional matrices can be used to directly estimate the ratings. This approach was first proposed by Funk [24] and is usually referred to as FunkMF. Other methods building on these ideas, such as SVD++ [25], have also exhibited good recommendation performance. Generally, MF methods tend to be more accurate than neighborhood models; however, they lack the explainability and versatility that make neighborhood-based methods prevalent in commercial applications [5], [17]. Furthermore, each new observation, such as the addition of a new user, affects the choice of latent space and requires the user-item matrix to be factorized again; this is a costly operation, and thus, predictions cannot be updated in real-time as new ratings are added to the dataset.

B. SIMILARITY MEASURES

Similarity calculation plays a fundamental role in the neighborhood-based approach to CF. In this context, a basic rating prediction for item $i$ by user $u$ is derived directly from an $n \times m$ user-item rating matrix $R$, where $n$ denotes the number of users and $m$ the number of items. In particular, it is common to use the weighted average of the ratings of a set of neighboring users, as follows:

$$\hat{r}_{u,i} = \frac{\sum_{v \in N(u,i)} \text{sim}(u, v) \cdot r_{v,i}}{\sum_{v \in N(u,i)} \text{sim}(u, v)}. \quad (1)$$

Here, $\text{sim}(u, v)$ represents the similarity between users $u$ and $v$, and $N(u,i)$ denotes the set of $k$-nearest neighbors (k-NN) for user $u$ (i.e., users with the highest similarity with respect to $u$) who have rated item $i$. Different users have different tendencies when rating items. Some may tend to give higher ratings than others, which would negatively affect the performance of Eq. 1. To account for these differences, a mean-centered version of this approach is adopted, which is defined as follows:

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{v \in N(u,i)} \text{sim}(u, v) \cdot (r_{v,i} - \bar{r}_v)}{\sum_{v \in N(u,i)} \text{sim}(u, v)}. \quad (2)$$

where $\bar{r}_u$ and $\bar{r}_v$ denote the average ratings of users $u$ and $v$, respectively.

As seen in Eq. 1, the similarity score is used to weight the contribution of other user ratings and to determine which users will influence the predicted result. The similarity is calculated between user vectors, where the elements of the vectors are the ratings given by the user to different items. A sparse user-item matrix $R$, where the elements correspond to the ratings if available, or zero if not, has the following structure:

$$u_1 \begin{bmatrix} 0 & 0 & \cdots & 3 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 0 \end{bmatrix}$$

Numerous similarity measures have been proposed to better capture the relationship patterns between users and items. The Pearson correlation coefficient (PCC), cosine similarity (COS), and the Jaccard index (JAC) are the most widely adopted similarity measures in the literature [12], [17], [26]–[28].

Given two user vectors $u$ and $v$, PCC($u, v$) is defined as follows:

$$\text{PCC}(u, v) = \frac{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_v} (r_{v,i} - \bar{r}_v)^2}}. \quad (3)$$

where $I_u$ and $I_v$ denote the sets of items rated by users $u$ and $v$, respectively. PCC measures the strength of the linear relationship between two vectors, regardless of the magnitude of their elements, implying that even when users have different average ratings, PCC will consider them to be similar as long as they have similar trends. A value of $+1$ indicates a strong positive correlation, whereas a value of $-1$ indicates a strong negative correlation. However, users with negative correlations are sometimes filtered out as a heuristic enhancement [17].

COS calculates the similarity of two vectors by measuring the cosine of the angle between them. Given two vectors $u$ and $v$, the cosine similarity is calculated as follows:

$$\text{COS}(u, v) = \frac{\sum_{i \in I_u \cap I_v} r_{u,i}r_{v,i}}{\sqrt{\sum_{i \in I_u} r_{u,i}^2} \sqrt{\sum_{i \in I_v} r_{v,i}^2}}. \quad (4)$$

A COS value of 0 implies that the vectors are orthogonal and thus completely dissimilar, whereas a value of 1 corresponds to vectors pointing in the same direction and therefore that are maximally similar. Unlike PCC, COS does not explicitly contain the means and variances for each user’s ratings in its definition.

JAC is naturally bound between 0 and 1. It assigns a value of 0 when there are no items rated simultaneously by both users, and a value of 1 when both users have rated the same items. The major weakness of JAC is that it does not consider the actual values of the ratings; it only considers the items that
were actually rated. Even when two users have diametrically opposed preferences, JAC will assign a score of 1 if they rated the same items.

There has been extensive research comparing the advantages and disadvantages of these measures, such as in [13] and [29]. The study of [27] summarized the drawbacks of these similarities as follows:

- Flat-value problem: If all the ratings in a vector have the same value, PCC cannot be computed because the denominator in Eq. 3 becomes 0. Similarly, if both vectors have constant ratings, COS is always 1 even if the constant value differs between users.
- Opposite-value problem: When two user vectors have completely opposite values, PCC will always be −1, even if the user preferences are not extremely opposite in terms of rating semantics.
- Single-value problem: If two users rated only one item in common, PCC cannot be calculated, whereas COS and JAC always yield a value of 1 irrespective of the actual rating values.
- Cross-value problem: If two users rated only two items in common, PCC will either yield a value of −1 if the values cross each other or a value of 1 otherwise.

To overcome these deficiencies of conventional similarity measures, researchers have proposed different approaches. Candillier et al. [12] demonstrated that weighting similarity measures such as COS and PCC with JAC significantly improves the methods’ performance. In this context, JAC ensures that the rating pairs share sufficient attributes for the similarity to be reliable. Ahn [30] developed a heuristic similarity measure based on three aspects: proximity, impact, and popularity (PIP), representing domain-specific interpretations of user ratings. PIP weights the semantic agreements and disagreements between users and exhibits a superior performance in cold-start conditions. However, its formula is expensive to compute, and because its score is not normalized, it can assume values greater than 1 making it less intuitive and difficult to interpret.

Liu et al. [31] proposed another similarity metric termed as NHSM to address the shortcomings of PIP. This proposal uses the definitions of proximity, significance, and singularity, combined with JAC, along with the mean and variance for the ratings, to achieve a remarkable improvement over PIP. Said et al. [32] proposed and investigated the effects of two weighting schemes that consider the degree of popularity of items. The results indicate that the weighting approaches have a negligible effect on COS but have a significant influence on PCC when the users have more than a few ratings in common.

