IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 26, 2017, accepted November 26, 2017, date of publication November 29, 2017,
date of current version December 22, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2778424

An Efficient Similarity Measure for User-Based Collaborative Filtering Recommender Systems Inspired by the Physical Resonance Principle

ZHENHUA TAN[1,2], (Member, IEEE), AND LIANGLIANG HE[1]
[1]Software College, Northeastern University, Shenyang 110819, China
[2]Academy of Information Technology, Northeastern University, Shenyang 110819, China

Corresponding author: Zhenhua Tan (tanzh@mail.neu.edu.cn).

This work was supported in part by the National Science Foundation of China under Grant 61772125 and Grant 61402097, in part by the Fundamental Research Funds for the Central Universities of China under Grant N151708005, and in part by the Natural Science Foundation of Liaoning Province of China under Grant 201602261.

ABSTRACT User-based collaborative filtering is an important technique used in collaborative filtering recommender systems to recommend items based on the opinions of like-minded nearby users, where similarity computation is the critical component. Traditional similarity measures, such as Pearson's correlation coefficient and cosine Similarity, mainly focus on the directions of co-related rating vectors and have inherent limitations for recommendations. In addition, CF-based recommendation systems always suffer from the cold-start problem, where users do not have enough co-related ratings for prediction. To address these problems, we propose a novel similarity measure inspired by a physical resonance phenomenon, named resonance similarity (RES). We fully consider different personalized situations in RES by mathematically modeling the consistency of users' rating behaviors, the distances between the users' opinions, and the Jaccard factor with both the co-related and non-related ratings. RES is a cumulative sum of the arithmetic product of these three parts and is optimized using learning parameters from data sets. Results evaluated on six real data sets show that RES is robust against the observed problems and has superior predictive accuracy compared with the state-of-the-art similarity measures on full users', grouped users', and cold-start users' evaluations.

INDEX TERMS User-based collaborative filtering, recommender system, similarity measure, RES.

I. INTRODUCTION

Recommender systems emerged in the 1990s [1], [2], aiming to provide personalized recommendations to users by predicting specific items or user preferences. With the rapid development of the Internet, recommender technologies have been applied to a variety of Internet-based systems, such as online videos, online shopping, and online social networks. Recommendation techniques mainly include content-based recommendation, collaborative filtering (CF) recommendation and hybrid recommendation. Content-based recommendation focuses on user profiles and preferences, while CF recommendation focuses on user ratings related to the user-item matrix to find a set of like-minded users or similar items, and hybrid recommendation combines two or more recommendation techniques [3]–[6]. Collaborative filtering techniques are more frequently implemented and often result in better predictive accuracy [7], [8]. These techniques recommend items based on the opinions of other like-minded users or identify items that are similar to those previously rated by the target user, and mainly include item-based CF, which associates an item with nearest neighbors, and user-based CF, which associates a set of nearest neighbors with each user [3], [9]–[11]. This paper focuses on the latter. User-based collaborative filtering usually calculates similarities between users to find nearest neighbors of the user according to users' past preferences [12]–[16].

The similarity measure is a critical component of user-based CF recommendation for computing similarities between users' past behaviors [12]–[14]. It provides a direct recommendation pattern since users with similar past preferences have similar opinions on items. Similarity is usually calculated based on a user-item rating matrix, where each row is a rating vector evaluated by a related user and each column includes ratings of a specific item given by users. The similarity measure models that are commonly used for CF recommender systems are mainly Cosine Similarity (COS) and Pearson's Correlation Coefficient (PCC) [17]–[19]. PCC calculates similarity as the covariance of two users' preferences (ratings) divided by their standard deviations based on co-related items. The formula is:

$$PCC(u,v) = \frac{\sum_{i \in I_{u,v}} (r_{u,i} - \bar{r}_u) \cdot (r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{u,v}} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{u,v}} (r_{v,i} - \bar{r}_v)^2}}, \tag{1}$$

where $I_{u,v}$ is the set of co-related items of both users u and v, $r_{u,i}$ is the rating of item i by user u, and $\bar{r}_u$ is the average rating of user u for all the correlated items. COS calculates the similarity between two users by measuring the value of the cosine angle between the two vectors of ratings; a smaller angle indicates greater similarity. The COS formula is:

$$COS(u,v) = \frac{\sum_{i \in I_{u,v}} r_{u,i} \cdot r_{v,i}}{\sqrt{\sum_{i \in I_{u,v}} r_{u,i}^2} \sqrt{\sum_{i \in I_{u,v}} r_{v,i}^2}}, \tag{2}$$

where $I_{u,v}$ and $r_{u,i}$ have the same meaning as in equation (1). However, the two traditional similarity measures have inherent limitations since they mainly focus on the directions of the rating vectors but ignore the lengths. Researchers observed the related problems of PCC and COS, and proposed a series of solutions, such as [12]–[15], [17], [19], and [20], and [22]–[28]. Breese *et al.* [17], Schwarz *et al.* [19], Guo *et al.* [13], and Ahn [14] analyzed the limitations of traditional PCC and COS. Lathia *et al.* [18] and Ma *et al.* [20] focused on analyzing the problems of PCC. Guo *et al.* [13] summarized four related shortcomings of PCC and COS in detail. We conclude that there are problems in the following seven aspects, where problems (3)(4)(5)(6) are also discussed in [13]:

(1) **Equal-ratio Problem**. While two rating vectors $\left(\vec{X}_{\mathrm{u}}, \vec{X}_{\mathrm{v}}\right)$ given by user pair (u, v) are of equal ratio, such as $\vec{X}_{\mathrm{u}} = \alpha \cdot \vec{X}_{\mathrm{v}}$, the COS or PCC will return '1.0' according to their computational formulas. For example, suppose there are two user pairs $(\mathrm{u}_1, \mathrm{v}_1)$ and $(\mathrm{u}_1, \mathrm{v}_2)$, with related rating vector pairs ([1, 2, 1], [1.5, 3, 1.5]) and ([1, 2, 1], [2.5, 5, 2.5]). Although the two pairs are quite different, the similarities returned by PCC and COS will be both '1.0', without any distinct space.

(2) **Unequal-length Problem**. Suppose the rating vectors for user pair (u_1, v_1) are [1, 2] and [1, 2], where there are only two co-related items. Meanwhile, suppose the rating vectors for pair (u_1, v_2) are [1, 2, 4, 5, 5, 5, 5, 5] and [1, 2, 5, 5, 5, 5, 5, 4], where the number of co-related items is eight. PCC and COS will give 1.0 as the similarity of pair (u_1, v_1), whereas the similarity of pair (u_1, v_2) will be less than 1.0. In this situation, user pairs with more co-related items (or more co-related history) may encounter unfair similarity evaluation by PCC or COS.

(3) **Flat-value Problem**. In the special case of the equal-ratio problem where all rating vectors are flat-valued, such as [1, 1, 1], [3, 3, 3], [5, 5, 5], PCC will be not computable and COS will always return 1.0.

(4) **Single-value Problem**. Suppose there is only one item co-related to two users (u_1, v_1). PCC will be not computable and COS will always return 1.0, regardless of the value of the co-related rating. For example, if the co-related rating vectors given by pair (u_1, v_1) are [1] and [5], respectively, the COS similarity between (u_1, v_1) will be 1.0.

(5) **Opposite-value Problem**. Suppose the two rating vectors have completely opposite values, such as [1, 2, 3] and [3, 2, 1]. PCC will always be -1.0.

(6) **Cross-value Problem**. If there are only two co-related items to two users, PCC will return -1.0 when the two rating vectors cross each other in values, such as [1, 5] and [5, 4], or [1, 3] and [2, 1]. When the two ratings vectors don't cross in values, such as [1, 5] and [4, 5], PCC will be 1.0.

(7) **Non-related Ignored Item Problem**. Traditional similarity measures are based on co-related items, and non-related items of users are ignored. Here, 'non-related items' means items not co-related by both users. Some examples of this problem are discussed in section IV.

Problems (1)(2)(3)(4)(7) occur for both PCC and COS, while problems (5)(6) are limitations of PCC. In addition, CF-based recommendation systems always suffer from the cold-start problem, where users do not have enough co-related ratings for prediction. Some researchers try to improve the traditional PCC or COS to obtain higher predictive accuracy of CF recommendations, such as [7], [17], [19], [20], [22], [24], but these models can't overcome the natural limitations of PCC and COS. Other researchers consider additional hidden factors of rating behaviors and propose novel similarity measures to improve the prediction performance, such as [12]–[15], [23], and [25]–[31].

To address the observed problems of traditional similarity measures, we aim to discover a new similarity computation method for user-based CF recommendation from a physical resonance phenomenon, to reduce the influences of those problems. As is well known, two simple harmonic motions with more similar directions in a physical resonance system are more consistent and will have higher resonance amplitude. Inspired by this, we propose a novel similarity measure named RES, short for *resonance similarity*, consisting of a *consistency* component, *distance* factor and *Jaccard* factor. Users' rating behaviors in a CF recommender system are regarded as simple harmonic vibration motions in a resonance system, and the critical *consistency* component in RES is the measurement of two users' rating consistency by modeling users' initial phase angles in a virtual resonance system that is similar to the simple harmonic vibration system, where the closer the users' initial phase angles are, the higher their similarity is. Two factors are considered to weight the *consistency* component: one is the *distance* factor, which reflects the similarity of the users' opinions on the same items, and the other is the *Jaccard* factor, which weights the *consistency* component by both co-related ratings and *non-related* ratings, to make the RES similarity measure more reasonable. Results based on six real datasets show that RES outperforms PCC, COS and other related state-of-the-art similarity measures. This work makes the following main contributions.

(1) To the best of our knowledge, this is the first time that a similarity measure based on the physical resonance principle is proposed, and it represents a new research attempt on similarity in CF recommender systems. We fully consider different personalized situations in RES by mathematically modeling the *consistency* of users' rating behaviors, *distance* of users' opinions, and *Jaccard* factor with both co-related and non-related ratings.

(2) The proposed RES uses a cumulative sum method to calculate the arithmetic product of *consistency*, *distance* and *Jaccard*. This method is effective and robust against the existing problems, such as the Equal-ratio Problem, Unequal-length Problem, and other problems that PCC and COS encountered.

(3) Datasets of CF recommender systems are usually very sparse. To alleviate the influence of the sparsity of each dataset, we propose parameters for each part of RES, and these parameters are learned from real datasets to optimize the predictive accuracy. Results evaluated on six real datasets show that RES has superior predictive accuracy compared to state-of-the-art similarity measures and is also efficient against the cold-start problem.

The rest of this article is organized as follows. We introduce the related work in Section II and RES similarity in Section III. Analysis by examples for RES is described in Section IV, and experiments and evaluation are described in Section V, followed by conclusions in Section VI.

II. RELATED WORK

Collaborative filtering (CF) techniques play a significant role in recommender systems and are mainly classified into user-based CF and item-based CF. User-based CF discovers users with similar interest or preferences on items to a given user based on users' similarities, while item-based CF usually recommends similar items to a user based on items' similarities.

The core technique of CF is to find a set of similar users or items, and the similarity measure is the critical component. Traditional similarity measures, such as PCC and COS, have been applied in CF for decades. Since there are inherent problems in PCC and COS [7], [12]–[14], [18], researchers have proposed many improved or new similarity models.

Some researchers focused on increasing the predictive accuracy by improving traditional PCC or COS. Candillier *et al.* [7] analyzed the limitations suffered by PCC and COS, and utilized Jaccard similarity to weight traditional similarity measures to benefit from their complementarity. Breese *et al.* [17] adopted the inverse user frequencies as weights to restrict the contributions of popular items in the PCC computation, and proposed an extensive set of experiments regarding the predictive performances of statistical algorithms for CF or recommender systems. Mykhaylo *et al.* [19] demonstrated that the COS similarity measure and PCC measure give incorrect results when predicting recommendations, and observed that the inverse Euclidean distance was more suitable as the actual similarity between the rating vectors. Ma *et al.* [20] improved the PCC by adding a parameter to overcome the potential decrease of accuracy. Said *et al.* [22] observed that PCC and COS were commonly computed without taking the popularity of the set of the two users' co-related items into consideration, investigated the effects of common weighting schemes on different types of users, and showed that different weighting schemes had different effects on the predictive performance. Aygün and Okyay [24] proposed the age-parameterized Pearson similarity by adding an age-based time parameter to PCC. With the new parameter, the improved PCC was more accurate than the traditional one. However, the above improved similarity measures can't change the inherent problems of PCC or COS, which are described in Section I.

Other researchers proposed similarity measures by considering additional hidden factors from rating behaviors. Ahn [14] discussed the problems of traditional similarity measures in CF and proposed a heuristic similarity measure called PIP based on the minute meanings of co-ratings. PIP had three semantic heuristics, namely, Proximity, Impact and Popularity, and was developed by utilizing domain-specific interpretations of user ratings of products to overcome the problems of traditional similarity and distance measures under new-user cold-start conditions. However, PIP didn't consider the influences of non-related items. Guo *et al.* [13] proposed a novel Bayesian similarity measure based on the Dirichlet distribution that considers both the directions and lengths of the rating vectors. The algorithm also improved the rating semantics by bounding the three parts of PIP within 0∼1. This similarity measure can overcome the limitations of PCC and COS, and has excellent predictive accuracy. Bobadilla *et al.* [23] proposed a singularity measure by using hidden attributes in the CF processes to obtain higher predictive accuracy. However, the performance of the singularity measure is limited when there are fewer singularities of co-related items. Laveti *et al.* [15] proposed a weighted ensemble hybrid similarity metric model by combining two or more traditional similarity metrics. Bell *et al.* [25] proposed an approach for predicting user ratings of items by integrating complementary models that focus on patterns at different scales. The proposed approach combined a global component, regional component, and local component, where each modeled the data at a different scale to estimate the unknown ratings. Farnaz *et al.* [26] presented a concept to show that all data in recommender systems are important. Huang and Dai [27] proposed a distance similarity measure weighted by nearest neighbors for each target item that takes the proportion of co-related ratings into account. They obtained improved accuracy on the Movielens Datasets. Reshma *et al.* [28] proposed an approach to improve the predictive accuracy and reduced the influences of the sparsity problem and cold-start problem in recommender systems by finding nearest neighbors from rating patterns and social behaviors.

Incorporating human factors in recommendations is also a very important method which can significantly improve the recommendation accuracy. Lekakos and Giaglis [29] proposed a *lifestyle* approach to improve the prediction accuracy by efficiently managing the problem of limited data availability. Utilizing this *lifestyle* approach, they proposed a hybrid recommendation mechanism to make predictions on available ratings for the items unobserved by the target user, populating related rating vector [30], to address the most important drawback in CF algorithms, such as sparsity problem, cold start problem and new user problem. Winoto and Tang [31] studied the relationships between users' personalized rating behaviors and users' mood states, and proposed an efficient mood-aware recommendation mechanism, by importing user mood factors into traditional recommendation techniques.

Different from the above models, this paper borrows the concept of physical resonance principles to propose a novel similarity measure. We consider the personalization of user rating behaviors in each part of the proposed RES, including *consistency*, *distance* and *Jaccard*. Results show superior efficiency.

III. PROPOSED SIMILARITY MEASURE: RES

To address the problems described in the first section, we introduce a novel similarity measure for CF systems in this section. The core component of this measure, namely, *consistency*, is inspired by the physical resonance of simple harmonic motion; thus, we name this new similarity resonance similarity (***RES***). In our opinion, the rating behavior of users in a specific user-based CF system is similar to the simple harmonic motion of particles in a specific vibration system, in which the resonance amplitude of two particles will be larger when the initial phase angles of the two particles are closer. Thus, we measure two users' rating consistency by modeling the users' initial phase angles in a virtual resonance system that is similar to the simple harmonic vibration system, in which the closer the users' initial phase angles are, the larger their similarity is. The *consistency* component of *RES* can be calculated for any pair of users, even in a *cold-start* situation. Moreover, we consider the *distance* factor to measure the differences between two users' ratings relative to the average rating of a specific co-related item. The *distance* factor reflects the similarity of users' opinions on the same items. Moreover, the number of *non-related ratings*, which are not co-related by user pairs, is very important for similarity measurement, as well as that of co-related ratings, so we import a *Jaccard* factor to weight the *RES* to make

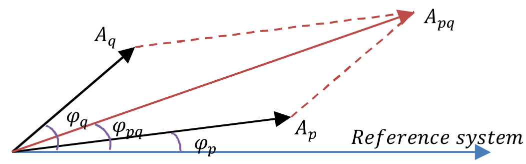


FIGURE 1. Resonance superposition of simple harmonic motion.

the similarity measure more rational. The proposed similarity *RES* can be expressed as:

$$RES(u,v) = \sum\nolimits_{I_{u,v}} C(u,v,k_1) * D(u,v,k_2,k_3) * J(u,v,k_4), \tag{3}$$

where $C(u,v,k_1)$ is the *consistency* component, $D(u,v,k_2,k_3)$ is the *distance* factor, and $J(u,v,k_4)$ is the *Jaccard* factor. Parameters $k_1 \sim k_4$ are learned from the real dataset to make the results of related measures sharper. To bound *RES* within $[0, 1)$, we propose an adjustment using the *arctan* function:

$$sim_{RES}(u,v) = \frac{arctan\left(RES(u,v)\right)}{0.5\pi}. \tag{4}$$

The details of *RES* are described in the following subsections.

A. CONSISTENCY COMPONENT IN RES

First, let's introduce the physical resonance of simple harmonic motion, in which at least two particles with simple harmonic motion oscillate with superposed amplitude at a specific frequency. Suppose two particles, named p and q, are in a simple harmonic vibrational motion system with the same angular frequency, where $p(t) = A_p sin\left(\omega t + \varphi_p\right)$ and $q(t) = A_q sin(\omega t + \varphi_q)$. The superposition equation of p and q is $S_{pq}(t) = p(t) + q(t) = A_{pq} sin\left(\omega t + \varphi_{pq}\right)$, where $\varphi_{pq} = arctan\left(\frac{A_p \sin\varphi_p + A_q \sin\varphi_q}{A_p \cos\varphi_p + A_q \cos\varphi_q}\right)$. Thus, the resonance amplitude A_{pq} is:

$$A_{pq} = \sqrt{A_p^2 + A_q^2 + 2A_pA_q cos\left(\varphi_q - \varphi_p\right)}, \tag{5}$$

where $\left|A_p - A_q\right| \leq A_{pq} \leq A_p + A_q$; A_{pq} will attain its maximum value when $\varphi_q - \varphi_p = 2k\pi$ and its minimum value when $\varphi_q - \varphi_p = (2k+1)\pi$. Figure 1 shows the resonance superposition of $p(t)$ and $q(t)$.

The final resonance amplitude A_{pq} is determined by the difference of the two related initial phase angles, and closer φ_q and φ_p are, the larger A_{pq} is. In other words, in a resonance system, simple harmonic motions with closer directions are more consistent and result in higher resonance amplitude, and the value of the resonance amplitude A_{pq} indicates the degree of consistency between φ_q and φ_p.

This phenomenon inspires us to think about the consistency of users' rating vectors. Two users with similar rating directions in a user-based CF system may have similar past preferences. Thus, we consider calculating the *consistency* of users' rating behaviors similar to the physical resonance amplitude. Assume all users are in the same simple harmonic vibration motion system: $X_u(t) = sin(\omega t + \varphi_u)$ for all u. The key task is to model each user's rating behavior as a related initial phase angle φ_u. Following two aspects are considered to measure the rating behavior for a given user u:

- The first aspect is the basic distance between user's rating and the median rating, which is defined as the *base of the user's initial phase angle*. Let R_{max} and R_{min} denote the maximum and minimum ratings in the rating scale, respectively, and Let $R_{med} = ((R_{max} - R_{min})/2)$ denote the median of the rating scale. We take $\left(r_{u,i} - R_{med}\right)$ to be the *base of the user's initial phase angle*, which is denoted as φ_u^{base}.

$$\varphi_u^{base} = \left(r_{u,i} - R_{med}\right) \tag{6}$$

- The second aspect is the distance between user's rating and the average rating for a specific item, which is named the *personalization of the user's rating habit*. In real recommender systems, users have different rating preferences. Some give relatively higher ratings for items than others, while some have very strict rating criteria for the same items. Usually, we use the average rating $\bar{r}_u$ to represent the rating personalization of u. Let μ be the overall average rating for all items. The personalized distance between $\bar{r}_u$ and μ is defined as $(\bar{r}_u - \mu)$.