Bobadilla et al. [19] presented a similarity measure that combines conventional similarity metrics; however, it optimizes the weights of each similarity through neural networks. The results show an improvement in the prediction quality in cold-start situations. This metric is superior to PIP in terms of performance and computation time. Laveti et al. [33] proposed a weighted hybrid ensemble similarity metric combining two or more conventional approaches that demonstrates an improvement in recommendation accuracy; however, it relies on a high number of rating samples and neighbors to achieve high performance. Guo et al. [27] defined a Bayesian similarity measure based on the Dirichlet distribution that considers the direction and length of the rating vectors while also considering the rating semantics of all rating pairs. Although the approach can perform well and can attain good generality, it includes a number of hyperparameters that require tuning to achieve these results.

**C. SPARSITY IN NEIGHBORHOOD-BASED MODELS**

In practice, users tend to rate a small subset of the item catalogue, consequently making user-item matrices sparse. Sparsity is a major problem in RS because when most ratings are unspecified, finding reliable neighbors is difficult, and the number of co-rated items between users is small. In scenarios with high levels of sparsity, recommendations are often biased and inaccurate. One of the earliest approaches for tackling sparsity in rating matrices is “default voting,” proposed by Breese et al. [34]. Default voting involves using predetermined values to replace missing data and increase the number of mutually rated items between users. Default votes are only applied in cases in which two users are compared and at least one of the users rated the item. However, replacing missing data usually introduces a significant amount of bias that affects the performance of the estimations.

Wang et al. [35] developed an algorithm that combines ratings from both similar users and items to reduce the dependency on missing data. The experiments demonstrate robustness against data sparsity and improved prediction accuracy when compared with pure user-based or item-based approaches. Zhang and Pu [36] proposed a recursive algorithm that can make coarse predictions for the missing rating values of neighboring users, thus alleviating data sparseness. The proposed approach shows promising results, achieving higher prediction accuracy than the conventional approach using PCC. However, the performance of this algorithm also depends on several parameters tuned to every specific dataset, and it has a high computational cost.

Luo et al. [16] addressed the data sparsity problem by describing relationships between users through local and global user similarities. Local similarities are represented as edges of a user graph and are determined based on surprisal-based vector similarity. Global similarities are calculated as the maximin distance of any two nodes in the graph. The results show that, under sparse dataset conditions, the global user similarity can improve the performance of algorithms that use only local similarities.

Another approach for solving the data sparsity problem was proposed by Bessa et al. [37], who advanced a method that considers user communities with similar tastes and predicts new relations within these communities. The results reveal that although this method did not increase the global coverage, it improved the predictions of already covered items, alleviating some of the drawbacks of the sparsity.
problem. Hawashin et al. [38] proposed a hybrid similarity measure based on explicit user interests; the proposed method achieved good performance even when no co-rated items existed between two users. However, it did not consider the semantic meanings of the ratings, and it depended on the existence and quality of explicit user interests. Moreover, the process of calculating user interests was computationally expensive.

III. PROPOSED METHOD

In this section, we detail our proposed novel CF approach for predicting user rating values. Our method was inspired by the strengths of previous approaches while also considering their weaknesses. However, the proposed method retains the characteristics that make neighborhood models widely used; that is, their relative simplicity, ease of maintenance, and high explainability. In real-world use scenarios, neighborhood approaches should remain easy to tune and should not depend on complex optimization procedures that require long computation times, especially if they must be updated every time a new user or item is added.

Our proposed recommender, SHB, is an ensemble of a switching hybrid and a basic collaborative filter with JAC. The switching hybrid consists of a CF method that uses our proposed similarity measure, and a CF technique that computes the prediction based on global statistics. To determine the final prediction, first, the target user is given similarity scores with respect to the other users. These serve as the weights in Eq. 2 for predicting the desired rating. The proposed similarity measure used in this step considers the distance between user ratings as well as the deviation between these ratings and the median of the rating scale. If the user-item matrix is highly sparse, the target user might not have sufficient neighbors to make a meaningful prediction; that is, the recommender will rely on users with low similarity with the target, which may often lead to inaccurate predictions. In this situation, the switching hybrid relies instead on the biases for the target user and item as well as on the global mean. As a rule of thumb, the switching criterion considers a minimum of 10 neighbors as the threshold. Finally, an alternative prediction is derived from a neighborhood-based recommender using JAC. This is combined with the prediction of the switching hybrid to yield the final rating.

A. PROPOSED SIMILARITY MEASURE

As explained in Section II-B, several studies have exhibited promising results by incorporating the semantic meaning of the ratings when defining new similarity measures. Studying the semantic nuances of user ratings is essential for developing a measure of similarity that can realistically quantify the resemblance among user tastes. The purpose of a rating scale is to allow respondents to express both the direction and strength of their opinions regarding a topic [39]. Ratings, unlike other implicit measures of interest, such as total clicks or number of items purchased, are ordinal categorizations that can be assumed to reflect the user’s degree of indifference toward an item. We observe that to compare two users’ ratings, it is not only useful to consider the difference between the ratings but also how far the ratings are from the rating scale’s median. The intuition behind this is that users who give ratings closer to the median tend to feel more indifferent toward the item, whereas users with more extreme ratings reflect stronger preferences. Furthermore, we consider that stronger preferences provide more reliable information for determining whether or not two users have similar tastes.

We propose a new similarity measure that considers two main aspects: the difference between user ratings and how extreme these are. For users and who have rated item with ratings and respectively, the distance is determined as follows:

\[ \text{sqd}(r_{u,i}, r_{v,i}) = (r_{u,i} - r_{v,i})^2. \]  

(5)

The squared difference is used to ensure that this is a smooth and symmetric measure of the distance between ratings. Without loss of generality, let \( R = \{1, 2, \ldots, R_r\} \) represent an integer rating scale with a median given as follows:

\[ \bar{r}_s = \frac{R_s + 1}{2}. \]  

(6)

The rating strength \( r_p \) for a pair of user ratings is defined as follows:

\[ r_p(r_{u,i}, r_{v,i}) = \frac{\max(|r_{u,i} - \bar{r}_s|, |r_{v,i} - \bar{r}_s|)}{\bar{r}_s - 1}. \]  

(7)

This satisfies \( 0 \leq r_p \leq 1 \) and reflects how extreme the ratings are. Higher \( r_p \) values indicate more extreme ratings. A value of 1 indicates that at least one of \( r_{u,i} \) or \( r_{v,i} \) is equal to either the lowest or highest possible rating. Meanwhile, a value of 0 implies that both \( r_{u,i} \) and \( r_{v,i} \) are equal to the median.

A measure of dissimilarity \( \text{dis}(u, v) \) between users and can now be expressed by the following equation:

\[ \text{dis}(u, v) = \frac{1}{|I_u \cap I_v|} \sum_{i \in I_u \cap I_v} \text{sqd}(r_{u,i}, r_{v,i}) \cdot r_p(r_{u,i}, r_{v,i}), \]  

(8)

where \( I_u \) and \( I_v \) denote the sets of items rated by users and respectively. The corresponding similarity measure between users and is defined as follows:

\[ \text{sim}(u, v) = \frac{1}{\text{dis}(u, v) + 1}. \]  

(9)

B. PREDICTED RATINGS FROM USER AND ITEM BIASES

In highly sparse datasets, the target user may have very few neighbors. When this happens, ratings predicted using Eq. 1 are influenced by users with low similarity to the target. This is problematic because users who are weakly correlated with the target user may cause the recommender to yield inaccurate predictions. This phenomenon can also be explained as overfitting caused by the normalization factor in Eq. 1 when only low-similarity neighbors are present. Moreover, when the target user has no neighbors, the rating for that specific user-item combination cannot be predicted. In these cases, the naive approach involves reporting the global average or
the average of all the ratings by the target user [17]. As an alternative, we consider the global, item, and user averages to compute the predicted rating when the number of neighbors found is insufficient.

In our proposed method, we first calculate the user bias (the deviation between the user rating average and total average) and add this to the item average. Formally, the predicted rating \( \hat{r}_{u,i} \) of user \( u \) for item \( i \) is calculated as follows:

\[
\hat{r}_{u,i} = \bar{r}_i + (\bar{r}_u - \bar{r})
\]

(10)

where \( \bar{r}_i \) denotes the average rating for item \( i \) by all users who rated it, \( \bar{r}_u \) is the average rating user \( u \) gave to the items they rated, and \( \bar{r} \) is the overall average of all observed ratings. The term \( \bar{r}_u - \bar{r} \) represents user bias. A user has positive bias if the user has an inclination to rate items favorably. Conversely, when the user’s ratings tend to be critical, the user exhibits a negative bias. Following this approach, the ratings of all users for the target item are considered in equal proportion but are regulated by the preferences of the target user, encoded in their rating trends.

A similar idea was proposed by Koren [25], who observed that a baseline estimate for a user-item pair \( b_{u,i} \) can be calculated as follows:

\[
b_{u,i} = \bar{r} + b_u + b_i
\]

(11)

where \( b_u \) and \( b_i \) indicate the observed deviations from the average for user \( u \) and item \( i \), respectively. To estimate \( b_u \) and \( b_i \), Koren proposes solving the following regularized least-squares optimization problem:

\[
\min_{b_u, b_i} \sum_{u,i \in R} (r_{u,i} - \bar{r} - b_u - b_i)^2 + \lambda \left( \sum_u b_u^2 + \sum_i b_i^2 \right)
\]

(12)

where the first sum runs over the elements present in the sparse user-item matrix \( R \). The second expression is a regularization term to avoid overfitting on the optimization procedure, and \( \lambda \) is the regularization parameter.

The optimization of Eq. 12 must be performed over a sparse matrix, typically using approximate methods such as stochastic gradient descent (SGD). Thus, the optimal values of the biases \( b_u \) and \( b_i \) are updated in steps, following the slope (gradient) of the objective function downward, until a minimum is reached. This is not the case for our proposed method in Eq. 10, where the biases are approximated by the explicit formulas \( b_u = \bar{r}_u - \bar{r} \) and \( b_i = \bar{r}_i - \bar{r} \). Our experiments in Section IV provide evidence that both approaches yield comparable results when applied to our SHB proposal.

C. PROPOSED RECOMMENDER SYSTEM

Subsections III-A and III-B introduce two different methods to make a rating prediction, each of which performs best under different situations. To choose the optimal method, a switching hybrid method is introduced. The first recommender R1 (neighborhood-based) yields better predictions if a set of representative neighbors can be found in the dataset. Meanwhile, the second recommender R2 (based on rating biases) can more adequately capture the underlying patterns encoded by the global statistics, which prove useful if a better-customized prediction is not available. Consequently, we propose a switching criterion that selects R2 if the number of neighbors is below a threshold value, and R1 otherwise.

The recommendations of the switching hybrid recommender are as follows:

\[
\hat{r}_{u,i} = \begin{cases} 
\hat{r}_{u,i}^{(1)}, & \text{if } |N(u, i)| \geq \text{threshold} \\
\hat{r}_{u,i}^{(2)}, & \text{otherwise}
\end{cases}
\]

(13)

where \( \hat{r}_{u,i}^{(1)} \) denotes the recommendation of R1, calculated using the formula in Eq. 2. The recommendation of R2, \( \hat{r}_{u,i}^{(2)} \), is computed using Eq. 10. The threshold value was determined heuristically from the analysis of diverse datasets and set to 10 in our experiments.

As a final step, we introduce a hyperparameter \( \alpha \) to combine the predicted rating from the switching hybrid with that of a standard recommender using JAC. This yields the final prediction of the proposed SHB as follows:

\[
\hat{r}_{u,i} = \alpha \cdot \hat{r}_{u,i}^{(1)} + (1 - \alpha) \left( \bar{r} + \frac{\sum_{v \in N(u,i)} \text{JAC}(u, v) \cdot (r_{v,i} - \bar{r}_v)}{\sum_{v \in N(u,i)} \text{JAC}(u, v)} \right)
\]

(14)

The following is an intuitive, step-by-step description of the proposed algorithm, which is also illustrated in Fig. 1.