To discover additional details to design a more rational initial phase angle φ_u for user u, we define a condition for discussion as:

$$condition = \left(r_{u,i} - R_{med}\right) \times (\bar{r}_u - \mu). \tag{7}$$

(1) When $condition > 0$, $\left(r_{u,i} - R_{med}\right)$ and $(\bar{r}_u - \mu)$ are of the same sign. We assume $\left(r_{u,i} - R_{med}\right) > 0$ under this condition, which means φ_u^{base} is a positive number. In this case, $(\bar{r}_u - \mu) > 0$, which means the user has less strict rating criteria than the average user, such as user a in Figure 2, so we need to rationally reduce φ_u^{base} to obtain a more rational initial angle for such users. If we assume $\left(r_{u,i} - R_{med}\right) < 0$ and $(\bar{r}_u - \mu) < 0$ under this condition, φ_u^{base} is a negative number and the user tends to give lower ratings to items, such as user c in Figure 2, so we also need to rationally reduce φ_u^{base}. Let φ_u^{+} denote the initial phase angle of user u under $condition \geq 0$. We use the reciprocal of the arithmetic difference between $\bar{r}_u$ and μ as the main reduction factor of φ_u^{base}, and:

$$\varphi_u^{+} = \left(\frac{1}{1 + |\bar{r}_u - \mu|}\right) \cdot \varphi_u^{base}. \tag{8}$$

(2) When $condition < 0$, $\left(r_{u,i} - R_{med}\right)$ and $(\bar{r}_u - \mu)$ are of opposite sign. When $\left(r_{u,i} - R_{med}\right) > 0$ and $(\bar{r}_u - \mu) < 0$, φ_u^{base} is positive and the user has stricter rating behaviors, such as user b in Figure 2, so we need to rationally increment φ_u^{base}. In addition, when $\left(r_{u,i} - R_{med}\right) < 0$ and $(\bar{r}_u - \mu) > 0$, φ_u^{base} is negative and the user prefers to give higher ratings to items, such as user d in Figure 2, so we also need to increase the initial phase angle based on φ_u^{base}. Therefore, let φ_u^{-} denote the initial phase angle of user u under $condition < 0$.

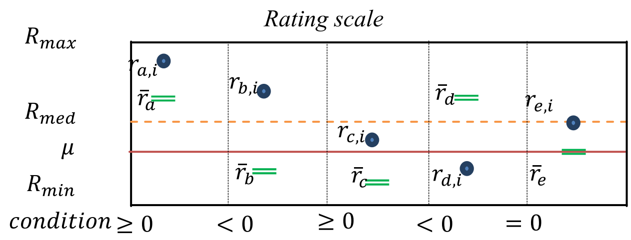


FIGURE 2. Demonstrations of personalization of user rating behaviors.

We use the arithmetic difference between $\bar{r}_u$ and μ as the main increment factor of φ_u^{base}, and:

$$\varphi_u^{-} = \left(1 + \frac{|\bar{r}_u - \mu|}{(R_{max} - R_{min})/2}\right) \cdot \varphi_u^{base}. \tag{9}$$

Here, we constrain the range as $\varphi_u^{-} \in [R_{min} - \mu, R_{max} + \mu]$.

(3) When $condition = 0$, such as for user e in Figure 2, $\varphi_u^{+} = \varphi_u^{-}$ according to the above two equations.

Therefore, based on the above discussions, we design the initial phase angle $\varphi(u)$ and map $\varphi(u)$ into the range $[0, \pi]$. The final proposed $\varphi(u)$ is defined as:

$$\varphi(u) = \begin{cases} \dfrac{\pi}{R_{max} - R_{min}} \cdot \varphi_u^{+} & when \ \left(r_{u,i} - R_{med}\right)(\bar{r}_u - \mu) \geq 0 \\ \dfrac{\pi}{R_{max} - R_{min}} \cdot \varphi_u^{-} & when \ \left(r_{u,i} - R_{med}\right)(\bar{r}_u - \mu) < 0. \end{cases} \tag{10}$$

Finally, based on the resonance amplitude equation, we calculate the proposed *consistency* measure of user pair (u, v) by $\sqrt{A_u^2 + A_v^2 + 2A_uA_v \cos(\varphi(u) - \varphi(v))}$, and we set the amplitudes A_u and A_v equal to 0.5. Therefore, the *consistency* measure in RES is defined as:

$$C(u, v, k_1) = \left(\sqrt{0.5 + 0.5\cos(\varphi(u) - \varphi(v))}\right)^{k_1}, \tag{11}$$

where $k_1 (> 0)$ is used to make the result sharper, and $C(u, v, k_1) \in [0, 1]$. This parameter is learned from the real dataset, and the learning method is described in the last part of this section.

B. DISTANCE FACTOR IN RES

According to the *consistency* measure equations (10) and (11), the arithmetic difference of $(\varphi(u_1) - \varphi(v_1))$ mainly depends on $\left(r_{u_1,i} - r_{v_1,i}\right)$. When $(\varphi(u_1) - \varphi(v_1))$ is equal to $(\varphi(u_2) - \varphi(v_2))$, the *consistency* of pair (u_1, v_1) will be equal to that of pair (u_2, v_2). However, we observed that ratings given by users for a co-related item are usually different from the average rating of the item, as illustrated in Figure 3. When distances between user pairs remain the same, the similarities are quite different for different arithmetic deviations of ratings from $\bar{r}_i$ of a specific co-related item. Therefore, we design a *distance* factor to complement the *consistency* component in RES, to obtain a more rational similarity measure. In this paper, the *distance* factor reflects the similarity of users' opinions on the same items, and it is proposed mainly

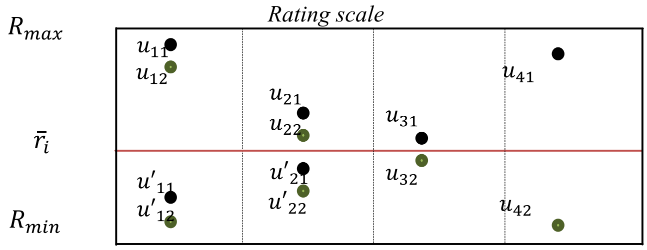


FIGURE 3. **Demonstrations of rating distances.**

based on the following two situations, namely, situation 1 and situation 2.

In **Situation 1**, the ratings given by user pair (u, v) are both above (or both below) the average rating of a specific co-related item: $\left(r_{u,i} - \bar{r}_i\right)\left(r_{v,i} - \bar{r}_i\right) \geq 0$. In this situation, we define the *distance* factor from two aspects. The first is the arithmetic difference between $r_{u,i}$ and $r_{u,i}$ given by the related two users; larger $\left|r_{u,i} - r_{v,i}\right|$ results in lower similarity. Therefore, we propose a decreasing exponential function $d_1^+(u, v)$ to calculate this influence factor:

$$d_1^+(u, v) = e^{-\left|r_{u,i} - r_{v,i}\right|}. \tag{12}$$

Here, $d_1^+(u, v)$ represents the influence of distance differences of pair (u, v) when $\left(r_{u,i} - \bar{r}_i\right)\left(r_{v,i} - \bar{r}_i\right) \geq 0$.

The second aspect is the average distance from $\bar{r}_i$ of pair (u, v), i.e., the average of $\left|r_{u,i} - \bar{r}_i\right|$ and $\left|r_{v,i} - \bar{r}_i\right|$. A larger average distance from $\bar{r}_i$ implies that the users have more distinct opinions; thus, their similarity is greater. Let's take pair (u_{11}, u_{12}) and pair (u_{21}, u_{22}) in Figure 3 for example, where their ratings are all above the average rating $\bar{r}_i$. Although the rating distance $d_1^+(u_{11}, u_{12})$ is close to $d_1^+(u_{21}, u_{22})$, the ratings given by user pair (u_{11}, u_{12}) are higher than those given by pair (u_{21}, u_{22}). It indicates pair (u_{11}, u_{12}) has more distinct opinions on the same item than pair (u_{21}, u_{22}). So pair (u_{11}, u_{12}) should be motivated by a larger similarity factor than pair (u_{21}, u_{22}) in RES in this aspect. Therefore, we design an increasing exponential function $d_2^+(u, v)$ to calculate this influence factor:

$$d_2^+(u, v) = e^{\frac{1}{2}\times\left(\left|r_{u,i} - \bar{r}_i\right| + \left|r_{v,i} - \bar{r}_i\right|\right)}, \tag{13}$$

where $d_2^+(u, v)$ denotes the influence of the average distance when $\left(r_{u,i} - \bar{r}_i\right)\left(r_{v,i} - \bar{r}_i\right) \geq 0$. Based on $d_1^+(u, v)$ and $d_2^+(u, v)$, we define the *distance* factor calculation equation under situation 1 as:

$$D^+(u, v, k_2) = \left(d_1^+(u, v) \times d_2^+(u, v)\right)^{k_2}, \tag{14}$$

where k_2 (> 0) is a parameter learned from the real dataset to make the result sharper.

In **Situation 2**, the ratings given by two users are located on different sides relative to the average rating of a specific co-related item: $\left(r_{u,i} - \bar{r}_i\right)\left(r_{v,i} - \bar{r}_i\right) < 0$, such as user pairs (u_{31}, u_{32}) and (u_{41}, u_{42}) in Figure 3. In this situation, one user's rating is bigger than $\bar{r}_i$ and the other's is less than $\bar{r}_i$. This means the two users have different opinions about the co-related item. Thus, we use the distance between the two users' ratings to calculate the *distance* factor in this situation; a larger distance between users' ratings implies a smaller similarity factor between users. For example, pair (u_{31}, u_{32}) and pair (u_{41}, u_{42}) have different opinions on a specific item; however, the rating distance of pair (u_{31}, u_{32}) is less than that of pair (u_{41}, u_{42}). Thus, pair (u_{31}, u_{32}) has a larger similarity factor than pair (u_{41}, u_{42}). We design a decreasing exponential function $d_1^-(u, v)$ to calculate this influence, similar to $d_1^+(u, v)$:

$$d_1^-(u, v) = e^{-\left|r_{u,i} - r_{v,i}\right|}. \tag{15}$$

Therefore, the distance factor is based on $d_1^-(u, v)$ when $\left(r_{u,i} - \bar{r}_i\right)\left(r_{v,i} - \bar{r}_i\right) < 0$, and the related equation for situation 2 is proposed as:

$$D^-(u, v, k_3) = \left(d_1^-(u, v)\right)^{k_3}, \tag{16}$$

where k_3 (> 0) is also the parameter learned from the real dataset to make the result sharper.