1) Compute the prediction \( \hat{r}_{u,i}^{(1)} \) using Eq. 2 and the similarity measure defined in Eq. 9.
2) Compute the prediction \( \hat{r}_{u,i}^{(2)} \) as described in Eq. 10.
3) If \( |N(u, i)| \) (total neighbors of \( u \) for item \( i \)) is greater than or equal to 10, set \( \hat{r}_{u,i} \) equal to \( \hat{r}_{u,i}^{(1)} \), otherwise set it to \( \hat{r}_{u,i}^{(2)} \).
4) Calculate an additional rating prediction using Eq. 2 with JAC as the similarity metric.
5) Compute the final rating \( \hat{r}_{u,i} \) as the linear combination (weighted by hyperparameter \( \alpha \)) of the ratings calculated in the previous two steps, as shown in Eq. 14.

D. COMPLEXITY AND SCALABILITY ANALYSIS OF THE RECOMMENDER SYSTEM

To consider the computational complexity of the proposed recommender system we use the Big O notation. This expresses the asymptotic performance bounds that an algorithm which implements the recommender can be expected to achieve.

The amount of computations required by the proposed method depends on the number of users and items in the dataset. As illustrated in Fig. 1, the prediction of rating scores requires the calculation of the user similarity matrix from the user-item matrix. This is the most computationally expensive operation in the recommender, and thus it determines its time and space complexities, which scale non-linearly with the number of users and items. Algorithm 1 summarizes this procedure, and shows that the total number of similarity
FIGURE 1. Overview of the proposed recommender system. The sparse user-item matrix is used to estimate the similarities and global statistics between users. These serve as inputs to a nested pair of hybrid recommenders.

The high time complexity involved in the computation of user similarity matrices makes their online update impractical for situations such as e-commerce systems, in which the number of users and items can exceed tens or hundreds of millions. This poor scalability problem is a known challenge that affects not only our proposed method, but is inherent to neighborhood-based methods in general. Several works have attempted to address the scalability problem in CF. While it is beyond the scope of this paper to conduct an extensive analysis of these techniques, it is worth mentioning that common applications need not update the similarity matrix online. Given that its calculation could take hours or even days to complete for large real-world datasets, it is common to pre-compute it offline and assume it remains constant throughout a subsequent, low-latency online phase which computes the rating predictions. Additionally, the memory requirements to store the pre-computed similarity matrix can be significantly reduced if, instead of storing it entirely, only the tuples of users with a similarity larger than a predetermined threshold are kept.

IV. EXPERIMENTAL SETTING

In this section, we present the experiments conducted to test the performance of the proposed SHB recommender. Our results were compared with those of five different neighborhood-based approaches, two MF methods, and the global average as a baseline. Each neighborhood-based approach uses a different similarity measure. We selected PCC, COS, and JAC, as well as the hybrid variations PCC+JAC and COS+JAC because of their widespread use in practical applications. Other related approaches were not included in our experiments because of their drawbacks detailed in Section II. In particular, they might require extensive dataset-dependent parameter tuning to reach their optimal results or additional user or item features to complement their predictions. Furthermore, some of the most complex approaches do not provide available implementations that can scale to the size of the datasets used in this experiment.

The tests were conducted using five real-world public datasets and one synthetic dataset. This work considers the application of RS to predict user preferences (in the form of ratings); therefore, we measured their accuracy by comparing the predicted rating against the ground truth. We report the accuracy in terms of the root-mean-square error (RMSE) and mean absolute error (MAE), which are two commonly used metrics for evaluating the performance of recommendation systems. The RMSE measures the average magnitude of the error in the predictions, while the MAE gives an idea of the magnitude of the prediction errors, regardless of their sign.

Algorithm 1 Similarity Matrix Calculation for an $n \times m$ User-Item Matrix $R$

Input: user-item matrix $R$

Output: $n \times n$ user similarity matrix

1: for $i \leftarrow 1$ to $n$ do
2:     $S[i, i] \leftarrow 1$
3:     for $j \leftarrow i + 1$ to $n$ do
4:         $S[i, j] \leftarrow \text{similarity}(i, j)$
5:         $S[j, i] \leftarrow S[i, j]$
6:     end for
7: end for
8: return $S$
used metrics in the literature. These metrics are defined as follows [5]:

\[
RMSE = \sqrt{\frac{1}{|T|} \sum_{(u,i) \in T} (r_{u,i} - \hat{r}_{u,i})^2}, \quad (15)
\]

and

\[
MAE = \frac{1}{|T|} \sum_{(u,i) \in T} |r_{u,i} - \hat{r}_{u,i}|, \quad (16)
\]

where \( T \) denotes the test set, \( r_{u,i} \) the ground truth, and \( \hat{r}_{u,i} \) the predicted rating. The tests were performed using the k-fold cross-validation method. The reported results are the average of the cross-validation trials, in which smaller values of RMSE and MAE indicate better predictive accuracy. The plots presented were calculated with error bands showing 99% confidence intervals.

The two error metrics, RMSE and MAE, each provide a different piece of information which may be more useful...
TABLE 2. Characteristics of the real-world datasets used in the experiments.

| Dataset          | Total users | Total items | Total ratings | Rating scale | Sparsity | Items category | Source                        |
|------------------|-------------|-------------|---------------|--------------|-----------|----------------|-------------------------------|
| MovieLens 100k   | 610         | 9,724       | 100,336       | [0.5, 5]     | 0.983     | Movies         | MovieLens website            |
| MovieLens 1M     | 6,040       | 3,706       | 1,000,209     | [1, 5]       | 0.9553    | Movies         | MovieLens website            |
| Epinions         | 40,163      | 139,738     | 664,123       | [1, 5]       | 0.9995    | Consumer items | Epinions website             |
| Book-crossing    | 105,283     | 340,556     | 1,149,780     | [0, 10]      | 0.9999    | Books          | Book-crossing website        |
| Jester (dataset 3) | 54,905     | 140         | 1,842,370     | [-10, 10]    | 0.7603    | Jokes          | Jester joke recommender system |

 depending on the use case for the recommender. The squared component in the RMSE indicates large errors, which makes it more suitable in applications where even a few large deviations are undesired. In contrast, MAE is a more impartial indicator of the typical error and tends to be more robust to outliers. Another approach used to evaluate RS involves considering the task as a classification problem. When the recommendation task is not to predict ratings, but to generate a list of interesting items, then the score is only used to determine if the item should be recommended or not. In this case, metrics such as precision and recall are more suitable. If the order of the recommended list is also important, then metrics such as normalized discounted cumulative gain and mean reciprocal rank are useful to evaluate both the adequacy of the selected recommendations and their ranking.