Based on $D^+(u, v, k_2)$ and $D^-(u, v, k_3)$, the final proposed *distance* factor in RES is described as:

$$D(u, v, k_2, k_3) = \begin{cases} D^+(u, v, k_2) & when\ \left(r_{u,i} - \bar{r}_i\right)\left(r_{v,i} - \bar{r}_i\right) \geq 0 \\ D^-(u, v, k_3) & when\ \left(r_{u,i} - \bar{r}_i\right)\left(r_{v,i} - \bar{r}_i\right) < 0. \end{cases} \tag{17}$$

C. JACCARD FACTOR IN RES

Traditional similarity measures usually depend on the co-related ratings, and the length of the co-related ratings may have drawbacks for the similarity calculation. In this paper, the non-related ratings between users also play an important role in the similarity measure. We take three user pairs for examples, marked as (u_1, v_1), (u_2, v_2), and (u_3, v_3). Their scales of ratings and scales of co-related ratings are as follows: (E1) For pair (u_1, v_1), $\left|I_{u_1}\right| = 15$, $\left|I_{v_1}\right| = 50$ and $\left|I_{u_1,v_1}\right| = 2$. (E2) For pair (u_2, v_2), $\left|I_{u_2}\right| = 15$, $\left|I_{v_2}\right| = 50$ and $\left|I_{u_2,v_2}\right| = 15$. (E3) For pair (u_3, v_3), $\left|I_{u_3}\right| = 150$, $\left|I_{v_3}\right| = 500$ and $\left|I_{u_3,v_3}\right| = 15$. In the first example (E1), based on traditional similarity measures such as PCC and COS, the similarity of pair (u_1, v_1) is occasionally very high since their co-related ratings are too small (e.g., the two rating vectors for co-related items are similar). In examples (E2) and (E3), the scales of the co-related ratings of pair (u_2, v_2) and pair (u_3, v_3) are the same, but the scales of non-related ratings are quite different. Traditional similarities for the two pairs are very close if the ratings for co-related items given by the two pairs are relatively similar. However, the similarity of pair (u_2, v_2) is higher than that of pair (u_3, v_3) if co-related ratings are similar, because pair (u_2, v_2) has fewer non-related ratings than pair (u_3, v_3). Therefore, we define the number of non-related ratings as a complementary factor for similarity, in addition to the number of co-related ratings, by using the Jaccard coefficient. Jaccard calculates the correlation between users by measuring the overlap of the two vectors as $Jaccard(u, v) = \frac{\left|I_{u,v}\right|}{\left|I_u \bigcup I_v\right|}$, where $\left|I_{u,v}\right|$ denotes the number of co-related items of both users and $\left|I_u \bigcup I_v\right|$ includes both non-related and co-related ratings. We use *Jaccard* as a weight for the proposed similarity as:

$$J\left(u, v, k_4\right)=\left(\frac{\left|I_{u,v}\right|}{\left|I_u\right|+\left|I_v\right|-\left|I_{u,v}\right|}\right)^{k_4}, \tag{18}$$

where $k_4\ (>0)$ is learned from the real dataset to make the result sharper.

D. ALGORITHM DESCRIPTION

In this subsection, we describe the algorithm to compute the RES similarity in Algorithm 1. We mainly present the pseudo-code related to the proposed equation (3). If readers need to bound RES to 0∼1, please refer to the adjusted RES, as described in equation (4). Parameters mentioned in Algorithm (1) are described in the next subsection. Once the parameters are confirmed, the computational complexities of *consistency*, *distance* and *Jaccard* are a constant $O(1)$, and the general complexity of RES is $O(\left|I_{u,v}\right| \cdot O(1))$.

Algorithm 1 The Computation of Resonance Similarity RES(u, v)

Parameters. $\mathbb{K}=[k_1, k_2, k_3, k_4]$.
Input. From Dataset: ratings r_u and r_v of users u and v, rated items I_u and I_v, average ratings $\bar{r}_u$ and $\bar{r}_v$, item i's average rating $\bar{r}_i$, users' overall average rating μ, R_{max}, R_{min}.
Output. Resonance Similarity RES(u, v) for user pair (u, v).

(1) $R_{med} \leftarrow (R_{max}+R_{min})/2$;
(2) RES$(u, v) \leftarrow 0$;
(3) **For each** $i \in I_u \cap I_v$ **do**
(4) Obtain rating pair $\left(r_{u,i}, r_{v,i}\right)$ of user pair (u, v);
(5) Compute the initial phase $\varphi(u)$ by Eq. (10) for user u;
(6) Compute the initial phase $\varphi(v)$ by Eq. (10) for user v;
(7) Compute the *consistency* measure $C(u, v, k_1)$ for pair (u, v) by Eq. (11);
(8) Compute the *distance* factor $D(u, v, k_2, k_3)$ for pair (u, v) by Eq. (17);
(9) Compute the *Jaccard* factor $J(u, v, k_4)$ for pair (u, v) by Eq. (18);
(10) RES$(u, v) \leftarrow$ (RES$(u, v)+C(u, v, k_1)*D(u, v, k_2, k_3)*J(u, v, k_4)$)
(11) **End**
(12) **Return** RES(u, v)

E. OPTIMIZING PERFORMANCE

As described in previous, we have considered the personalization of users' rating behaviors in *consistency* component, *distance* factor and *Jaccard* factor, and the RES is a cumulative sum of the product of $C(u, v, k_1) * D(u, v, k_2, k_3) * J(u, v, k_4)$, where $k_1 \sim k_4$ are exponent parameters according to equations (11) ∼ (18). However, datasets of different recommender systems are also personalized, and different datasets are quite different in rating scales, the number of ratings, data densities and etc. This general personalization depends on the dataset itself and should be learned from the given datasets. With such a consideration, we utilize four exponent parameters $k_1 \sim k_4$ to learn the personalization of the given dataset, and the main purpose is to get higher predictive accuracy by obtaining optimized personalized exponential weights for $C(u, v, k_1)$, $D(u, v, k_2, k_3)$ and $J(u, v, k_4)$.

In this paper, parameters are learned one by one according to the sequence $k_1 \rightarrow k_2 \rightarrow k_3 \rightarrow k_4$. We learn the parameter values sequentially by applying stratified multi-objective optimization method iteratively, with the objective of obtaining the minimum MAE by $RES(u, v)$ when giving $k_1 \sim k_4$ to *RES*. The prediction of a user rating of an item is usually based on the ratings of nearest neighbors in a user-based CF system. Once the similarity has been obtained, the prediction is calculated by the following equation in this paper.

$$p\left(u, i\right)=\bar{r}_u+\frac{\sum_{v \in U(i)}\left(RES(u, v) \cdot\left(r_{vi}-\bar{r}_v\right)\right)}{\sum_{v \in U(i)}|RES(u, v)|}, \tag{19}$$

where $RES(u, v)$ is the proposed similarity RES with input parameter vector $\mathbb{K}=[k_1, k_2, k_3, k_4]$, and $U(i)$ is the set of users related to item i. The Mean Absolute Error rate is used to evaluate the predictive accuracy:

$$\mathrm{MAE}=\frac{\sum_{(\mathrm{u,i}) \in \mathbb{T}}\left|\mathrm{p}\left(\mathrm{u, i}\right)-\mathrm{r_{u,i}}\right|}{|\mathbb{T}|}, \tag{20}$$

where $\mathbb{T}$ is the rating dataset of (user, item) and $|\mathbb{T}|$ is the cardinality of set $\mathbb{T}$. To find the minimum MAE, we first obtain an initialized MAE^* using $\mathbb{K}=[1,0,0,0]$. Then, we optimize the MAE by the following equation (21), as shown at the bottom of the next page.

The $g\left(\mathbb{K}\right)$ is an iterative optimizing function that includes four sub functions in each iteration: $f\left(k_1\right), f\left(k_2\right), f\left(k_3\right)$ and $f\left(k_4\right)$. When the initialized MAE^* and $\mathbb{K}$ are ready, the function $f\left(k_1\right)$ will try to find a new minimum $MAE=f_1^*$ by learning parameter $k_1>0$. When $f\left(k_1\right)$ stops, the parameter vector will be updated to $\mathbb{K}^*=[k_1^*, k_2, k_3, k_4]$ for the next function $f\left(k_2\right)$, and $f\left(k_2\right), f\left(k_3\right)$ and $f\left(k_4\right)$ will perform processes similar to that performed by $f\left(k_1\right)$. The MAE will be updated to a new MAE^* when $f\left(k_4\right)$ stops, and the new optimizing iteration $g\left(\mathbb{K}\right)$ will continue until $\Delta MAE<\varepsilon$, where ε is a small floating-point value (such as 0.0001 in this paper). Figure 4 shows the optimization processes. Experimental details are described in section V.

Obviously, the computational complexity of the above optimization is big to $O(n^4)$. However, for a given dataset, the four exponential parameters are precalculated before they are used directly in similarity calculations and predictions. The recommender system can make a refresh policy to restart the optimization when the recommendation data scale growing on to an extent. Additionally, we believe that the optimization method could also be applied into other similarity

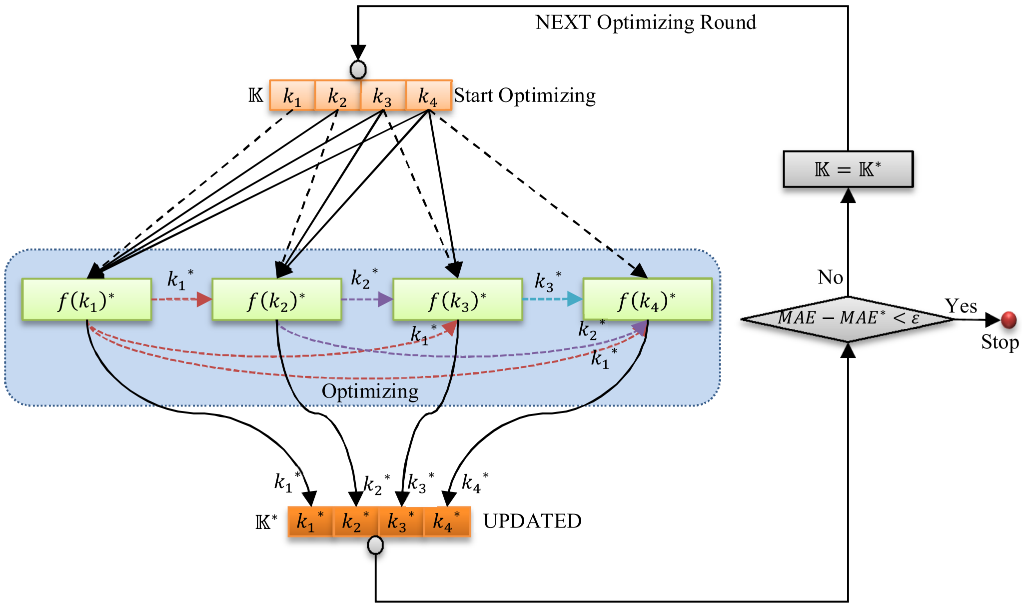


FIGURE 4. Optimization processes for parameters and performance of RES.

measures by adding exponential weights, especially when they consist of several parts.