A. EVALUATION DATASETS

The real-world datasets used in our experiments are summarized in Table 2. Different types of datasets were used to verify the generalization power of the results. All real-world datasets are publicly available for research use and vary in terms of the total number of users, items, ratings, sparsity, and the type of items being rated. MovieLens 100k and MovieLens 1M [45] contain the ratings of users who evaluated at least 20 movies. These users were randomly sampled from the complete MovieLens dataset, which contains 26,000,000 ratings from 270,000 users and 45,000 movies. Epinions [46] and Book-crossing [47] data were collected using a crawler that browsed over epinions.com and bookcrossing.com, respectively. The Epinions dataset comprises user ratings from various consumer items, whereas the Book-crossing dataset comprises book ratings by members of an online book club. The complete Jester [48] dataset contains 6.5 million anonymous ratings of jokes that were collected from users of the Jester joke recommender system. Consequently, we used Jester (dataset 3), which is similar to the MovieLens datasets, containing a subset of the complete dataset after some jokes were removed and only users who rated 36 or more jokes were included. In the case of the Epinions and Book-crossing datasets, we applied similar filtering to retain only the data for users who rated at least 20 items.

To generate data for the synthetic dataset, we relied on the following assumptions:

- Users can be classified into types, where a type is a cluster of users who share similar tastes and interests. CF operates under the same assumption, identifying users in the same cluster, and calculating ratings based on the proximity of users within the cluster.

- The probability of a user being included in a particular user type is the same for all types. This assumption might not hold in real-world settings because some types might be more popular than others. However, the probability was set to be uniform to simplify the data generation process.

- All elements of the full matrix have the same probability of being missing in the sparse matrix. This assumption will not reproduce the known biases in the rating data. For example, users may be more likely to rate items for which they have strong opinions or some users may provide many more ratings than others [5].

The ratings in the synthetic dataset are the output of a random process modeled by a categorical distribution with probabilities for each possible score given for each pair of user type ($U_T$) and item ($i$). These probabilities were decided from a separate random process such that the histogram of the ratings for the ($U_T$, $i$)-pair will approximate a gaussian with a randomly selected mean. In addition, they are chosen to ensure that the full dataset has a specified mean and standard deviation.

The parameters used to generate the synthetic data are listed in Table 3.

The advantages of having a synthetic dataset are that it allows us to freely evaluate the proposed method under different conditions, such as different levels of sparsity or types of users, while having complete knowledge regarding the underlying patterns in the data as well as the ground truth for the full user-item matrix. In particular, this dataset allowed us to study the performance of different RS as sparsity increases. The same analysis using a subset of real-world datasets would have added the risk of bias of the data by the subsampling scheme.

V. RESULTS AND DISCUSSION

Our first analysis focuses on the proposed similarity metric of Eq. 9 and compares its performance with that of other
conventional similarity measures, along with the global mean as a baseline. To focus exclusively on the impact of the choice of similarity, we computed the rating predictions using Eq. 1. Therefore, this first analysis does not consider the proposed recommender SHB in its entirety but instead investigates the effects of the proposed similarity (SIM) on its own. Fig. 2 shows different plots of the RMSE and MAE for different values of k-NN. All similarities attained their highest RMSE and MAE values when the number of neighbors was small; as the number of neighbors increased, the performance improved asymptotically. Moreover, excluding the Book-crossing dataset, the proposed similarity (referred to as SIM in the figure) achieved the best performance, with the lowest RMSE and MAE values, indicating that the semantic information exploited by our method can efficiently discover rating patterns among like-minded users. Furthermore, the proposed method exhibits a more consistent behavior across datasets than other similarity metrics.

The second part of the analysis considered the proposed recommender SHB based on Eq. 14. We compared its performance with those of other CF methods. In addition, two widely used MF recommenders, FunkMF and SVD++, were included in the comparison. Although these MF approaches are known to have better performance than neighborhood-based methods, this is achieved at the cost of interpretability and a higher maintenance cost. Note that the purpose of including this comparison was not to suggest that neighborhood-based models can outperform MF approaches. In many cases, both approaches complement each other in large-scale systems. Rather, our reason to examine these differences was that in some particular applications, a small loss in performance can be outweighed by other practical benefits, such as better interpretability or flexibility to add new users and items. Tables 4 and 5 summarize the RMSE and MAE values for the different CF methods applied to all datasets. The lowest values for each row are marked in bold. In addition, to emphasize the distinction between neighborhood-based and MF methods, Tables 4 and 5 also indicate in bold the best result among the neighborhood-based models even if they are outperformed by the MF ones (in this case, the MF value, also in bold, is marked with a star). The bottom rows of the tables show the average error across all datasets.

Table 6 presents the scores from statistical significance tests between the proposed method and the other neighborhood-based approaches from Tables 4 and 5. From Tables 4 and 5, we can confirm that while the MF methods tend to have the lowest error, the proposed recommender achieves comparable levels of performance. Furthermore, it appears to generalize better to different datasets, achieving the lowest total average error among all neighborhood-based approaches. It also outperforms other neighborhood-based methods on datasets with the highest sparsity. On datasets of relatively higher density, several CF methods, including the proposed method, seem to reach a minimum error level. One interesting result that can be observed is that FunkMF and SVD++ significantly outperformed the other techniques when applied to the synthetic dataset. This is explained by noting that the synthetic dataset does, in fact, consist of a few separately defined user types and therefore naturally lends itself to the low-dimensional representation of the MF approaches.

The third part of the analysis considers again the performance of the proposed SHB recommender compared with that of other CF methods, focusing on the error metrics when different numbers of nearest neighbors are considered. The results are shown in Fig. 3. The proposed method achieves the lowest or close to the lowest errors in all datasets. The RMSE and MAE values for SHB stabilized when approximately 20 neighbors were considered, implying that the
The proposed method is well suited to datasets that are highly sparse. As for other methods, PCC and COS both benefit from being combined with JAC; this result is consistent with the findings in [12]. When considering the MovieLens 1M and Jester (dataset 3) datasets, these hybrid methods perform similarly to the proposal. This can be explained by noting that both datasets have relatively low sparsity. Therefore, more high-quality neighbors with more ratings can be found, and more reliable PCC and COS scores can be obtained.