IV. ANAYSIS BY EXAMPLES

We provide examples of the similarity computation in this section to illustrate the differences among the proposed RES, PCC, COS and PIP. The examples are designed according to the problems described in section I. The rating vectors in the examples have different numbers of ratings; example (5) has the most ratings in related vectors where there are eight co-related items. For all examples, assume the overall average rating of all items is $\mu = 3.1$, and $\bar{r}_{i_1} = 3.5$, $\bar{r}_{i_2} = 2.5$, $\bar{r}_{i_3} = 3.2$, $\bar{r}_{i_4} = 2.1$, $\bar{r}_{i_5} = 3.4$, $\bar{r}_{i_6} = 2.3$, $\bar{r}_{i_7} = 3.8$, and $\bar{r}_{i_8} = 2.8$. Here, $\bar{r}_{i_x}$ is the average rating of the x^{th} co-related item in each example. Then, we compute the similarity of user pair (u, v) in each example, and Table 1 shows the results. For RES, we compute the similarity by both equation (3) and equation (4), named RES_Eq3 and RES_Eq4, respectively, and assume the parameters are fixed as $k_1 = k_2 = k_3 = k_4 = 1$ (in real experiments, parameters are optimized based on the datasets), which means consistency component, distance factor and Jaccard factor have same weights during computation.

First of all, let's take 'example (1)' for instance to illustrate the computation processes of RES. In the example, rating vectors given by user pair (u, v) are [1, 2, 1] and [1.5, 3, 1.5], respectively. Under the assumption, there are $\mu = 3.1$, $\bar{r}_{i_1} = 3.5$, $\bar{r}_{i_2} = 2.5$, and $\bar{r}_{i_3} = 3.2$. Therefore,

- For item i_1, the consistency $C(u, v, 1) = 1.0$ according to equation (11), distance factor $D(u, v, 1, 1) = 5.7545$ according to equation (17), and Jaccard factor $J(u, v, 1) = 1.0$ according to equation (18). So the product of $C(u, v, 1) * D(u, v, 1, 1) * J(u, v, 1) = 5.7545$.
- For item i_2, the consistency $C(u, v, 1) = 0.98$, distance factor $D(u, v, 1, 1) = 0.3679$, and Jaccard factor $J(u, v, 1) = 1.0$, so the product of $C(u, v, 1) * D(u, v, 1, 1) * J(u, v, 1) = 0.3605$.
- For item i_2, the consistency $C(u, v, 1) = 1.0$, distance factor $D(u, v, 1, 1) = 4.2631$, and

$$\min_{\substack{s.t.:\mathbb{K} \\ Stop\ when\ \Delta MAE<\varepsilon}} g(\mathbb{K}) = \begin{cases} f(k_1) = \underset{\substack{s.t.:k_1>0;k_2,k_3,k_4\in\mathbb{K} \\ f_1<MAE^*}}{\arg\min}\, \mathrm{MAE} = \underbrace{f_1^*}_{Update:\ \mathbb{K}^*=[k_1^*,k_2,k_3,k_4]} \\ f(k_2) = \underset{\substack{s.t.:k_2>0;k_1,k_3,k_4\in\mathbb{K}^* \\ f_2<f_1^*}}{\arg\min}\, \mathrm{MAE} = \underbrace{f_2^*}_{Update:\ \mathbb{K}^*=[k_1^*,k_2^*,k_3,k_4]} \\ f(k_3) = \underset{\substack{s.t.:k_3>0;k_1,k_2,k_4\in\mathbb{K}^* \\ f_3<f_2^*}}{\arg\min}\, \mathrm{MAE} = \underbrace{f_3^*}_{Update:\ \mathbb{K}^*=[k_1^*,k_2^*,k_3^*,k_4]} \\ f(k_4) = \underset{\substack{s.t.:k_4>0;k_1,k_2,k_3\in\mathbb{K}^* \\ f_4<f_3^*}}{\arg\min}\, \mathrm{MAE} = \underbrace{f_4^*}_{\substack{Update:\ \mathbb{K}=[k_1^*,k_2^*,k_3^*,k_4^*] \\ MAE=MAE^*}}. \end{cases} \quad (21)$$

TABLE 1. Examples of PCC, COS, PIP and RES similarity measures, against the Equal-ratio Problem, Unequal-length Problem, Flat-value Problem, Opposite-value Problem, Single-value Problem, and Cross-value Problem.

Problem	Examples			PCC	COS	PIP	RES_Eq3	RES_Eq4
	ID	Vector u	Vector v				$\mathbb{K}$ = [1,1,1,1]	$\mathbb{K}$ = [1,1,1,1]
Equal-ratio	1	[1,2,1]	[1.5,3,1.5]	1.0	1.0	6015.4719	10.3781	0.9388
	2	[1,2,1]	[2.5,5,2.5]	1.0	1.0	2029.6406	2.2512	0.7339
	3	[1,2,1,1,4]	[0.5,1,0.5,0.5,2]	1.0	1.0	14913.2172	19.8975	0.9680
Unequal-length	4	[1,2]	[1,2]	1.0	1.0	5690.250	13.8312	0.9541
	5	[1,2,4,5,5,5,5,5]	[1,2,5,5,5,5,5,5]	0.9765	0.9971	28257.870	65.4555	0.9903
Flat-value	6	[1,1,1]	[1,1,1]	NaN	1.0	11911.860	25.6892	0.9752
	7	[1,1,1]	[3,3,3]	NaN	1.0	441.0	1.1164	0.5350
	8	[1,1,1]	[5,5,5]	NaN	1.0	3.0	0.0412	0.0262
Opposite-value	9	[1,5,1]	[5,1,5]	-1.0	0.4042	3.0	0.0125	0.0080
	10	[2,4,2]	[4,2,4]	-1.0	0.8165	75.0	0.2364	0.1478
	11	[2,3,2]	[4,3,4]	-1.0	0.9469	302.0	1.8632	0.6864
Single-value	12	[1]	[1]	NaN	1.0	5285.250	12.1825	0.9479
	13	[1]	[3]	NaN	1.0	147.0	0.5684	0.3291
	14	[1]	[5]	NaN	1.0	1.0	0.0137	0.0087
Cross-value	15	[1,5]	[5,1]	-1.0	0.3846	2.0	0.0002	0.0001
	16	[1,4]	[4,2]	-1.0	0.4706	31.0	0.0757	0.0481
	17	[2,4]	[3,2]	-1.0	0.8682	324.0	0.9626	0.4879
	18	[5,1]	[5,4]	1.0	0.8882	2585.250	3.6253	0.8287
	19	[5,1]	[5,2]	1.0	0.9833	3137.250	5.2020	0.8791
	20	[5,2]	[4,1]	1.0	0.9908	1536.0	1.9971	0.7045

Jaccard factor $J(u, v, 1) = 1.0$, so the product of $C(u, v, 1) * D(u, v, 1, 1) * J(u, v, 1) = 4.2631$.

According to equation (3), RES_Eq3 $= \sum_{I_{u,v}} C(u, v, 1) * D(u, v, 1, 1) * J(u, v, 1) = 5.7545 + 0.3605 + 4.2631 = 10.3781$, and RES_Eq4 $= \frac{arctan(10.3781)}{0.5\pi} = 0.9388$ according to equation (4).

Then, let's analyze the results of the 20 examples as shown in table 1. It is observed that the proposed RES is superior to PCC and COS against the problems.

For the Equal-ratio problem, RES for ([1,2, 1], [1.5, 3, 1.5]) is higher than RES for ([1, 2, 1], [2.5, 5, 2.5]) and ([1, 2, 1, 1, 4], [0.5, 1, 0.5, 0.5, 2]), whereas PCC and COS are always 1.0. This illustrates that RES can differentiate the similarities when the users' rating vectors are of equal ratio. For the Unequal-length problem, RES and PIP can return higher similarities for the user pair with longer length of co-related ratings. However, PCC and COS always return 1.0 for ([1, 2], [1, 2]), and return lower similarity for ([1, 2, 4, 5, 5, 5, 5, 5], [1, 2, 5, 5, 5, 5, 5, 5]). For the Flat-value and Single-value problems, PCC is non-computable and COS is always 1.0, whereas RES_Eq3 and RES_Eq4 return more rational similarities. In examples (6) to (8), the RES_Eq4 returns 0.9752, 0.5350 and 0.0262 for rating vectors ([1, 1, 1], [1, 1, 1]), ([1, 1, 1], [3, 3, 3]), and ([1, 1, 1], [5, 5, 5]), respectively, where RES has obviously distinguished results for these three distinct Flat-value rating vectors. RES_Eq3 and RES_Eq4 also have rational similarity results when measuring ([1], [1]), ([1], [3]) and ([1], [5]) in examples (12) to (14).For the Opposite-value problem, our proposed RES also produces more reasonable results, whereas PCC is always -1.0. Specifically, COS is 0.4042 for rating vectors [1, 5, 1] and [5, 1, 5] in example 9, whereas the proposed RES_Eq3 returns 0.0125 and RES_Eq4 returns 0.0080. Rating vectors [1, 5, 1] and [5, 1, 5] reflect much different opinions of users u and v, so we observed the RES produces more a realistic similarity than either PCC or COS against the Opposite-value problem. For the Cross-value problem, the PCC is either -1.0 or 1.0, whereas RES is more distinguished. COS can also return different results for different examples of Cross-values, but the results are not accurate enough. In example 15, COS is 0.3846 for the very different rating vectors [1, 5] and [5, 1], whereas RES_Eq3 returns 0.0002 and RES_Eq4 returns 0.0001. Intuitively, the two vectors are completely non-similar due to their extremely different opinions on the same items. Thus, our proposed RES is more rational for this Cross-value problem. We also compute the PIP similarities for all examples. Compared with PIP, RES_Eq4 can be bounded within 0∼1, and RES_Eq3 is similar to PIP in the weighted sum method. However, RES_Eq3 can return more accurate and rational results than PIP. In example (8), for rating vectors [1, 1, 1] and [5, 5, 5], PIP returns 3.0 and RES_Eq3 returns 0.0412, which is more satisfactory in terms of users' different personalizations. Similar situations occur in other examples, such as examples (5), (6), (7), (9), (10), (11) and (12). Particularly, in example 15, in which the opinions are extremely different, RES_Eq3 returns 0.0002, whereas PIP is 2.0.