The results for the Book-crossing dataset exhibit the best performance for all methods that include a JAC component. Furthermore, the performance appears to be independent of the choice of k-NN for all of them except SHB. Our proposal also converges to this value, reaching it for a k-NN of approximately 20. This behavior may be due to the high sparsity of the Book-crossing dataset (99.96%). The intersections between user vectors are very limited, thus restricting the precision of all similarity calculations. In this situation, the intersection of rated movies considered by JAC is the...
most reliable measure of similarity. This demonstrates that in general terms, JAC focuses on unique aspects of the data and therefore tends to complement the other similarity measures.

Although other similarities attempt to identify like-minded users from their ratings, JAC only considers whether the users rated an item or not, irrespective of the score given. Although JAC discards information exploited by other similarity measures, it still retains a strong implicit signal of preference because users do not choose at random the items to rate. It may be argued that the likelihood of a user rating an item depends, in part, on their perceived investment and expected return when consuming the item. For example, watching a movie requires investing a few hours; thus, users may be more inclined to watch those movies that they believe they will like. The act of rating a movie that the users must have watched is, therefore, not entirely random because the movie had to be selected first, and this selection bears an implicit signal of preference.

Another observation is that for some datasets, SHB reaches an optimal performance and then slowly deteriorates as k-NN increases. This trend can be explained by the effects of weakly correlated neighbors being added in the rating prediction. In the extreme case where all other users are treated as neighbors, even as the closest user has a large weight, a high number of low-similarity neighbors would introduce an appreciable noise, which ultimately affects the recommender performance. This phenomenon, along with the previously observed stabilization of the error from approximately 20 k-NNs, indicates that SHB only requires a small number of neighbors to attain its best predictions.

Next, we analyze the effects of our proposed method shown in subsection III-B by comparing the accuracy of two variations of SHB. The first variation uses the simplified predictions of Eq. 10 when the switching criterion is met. For the second variation, an alternative implementation replaces these predictions with those from the method proposed by [25], as shown in Eq. 12. The results, summarized in Table 7, indicate that both approaches yield comparable performance when used in conjunction with the proposed SHB recommender.

We now consider the performance of SHB with the contribution of JAC being set to zero ($\alpha = 1$) and compare it with the other CF methods when they are applied to the
FIGURE 5. Recommendation performance of different CF approaches. The RMSE and MAE are reported over different values for the hyperparameter $\alpha$. The synthetic dataset generated using different levels of sparsity.

The results are shown in Fig. 4. The first observation is that the RMSE for PCC and COS exhibits an approximately concave behavior with a pronounced decrease in performance as sparsity increases up to a specific value. Counterintuitively, the performance appears to improve for higher sparsities. The reason for this is that if the dataset becomes too sparse, there will be several cases where no neighbors are found. In this situation, the second term of Eq. 2 is taken to be zero, and thus, the predicted rating becomes the user mean. This is also the reason for the convergence of all methods to the same value as sparsity approaches 1. Although the performance of SHB also decreases as the dataset becomes more sparse, it retains some degree of accuracy even at very high sparsity levels until it reaches the user mean when there are no neighboring users in the dataset. Other methods that include a JAC contribution also exhibit this trend; however, the proposed method outperforms them at lower sparsities. A possible interpretation is that the information being exploited by SHB is more meaningful than that used by the other CF methods.
Finally, we consider the influence of JAC on the performance of the different CF approaches. Fig. 5 shows the RMSE and MAE for different values of the hyperparameter $\alpha$. The results show that including JAC leads to better predictions for all methods and datasets with the exception of COS+$\text{JAC}$ applied to the Epinions dataset, in which case the MAE is the smallest without the contribution of JAC. It can be observed that the proposed SHB method exhibits convex behavior, implying the existence of an optimal value of $\alpha$. This indicates that while one term in Eq. 14 overestimates the rating, the other must be underestimating it and, therefore, provides further evidence that each term focuses on different aspects of the data.

VI. CONCLUSION

We proposed a new similarity measure designed to overcome the drawbacks of conventional similarities. The proposed method considers the semantic nuances of ratings, which quantify the degree of resemblance between user tastes as represented on an ordinal scale. This makes it suitable for applications in which measuring a user’s degree of indifference toward an item is important. In addition, we introduced an RS based on an ensemble of a switching hybrid and a CF method with JAC. The switching hybrid relies on the predictions made either using a CF technique that uses the proposed similarity measure or from the global statistics of the dataset. Our approach is designed to retain the simplicity and explainability aspects which make neighborhood-based approaches a mainstream technique used in real-world applications.

The predictive accuracy of SHB was evaluated using the RMSE and MAE, and its performance was compared with that of other CF approaches. Experiments were conducted using five real-world datasets and a synthetic dataset. The results indicate that the error when using the proposed method is either lower or matches that of the best predictions obtained from conventional similarity measures. In particular, SHB consistently outperformed the other methods when applied to datasets with high sparsity. It achieved its best performance using only a few neighbors, making it more scalable. However, a small amount of overfitting was observed if k-NN was set excessively high, thus negatively affecting the quality of the predictions. Our analysis also confirms that combining individual similarity measures with JAC can consistently improve their performance. We observed that JAC encodes a strong signal of implicit preferences that effectively complements the aspects found by other similarities.

ACKNOWLEDGMENT

The authors thank Jorge Trevino for his valuable comments.

REFERENCES

[1] G. Klein, J. Orasana, and R. Calderwood, Decision Making in Action: Models and Methods (Cognition and Literacy Series). Norwood, NJ, USA: Ablex, 1993.

[2] T. Gilovich, D. Griffin, D. Kahneman, and C. U. Press, Heuristics and Biases: The Psychology of Intuitive Judgment. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[3] A. Chernev, U. Böckenholt, and J. Goodman, “Choice overload: A conceptual review and meta-analysis,” J. Consum. Psychol., vol. 25, no. 2, pp. 333–358, Apr. 2015.

[4] M. Solomon, Consumer Behavior: Buying, Having, and Being. London, U.K.: Pearson, 2017.

[5] F. Ricci, L. Rokach, and B. Shapira, Recommender Systems Handbook. New York, NY, USA: Springer, 2015.

[6] R. Burke, Hybrid Web Recommender Systems. Berlin, Germany: Springer, 2007, pp. 377–408.

[7] G. R. Foxall, Understanding Consumer Choice, 1st ed. London, U.K.: Palgrave Macmillan, 2005.