For the Non-related Ignored Item Problem described in section I, we first illustrate an artificial sub-dataset in Table 2 (a), where there are five users and ten items. Assume the overall average rating of all items is $\mu = 3.1$, and assume $\bar{r}_{i_1} = 3.5, \bar{r}_{i_2} = 2.5, \bar{r}_{i_3} = 3.2, \bar{r}_{i_4} = 2.1, \bar{r}_{i_5} = 3.4, \bar{r}_{i_6} = 2.3, \bar{r}_{i_7} = 3.8, \bar{r}_{i_8} = 2.8, \bar{r}_{i_9} = 3.1$, and $\bar{r}_{i_{10}} = 2.2$. We compute the similarity of user pair $(u1, ux)$, as shown in Table 2 (b).

PCC and COS are always 1.0 on the example data, without considering the length of non-related items. For user pair

TABLE 2. Examples of PCC, COS, PIP and RES similarity measures, against the Non-related Problem. (a) Artificial sub dataset. (b) Similarity.

(a)

Users	Items									
	$i1$	$i2$	$i3$	$i4$	$i5$	$i6$	$i7$	$i8$	$i9$	$i10$
$u1$	1	4		5		4		5		5
$u2$	1	4	5	5	3		5		4	
$u3$	1	4		5	4				2	
$u4$	1	4								5
$u5$	1	4	2			4		5		

(b)

User Pairs	Rating Vectors	PCC	COS	PIP	RES_Eq3 $\mathbb{K} = [1,1,1,1]$	RES_Eq4 $\mathbb{K} = [1,1,1,1]$
$(u1, u2)$	[1,4,5],[1,4,5]	1.0	1.0	13198.14	8.0396	0.9212
$(u1, u3)$	[1,4,5],[1,4,5]	1.0	1.0	13198.14	9.5014	0.9332
$(u1, u4)$	[1,4,5],[1,4,5]	1.0	1.0	12782.61	11.0363	0.9425
$(u1, u5)$	[1,4,4,5],[1,4,4,5]	1.0	1.0	11855.97	8.4990	0.9254

(u1, u2), there are three co-related items ($i1$, $i2$, $i4$) and seven non-related items (i3, $i5$, $i6$, $i7$, $i8$, $i9$, $i10$), and RES returns 8.0396 by equation (3) and 0.9212 by equation (4), where both consider the non-related items through the Jaccard factor. Meanwhile, although there are the same three co-related items ($i1$, $i2$, $i4$) for pair ($u1$, $u3$), there are six non-related items ($i5$, $i6$, $i7$, $i8$, $i9$, $i10$), and RES for ($u1$, $u3$) is higher than that for ($u1$, $u2$). For pair ($u1$, $u4$), there are only three non-related items ($i5$, $i6$, $i8$) and the co-related rating vectors remain the same; thus, RES is higher than those above. The same situation occurs for pair ($u1$, $u5$). Of course, PIP shows similar performances in these examples, except for user pairs ($u1$, $u2$) and ($u1$, $u3$), where PIP returns the same value of 13198.14, but RES can distinguish between the two pairs. Therefore, RES has ability to overcome the Non-related Ignored Item Problem.

The above examples prove the efficiency of RES, and prove that RES can overcome the limitations of PCC and COS to some extent. We perform a series of experiments in the next section to obtain additional evidence.

V. EXPERIMENTS AND EVALUATION

In this section, we perform a series of experiments to evaluate the efficiency of the RES measure on six real datasets: Movielens-100K, Movielens-latest-small(100K), Movielens-1M, FilmTrust, MiniFilm and Epinions. The prediction is calculated by following the previous equation (19):

$$p(u, i) = \bar{r}_u + \frac{\sum_{v \in U(i)} (sim(u, v) \cdot (r_{vi} - \bar{r}_v))}{\sum_{v \in U(i)} |sim(u, v)|}, \tag{22}$$

where $sim(u, v)$ is the similarity measure. The evaluation performance of predictive accuracy is measured by the metric Mean Absolute Error (MAE) described in equation (20), and a lower MAE value indicates better predictive accuracy.

We design five experiments to evaluate the performance of RES. (1) The first experiment is to optimize MAE by learning parameter vector $\mathbb{K}$ from the full ratings of all users in the experimental datasets for RES, and the optimized results will be used for comparison with other similarity measures.

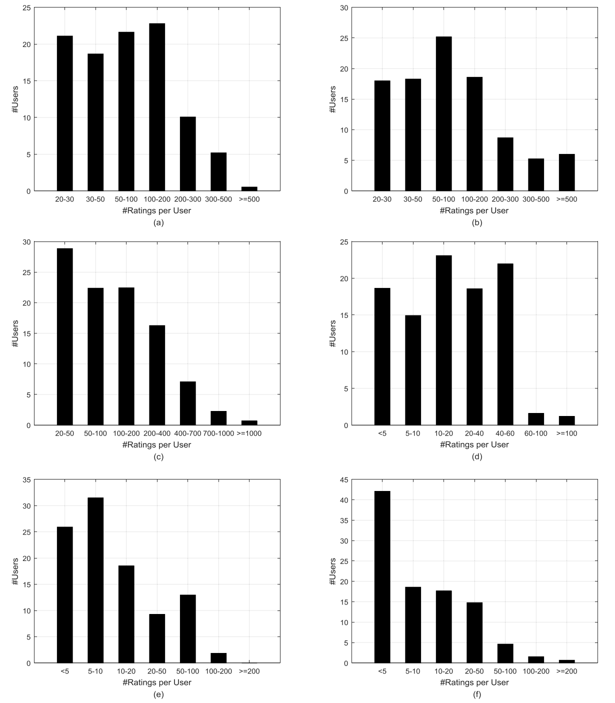


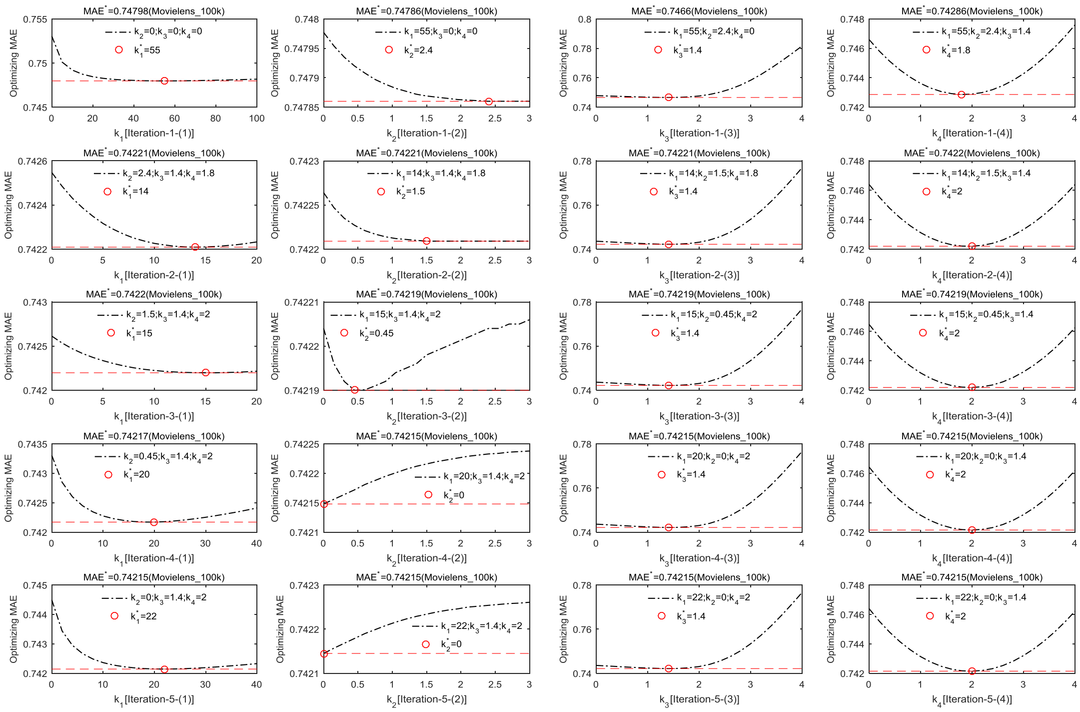
FIGURE 5. Distributions of the number of users according to the number of ratings per user.

(2) The second experiment is to evaluate the performances of different parts of RES, namely, the *consistency* component, *distance* factor and *Jaccard* factor. (3) The third experiment is to evaluate the performance of RES on all users with full ratings, and compare with those of other measures, to prove the global efficiency of RES similarity. (4) The fourth experiment is to evaluate the performance of RES on users grouped by the number of ratings, to prove the efficiency of RES on partial datasets. (5) The last experiment is to evaluate the performance of RES on cold-start users related to less than 20 ratings, to prove the efficiency of RES against the cold-start problem.

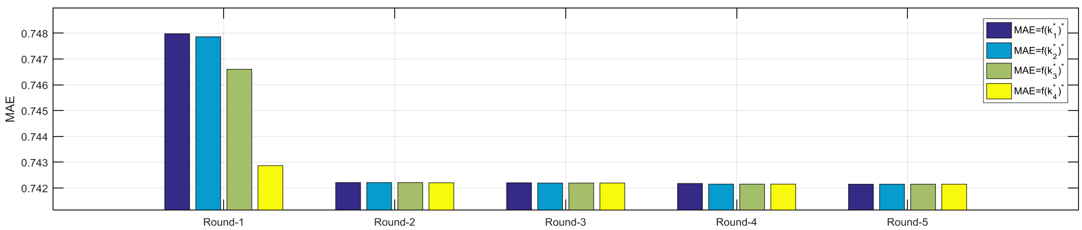
Each dataset in the experiments is separated into five parts. To perform a full evaluation of similarity performance, we use a cross-evaluation method in the experiments in which four parts (80%) of a given dataset are used for training while the remaining part (20%) is used for prediction. As a result, each part will be used for both training and prediction by cross-evaluations. The final performance on a given dataset is expressed as the average of the cross-prediction results.

A. EXPERIMENTAL DATASETS

As mentioned above, our experiments are performed on six real datasets, and Table 3 shows their statistics. The first three datasets are from Movielens. Movielens-100K has 943 users and 1682 items with 100K ratings. Movielens-latest-small is the latest small dataset of Movielens and has 700 users and 10K items with 100K ratings. MovieLens-1M has 6040 users and 3706 items with 100M ratings. The fourth



(a)



(b)

FIGURE 6. **Processes of performance optimization for RES. The illustrated results are obtained on dataset Movielens-100K. The processes on other datasets are similar to the above. (a) Optimization Iterations for MAE by learning parameters $k_1 \sim k_4$. (b) Trend of optimized MAE by learning parameters $k_1 \sim k_4$.**

TABLE 3. **Six experimental datasets.**

Data Set	#users	#items	#ratings	#density	rating scale
Movielens-100K	943	1682	100000	6.30%	[1, 5]
Movielens-latest-small	700	10000	100000	1.43%	[0.5, 5]
Movielens-1M	6040	3706	1000209	4.47%	[1, 5]
MiniFilm	55	334	1000	5.44%	[0.5, 4]
FilmTrust	1508	2071	35497	1.14%	[0.5, 4]
Epinions	40163	139738	664824	0.0118%	[1, 5]

dataset is FilmTrust, which has 1508 users and 2071 items with 35497 ratings. The fifth is a small mini dataset called MiniFilm, which has only 55 users and 334 items with 1K ratings. Epinions is a relatively large dataset with 40K users, 139K items and 664K ratings. We also describe the related rating densities and rating scales in the table.

Figure 5 shows the distribution of the number of users according to the number of ratings given by per user in the given dataset.

Most users have 20∼200 related items in datasets Movielens-100K and Movielens-latest-small, and most users in Movielens-1M are related to 20∼400 items. In datasets FilmTrust, MiniFilm and Epinions, more than half of the users are related to less than 20 items, and most of them are only related to less than 5 items. Users with no more than 20 ratings are assumed to be cold-start users, so these three datasets will be used in the cold-start experiments.

B. EXPERIMENT (1): OPTIMIZING PERFORMANCE AND LEARNING PARAMETERS

First, according to the optimization method described in equation (21), the predictive accuracy MAE of RES is optimized by learning parameter vector $\mathbb{K} = [k_1, k_2, k_3, k_4]$ on the six real datasets. We present the optimization processes on dataset Movielens-100K as a demonstration in Figure 6, and the optimizations on the other five datasets follow similar processes. Figure 6(a) demonstrates the iteration processes

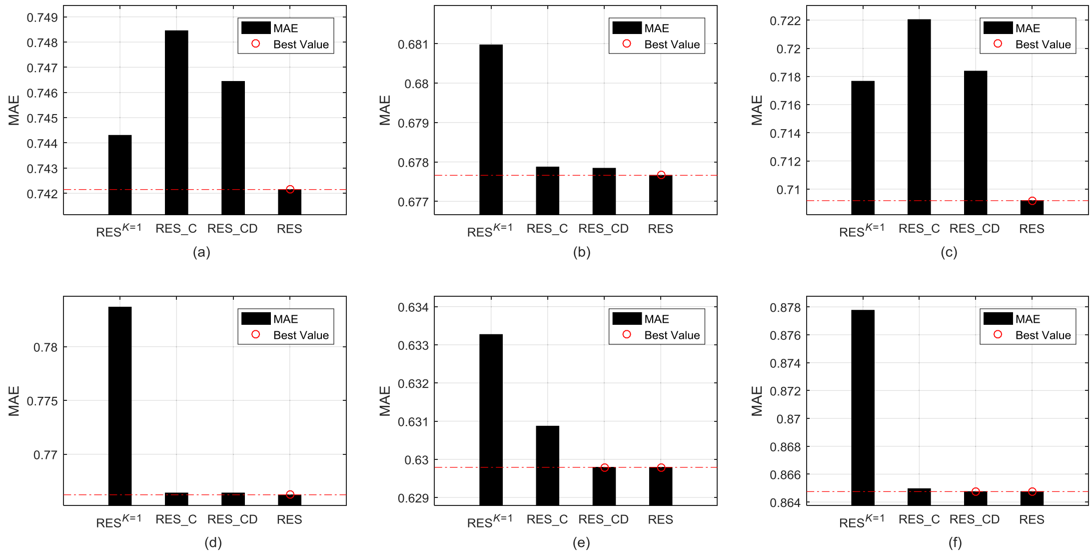


FIGURE 7. Performances of different parts in RES, on datasets Movielens-100K (a), Movielens-latest-small (b), Movielens-1M (c), MiniFilm (d), FilmTrust (e) and Epinions (f).

for MAE by learning $\mathbb{K} = [k_1, k_2, k_3, k_4]$ based on a stratified multi-objective optimization method, and each round of iteration has four steps. At the beginning, we initialized $MAE^* = 0.753046$ by using $\mathbb{K} = [1, 0, 0, 0]$ as an input for the first iteration, according to Algorithm 1. The optimization result of the first round of iteration is $MAE^* = 0.74286$ with $\mathbb{K}^* = [55, 2, 1.4, 1.8]$. As the iterations proceed, the optimized MAE^* decreases step by step, and MAE^* is optimized to 0.74215 at the end of the fourth round of iteration. During the fifth round of iteration, the MAE^* remains 0.74215, and $\Delta MAE < 0.0001$ all the time, so the optimization stops. Therefore, after five iterations, the optimization result is $MAE^* = 0.74215$ with $\mathbb{K}^* = [22, 0, 1.4, 2]$. Figure 6(b) demonstrates the trend of optimized MAE, obtained by learning parameters $k_1 \sim k_4$. The first round of optimization has a more dramatic influence on the predictive accuracy than the remaining rounds, which have relatively small influences on the optimized MAE. According to our observations during the experiments, if readers require quick optimization, they can try optimizing with only one round of iteration, which will result in relatively good accuracy.

Similarly, we optimize the predictive accuracy on all six real datasets with the same optimization method. The final optimized parameters $k_1 \sim k_4$ are recorded in Table 4. We observe that k_1 is always larger than 0.00, while the other parameters were optimized to 0.00 sometimes. This means that the *consistency* component is critical in RES, and *distance* factor and *Jaccard* factor are optional in RES according to the situation of the dataset.

C. EXPERIMENT (2): PERFORMANCES OF DIFFERENT PARTS IN RES

The purpose of this experiment is to evaluate the influences of *consistency* component $C(u, v, k_1)$, *distance* factor $D(u, v, k_2, k_3)$ and *Jaccard* factor $J(u, v, k_4)$ of RES. Performances are evaluated using the following combinations.

TABLE 4. Parameters $k_1 \sim k_4$ learned from full ratings.

Data Sets	Parameters				Optimized MAE
	k_1	k_2	k_3	k_4	MAE^*
Movielens-100K	22.00	0.00	1.40	2.00	0.74215
Movielens-latest-small	93.00	0.00	0.16	0.25	0.67766
Movielens-1M	20.00	5.00	2.20	3.40	0.70918
MiniFilm	29.00	0.00	0.00	0.29	0.62979
FilmTrust	46.00	1.10	1.20	0.00	0.76623
Epinions	0.35	0.09	0.15	0.01	0.86475

TABLE 5. Performances (MAEs) of different similarity measures.

Data Set	MAE by Similarity Measure			
	RES_C	RES_CD	RES^*	$RES^{k=1}$
Movielens-100K	0.74845	0.74644	0.74215*	0.74430
Movielens-latest-small	0.67787	0.67784	0.67766*	0.68097
Movielens-1M	0.72204	0.71838	0.70918*	0.71766
MiniFilm	0.76640	0.76640	0.76623*	0.78368
FilmTrust	0.63087	0.62979*	0.62979*	0.63327
Epinions	0.86496	0.86475*	0.86475*	0.87776

(1) **Only Consistency Considered**. We evaluate the performance of the *consistency* component of RES with the equation $RES\_C(u, v) = \sum_{I_{u,v}} C(u, v, k_1)$, to evaluate the influence of this critical component.

(2) **Consistency with Distance Factor Considered**. We evaluate the performance of the combination of the first two parts of RES with $RES\_CD(u, v) = \sum_{I_{u,v}} C(u, v, k_1) * D(u, v, k_2, k_3)$, where the *distance* factor, in addition to the *consistency* component, will influence the performance, to verify the two parts' influences on the similarity measure.

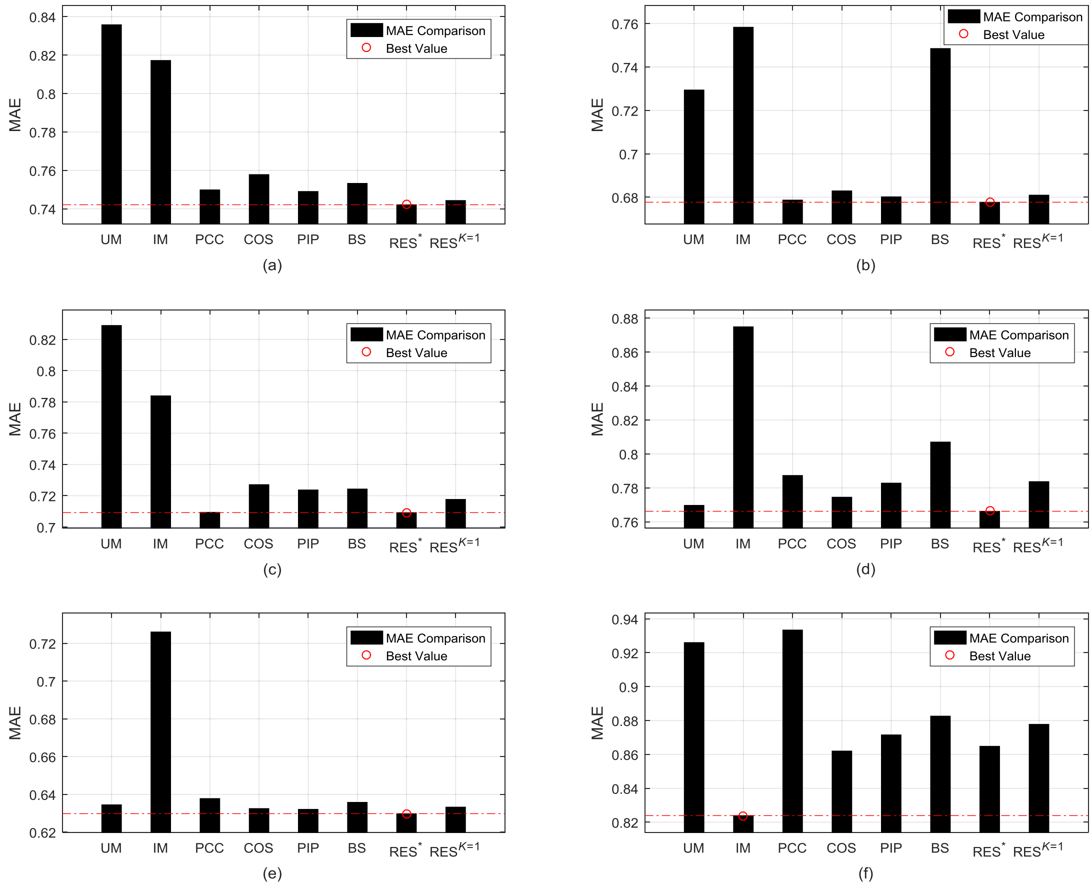


FIGURE 8. Performance Comparisons on all users with full ratings in different datasets: Movielens-100K (a), Movielens-latest-small (b), Movielens-1M (c.), MiniFilm (d), FilmTrust (e) and Epinions (f).