[8] S. Cai and Y. Xu, “Effects of outcome, process and shopping enjoyment on online consumer behaviour,” Electron. Commerce Res. Appl., vol. 5, no. 4, pp. 272–281, Dec. 2006.

[9] J. B. Schafer, J. Konstan, and J. Riedl, “Recommender systems in e-commerce,” in Proc. 1st ACM Conf. Electron. Commerce (EC), New York, NY, USA: Association for Computing Machinery, 1999, pp. 158–166.

[10] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, “Collaborative filtering and deep learning based recommendation system for cold start items,” Expert Syst. Appl., vol. 69, pp. 29–39, Mar. 2017.

[11] F. Liu and H. J. Lee, “Use of social network information to enhance collaborative filtering performance,” Expert Syst. Appl., vol. 37, no. 7, pp. 4772–4778, Jul. 2010.

[12] L. Candillier, F. Meyer, and F. Fessant, “Designing specific weighted similarity measures to improve collaborative filtering systems,” in Advances in Data Mining. Medical Applications, E-Commerce, Marketing, and Theoretical Aspects. Berlin, Germany: Springer, 2008, pp. 242–255.

[13] Z. Tan and L. He, “An efficient similarity measure for user-based collaborative filtering recommender systems inspired by the physical resonance principle,” IEEE Access, vol. 5, pp. 27211–27228, 2017.

[14] U. Shardanand and P. Maes, “Social information filtering: Algorithms for automating ‘word of mouth,’” in Proc. SIGCHI Conf. Hum. Factors Comput. Syst. (CHI). New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1995, pp. 210–217.

[15] R. M. Bell and Y. Koren, “Scalable collaborative filtering with jointly derived neighborhood interpolation weights,” in Proc. 7th IEEE Int. Conf. Data Mining, Oct. 2007, pp. 43–52.

[16] H. Luo, C. Niu, R. Shen, and C. Ulrich, “A collaborative filtering framework based on both local user similarity and global user similarity,” Mach. Learn., vol. 72, no. 3, pp. 231–245, Sep. 2008.

[17] C. C. Aggarwal, Recommender Systems: The Textbook, 1st ed. Cham, Switzerland: Springer, 2016.

[18] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, “Methods and metrics for cold-start recommendations,” in Proc. 25th Annul. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR). New York, NY, USA: Association for Computing Machinery, 2002, pp. 253–260.

[19] J. Bobadilla, F. Ortego, A. Hernando, and J. Bernal, “A collaborative filtering approach to mitigate the new user cold start problem,” Knowl.-Based Syst., vol. 26, pp. 225–238, Feb. 2012.

[20] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, “Using collaborative filtering to weave an information tapestry,” Commun. ACM, vol. 35, no. 12, pp. 61–70, Dec. 1992.

[21] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “GroupLens: An open architecture for collaborative filtering of netnews,” in Proc. ACM Conf. Comput. Supported Cooperat. Work (CSCW). New York, NY, USA: Association for Computing Machinery, 1994, pp. 175–186.

[22] R. Salakhutdinov, A. Mnih, and G. Hinton, “Restricted Boltzmann machines for collaborative filtering,” in Proc. 24th Int. Conf. Mach. Learn. (ICML). New York, NY, USA: Association for Computing Machinery, 2007, pp. 791–798.

[23] R. Bell, Y. Koren, and C. Volinsky, “Modeling relationships at multiple scales to improve accuracy of large recommender systems,” in Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD). New York, NY, USA: Association for Computing Machinery, 2007, pp. 95–104.

[24] S. Funk, “Netflix update: Try this at home,” Tech. Rep., Dec. 2006. [Online]. Available: https://sifter.org/simon/journal/20061211.html

[25] Y. Koren, “Factorization meets the neighborhood: A multifaceted collaborative filtering model,” in Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), 2008, pp. 426–434.

[26] D. Anand and K. K. Bharadwaj, “Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities,” Expert Syst. Appl., vol. 38, no. 5, pp. 5101–5109, May 2011.
H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, “A new user similarity model for recommender systems,” ACM Trans. Web, vol. 10, no. 2, pp. 1–30, May 2016.

A. Gazdar and L. Hidri, “A new similarity measure for collaborative filtering based recommender systems,” Knowl.-Based Syst., vol. 188, Jan. 2020, Art. no. 105058.

L. A. Hassanieh, C. A. Jaoudeh, J. B. Abdo, and J. Demerjian, “Similarity measures for collaborative filtering recommender systems,” in Proc. IEEE Middle East North Afr. Commun. Conf. (MENACOMM), Apr. 2018, pp. 1–5.

H. J. Ahn, “A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem,” Inf. Sci., vol. 178, no. 1, pp. 37–51, Jan. 2008.

H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, “A new user similarity model to improve the accuracy of collaborative filtering,” Knowl.-Based Syst., vol. 56, pp. 156–166, Jan. 2014.

A. Said, B. J. Jain, and S. Albayrak, “Analyzing weighting schemes in collaborative filtering: Cold start, post cold start and power users,” in Proc. 27th Annu. ACM Symp. Appl. Comput. (SAC), New York, NY, USA: Association for Computing Machinery, 2012, pp. 2035–2040.

R. N. Laveti, J. Ch, S. N. Pal, and N. S. C. Babu, “A hybrid recommender system using weighted ensemble similarity metrics and digital filters,” in Proc. IEEE 23rd Int. Conf. High Perform. Comput. Workshops (HiPCW), Dec. 2016, pp. 32–38.

J. S. Breece, D. Heckerman, and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering,” in Proc. 14th Conf. Uncertainty Artif. Intell. (UAI), San Francisco, CA, USA: Morgan Kaufmann, 1998, pp. 43–52.

J. Wang, A. P. de Vries, and M. J. T. Reinders, “Unifying user-based and item-based collaborative filtering approaches by similarity fusion,” in Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR), New York, NY, USA: Association for Computing Machinery, 2006, pp. 501–508.

J. Zhang and P. Pu, “A recursive prediction algorithm for collaborative filtering recommender systems,” in Proc. ACM Conf. Recommender Syst., New York, NY, USA: Association for Computing Machinery, 2007, pp. 57–64.

A. Bessa, A. Laender, A. Veloso, and N. Ziviani, “Alleviating the sparsity problem in recommender systems by exploring underlying user communities,” in Proc. 6th Alberto Mendelzon Int. Workshop Found. Data Manage., Ouro Preto, Brazil, Jun. 2012, pp. 35–47. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.366.5905

B. Hawashin, M. Lafi, T. Kanan, and A. Mansour, “An efficient hybrid similarity measure based on user interests for recommender systems,” Expert Syst., vol. 37, no. 5, p. e12471, Aug. 2019.

R. Garland, “The mid-point on a rating scale: Is it desirable,” Marketing Bull., vol. 2, pp. 66–70, 1991.

K. Yu, X. Xu, J. Tao, M. Ester, and H.-P. Kriegel, “Instance selection techniques for memory-based collaborative filtering,” in Proc. SIAM Int. Conf. Data Mining, Apr. 2002, pp. 59–74.

A. S. Das, M. Datar, A. Garg, and S. Rajaram, “Google news personalization: Scalable online collaborative filtering,” in Proc. 16th Int. Conf. World Wide Web (WWW), New York, NY, USA: Association for Computing Machinery, 2007, pp. 271–280.

G. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Recommender systems for large-scale e-commerce scalable neighborhood formation using clustering,” in Proc. 5th Int. Conf. Comput. Inf. Technol., vol. 50, Jan. 2002, pp. 291–324.

G.-R. Xue, C. Lin, Q. Yang, W. Xi, H-J. Zeng, Y. Yu, and Z. Chen, “Scalable collaborative filtering using cluster-based smoothing,” in Proc. 28th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR), New York, NY, USA: Association for Computing Machinery, 2005, pp. 21–114.

G. Takács, I. Pilászy, B. Németh, and D. Tikk, “Scalable collaborative filtering approaches for large recommender systems,” J. Mach. Learn. Res., vol. 10, pp. 623–656, Jun. 2009.

F. M. Harper and J. A. Konstan, “The MovieLens datasets: History and context,” ACM Trans. Interact. Intell. Syst., vol. 5, no. 4, pp. 1–19, Jan. 2016.

P. Massa, K. Souren, M. Salvetti, and D. Tomasoni, “Trustlet, open research on trust metrics,” Scalable Comput., Pract. Exper., vol. 9, pp. 1–11, Jan. 2008.

C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, “Improving recommendation lists through topic diversification,” in Proc. 14th Int. Conf. World Wide Web (WWW), New York, NY, USA: Association for Computing Machinery, 2005, pp. 22–32.

K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, “Eigentaste: A constant time collaborative filtering algorithm,” Inf. Retr., vol. 4, no. 2, pp. 133–151, 2001.

P. Ortal, M. Edahiro: Switching Hybrid Method Based on User Similarity and Global Statistics for CF

[27] G. Guo, J. Zhang, and N. Yorke-Smith, “A novel evidence-based Bayesian similarity measure for recommender systems,” ACM Trans. Web, vol. 10, no. 2, pp. 1–30, May 2016.

[28] A. Gazdar and L. Hidri, “A new similarity measure for collaborative filtering based recommender systems,” Knowl.-Based Syst., vol. 188, Jan. 2020, Art. no. 105058.

[29] L. A. Hassanieh, C. A. Jaoudeh, J. B. Abdo, and J. Demerjian, “Similarity measures for collaborative filtering recommender systems,” in Proc. IEEE Middle East North Afr. Commun. Conf. (MENACOMM), Apr. 2018, pp. 1–5.

[30] H. J. Ahn, “A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem,” Inf. Sci., vol. 178, no. 1, pp. 37–51, Jan. 2008.

[31] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, “A new user similarity model to improve the accuracy of collaborative filtering,” Knowl.-Based Syst., vol. 56, pp. 156–166, Jan. 2014.

[32] A. Said, B. J. Jain, and S. Albayrak, “Analyzing weighting schemes in collaborative filtering: Cold start, post cold start and power users,” in Proc. 27th Annu. ACM Symp. Appl. Comput. (SAC), New York, NY, USA: Association for Computing Machinery, 2012, pp. 2035–2040.

[33] R. N. Laveti, J. Ch, S. N. Pal, and N. S. C. Babu, “A hybrid recommender system using weighted ensemble similarity metrics and digital filters,” in Proc. IEEE 23rd Int. Conf. High Perform. Comput. Workshops (HiPCW), Dec. 2016, pp. 32–38.

[34] J. S. Breece, D. Heckerman, and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering,” in Proc. 14th Conf. Uncertainty Artif. Intell. (UAI), San Francisco, CA, USA: Morgan Kaufmann, 1998, pp. 43–52.

[35] J. Wang, A. P. de Vries, and M. J. T. Reinders, “Unifying user-based and item-based collaborative filtering approaches by similarity fusion,” in Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR), New York, NY, USA: Association for Computing Machinery, 2006, pp. 501–508.

[36] J. Zhang and P. Pu, “A recursive prediction algorithm for collaborative filtering recommender systems,” in Proc. ACM Conf. Recommender Syst., New York, NY, USA: Association for Computing Machinery, 2007, pp. 57–64.

[37] A. Bessa, A. Laender, A. Veloso, and N. Ziviani, “Alleviating the sparsity problem in recommender systems by exploring underlying user communities,” in Proc. 6th Alberto Mendelzon Int. Workshop Found. Data Manage., Ouro Preto, Brazil, Jun. 2012, pp. 35–47. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.366.5905

[38] B. Hawashin, M. Lafi, T. Kanan, and A. Mansour, “An efficient hybrid similarity measure based on user interests for recommender systems,” Expert Syst., vol. 37, no. 5, p. e12471, Aug. 2019.

[39] R. Garland, “The mid-point on a rating scale: Is it desirable,” Marketing Bull., vol. 2, pp. 66–70, 1991.

[40] K. Yu, X. Xu, J. Tao, M. Ester, and H.-P. Kriegel, “Instance selection techniques for memory-based collaborative filtering,” in Proc. SIAM Int. Conf. Data Mining, Apr. 2002, pp. 59–74.

[41] A. S. Das, M. Datar, A. Garg, and S. Rajaram, “Google news personalization: Scalable online collaborative filtering,” in Proc. 16th Int. Conf. World Wide Web (WWW), New York, NY, USA: Association for Computing Machinery, 2007, pp. 271–280.