TABLE 6. Performances (MAEs) of Different Measures: UserMean, ItemMean, PCC, COS, PIP, BS, RES* with optimized parameters, and RES with parameters equal to '1.0'.

Data Set	MAE by Similarity Measure							
	UserMean	ItemMean	PCC	COS	PIP	BS	$RES^{k=1}$	RES^{*}
Movielens-100K	0.83576	0.81719	0.74979	0.75778	0.74896	0.75317	0.74430	0.74215*
Movielens-latest-small	0.72943	0.75828	0.67868	0.68294	0.68017	0.74851	0.68097	0.67766*
Movielens-1M	0.82895	0.78395	0.70938	0.72707	0.72368	0.72425	0.71766	0.70918*
MiniFilm	0.76967	0.87491	0.78734	0.77447	0.77823	0.80697	0.78368	0.76623*
FilmTrust	0.63452	0.72604	0.63783	0.63254	0.63214	0.63575	0.63327	0.62979*
Epinions	0.92602	0.82387*	0.93337	0.86197	0.87154	0.88261	0.87776	0.86475

(3) **RES with Specific** $\mathbb{K} = [1, 1, 1, 1]$. We evaluate the performance of RES with all parameters $[k_1, k_2, k_3, k_4]$ equal to 1 for comparison with the RES with optimized parameters; such an RES is denoted as$RES^{k=1}(u, v)$.

(4) **RES Itself with Optimized Parameters**. We evaluate the RES itself: $RES(u, v) = \sum_{I_{u,v}} C(u, v, k_1) * D(u, v, k_2, k_3) * J(u, v, k_4)$, where the parameters are learned from the dataset.

We compare the performances of the above combinations on full ratings in each dataset. The optimized MAE and parameter vector $\mathbb{K} = [k_1, k_2, k_3, k_4]$ for RES are as same as those in Table 4. Table 5 shows the results, and Figure 7 presents the results in an intuitive way.

According to the results, *RES* outperforms *RES_C* and *RES_CD*. Generally, *RES_C* has worse performance where there is only *consistency* component in RES, and *RES_CD* has relatively higher predictive performance. Sometimes, for examples in datasets FilmTrust and Epinions, *RES_CD* performs similarly to *RES*, which means the third part $J(u, v, k_4)$ is not necessary for these datasets and the related parameter is optimized to 0.0. If we set all parameters as $k_1 = k_2 = k_3 = k_4 = 1$, performance of $RES^{k=1}$ is always the worst.

This experiment proves that each of the three parts of RES is indispensable. The consistency component is the critical component for RES, while the distance factor and Jaccard factor are important for improving the accuracy. Using opti-

TABLE 7. Performance Comparison on User Groups with Different #ratings, using the similarity measures PCC, COS, PIP, BS, and RES. (a) Movielens-100K. (b) Movielens-latest-small(00K). (c) Movielens-1M. (d) MiniFilm. (e) FilmTrust. (f) Epinions.

(a)

User Group	#Rating per user	Performance (MAE)				
		PCC	COS	PIP	BS	RES
1	[0,20]	0.9333	0.8953	0.9050	0.9066	**0.8923***
2	[0,40]	0.8745	0.8364	0.8392	0.8510	**0.8331***
3	[0,60]	0.8454	0.8190	0.8173	0.8276	**0.8140***
4	[0,80]	0.8313	0.8083	0.8040	0.8156	**0.8015***
5	[0,100]	0.8162	0.7952	0.7885	0.8026	**0.7854***
6	[0,120]	0.7977	0.7816	0.7737	0.7872	**0.7701***
7	[0,140]	0.7904	0.7777	0.7694	0.7810	**0.7660***
8	[0,160]	0.7795	0.7703	0.7611	0.7717	**0.7575***
9	[0,180]	0.7743	0.7676	0.7586	0.7677	**0.7547***
10	[0,200]	0.7697	0.7656	0.7566	0.7644	**0.7527***

(b)

User Group	#Rating per user	Performance (MAE)				
		PCC	COS	PIP	BS	RES
1	[0,20]	0.9103	0.9024	0.9047	**0.8651***	0.8985**
2	[0,40]	0.8421	**0.8157***	0.8208	0.8219	0.8176**
3	[0,60]	0.8240	**0.7931***	0.7994	0.8175	0.7945**
4	[0,80]	0.7973	**0.7684***	0.7738	0.8014	0.7698**
5	[0,100]	0.7759	**0.7469***	0.7508	0.7857	0.7481**
6	[0,120]	0.7675	0.7425	0.7448	0.7818	**0.7422***
7	[0,140]	0.7553	0.7335	0.7351	0.7757	**0.7326***
8	[0,160]	0.7480	0.7268	0.7276	0.7699	**0.7253***
9	[0,180]	0.7419	0.7244	0.7246	0.7693	**0.7223***
10	[0,200]	0.7385	0.7238	0.7235	0.7676	**0.7211***

(c)

User Group	#Rating per user	Performance (MAE)				
		PCC	COS	PIP	BS	RES
1	[0,40]	0.8517	0.8017	0.8024	0.8174	**0.7989***
2	[0,80]	0.7888	0.7627	0.7589	0.7689	**0.7572***
3	[0,120]	0.7591	0.7502	0.7464	0.7538	**0.7436***
4	[0,160]	0.7440	0.7425	0.7384	0.7438	**0.7337***
5	[0,200]	0.7354	0.7393	0.7352	0.7393	**0.7295***
6	[0,240]	0.7278	0.7361	0.7321	0.7349	**0.7247***
7	[0,280]	0.7234	0.7342	0.7303	0.7326	**0.7218***
8	[0,320]	**0.7188***	0.7319	0.7281	0.7296	**0.7188***
9	[0,360]	**0.7169***	0.7312	0.7275	0.7288	0.7177
10	[0,400]	**0.7150***	0.7306	0.7269	0.7276	0.7162

(d)

User Group	#Rating per user	Performance (MAE)				
		PCC	COS	PIP	BS	RES
1	[0,20]	0.9444	0.8979	0.8980	0.9114	**0.8829***
2	[0,25]	0.8901	0.8073	0.8061	0.8425	**0.7812***
3	[0,30]	0.8901	0.8073	0.8061	0.8425	**0.7811***
4	[0,35]	0.8540	0.7586	0.7587	0.7854	**0.7436***
5	[0,40]	0.8375	0.7127	0.7247	0.7499	**0.7039***
6	[0,45]	0.8302	0.7221	0.7328	0.7620	**0.7121***
7	[0,50]	0.8302	0.7221	0.7328	0.7620	**0.7121***
8	[0,55]	0.8302	0.7221	0.7328	0.7620	**0.7121***
9	[0,60]	0.8302	0.7221	0.7328	0.7620	**0.7121***

(e)

User Group	#Rating per user	Performance (MAE)				
		PCC	COS	PIP	BS	RES
1	[0,20]	0.6276	0.6158	0.6131	0.6198	**0.6110***
2	[0,30]	0.6206	0.6104	0.6089	0.6128	**0.6071***
3	[0,40]	0.6064	0.6034	0.6021	0.6113	**0.5985***
4	[0,50]	0.6154	0.6114	0.6099	0.6191	**0.6067***
5	[0,60]	0.6181	0.6149	0.6137	0.6218	**0.6107***
6	[0,70]	0.6219	0.6188	0.6178	0.6250	**0.6149***
7	[0,80]	0.6260	0.6229	0.6219	0.6284	**0.6189***
8	[0,90]	0.6275	0.6240	0.6228	0.6297	**0.6201***
9	[0,100]	0.6304	0.6260	0.6249	0.6310	**0.6220***

(f)

User Group	#Rating per user	Performance (MAE)				
		PCC	COS	PIP	BS	RES
1	[0,20]	0.9973	0.9538	0.9586	0.9563	**0.9534***
2	[0,40]	0.9765	**0.9148***	0.9235	0.9286	0.9150**
3	[0,60]	0.9690	**0.9009***	0.9105	0.9170	0.9014**
4	[0,80]	0.9622	**0.8912***	0.9014	0.9083	0.8921**
5	[0,100]	0.9567	**0.8857***	0.8955	0.9034	0.8866**
6	[0,120]	0.9538	**0.8814***	0.8906	0.9000	0.8822**
7	[0,140]	0.9509	**0.8779***	0.8868	0.8971	0.8787**
8	[0,160]	0.9480	**0.8750***	0.8837	0.8943	0.8757**
9	[0,180]	0.9455	**0.8720***	0.8809	0.8917	0.8731**
10	[0,200]	0.9431	**0.8700***	0.8787	0.8896	0.8712**

mized parameters in RES is necessary to obtain the best predictive accuracy, and their values depend on the real dataset.

D. EXPERIMENT (3): PERFORMANCE COMPARISON ON ALL USERS WITH FULL RATINGS

The purpose of this experiment is to evaluate the predictive accuracy of RES on all users with full ratings. We select some famous similarity measures for comparison, namely, PCC, COS, BS [13] and PIP [14]; here, BS stands for Bayesian Similarity, which was proposed in [13]. The related system bias for BS is fixed as $\delta = 0.04$. We also take $\bar{r}_u$ (the average rating of user u, named *UserMean*) and $\bar{r}_i$ (the average rating of item i, named *ItemMean*), which are baselines for the similarity computation, as additional comparison objects. We perform the experiment on the full ratings in each dataset. Table 5 shows the evaluation results of this experiment, where the MAEs of RES are optimized results and are the same as in Table 4. Performance comparisons on each dataset are shown in Figure 8.

The proposed RES (with parameter optimized) exhibits advanced performances on most datasets, and it has higher accuracy than the others, except on Epinions. Compared with traditional similarity measure COS, the predictive accuracies of RES are 2.06%, 0.77%, and 2.46% higher on datasets Movielens-100K, Movielens-latest-small and Movielens-1M, respectively. Moreover, the predictive accuracies of RES are 1.06% and 0.43% higher compared to COS on datasets MiniFilm and FilmTrust, respectively. Compared with PCC, the predictive accuracies of RES are always higher on the six datasets. The predictive accuracies of RES are 1.02%, 2.68%, 1.26% and 7.35% higher compared to PCC on datasets Movielens-100K, MiniFilm, FilmTrust

TABLE 8. Performance (MAE) Comparison on Cold-Start Users with different measures: PCC, COS, PIP, BS, and RES.

#ratings per user	(a) FilmTrust					(b) Epinions					(c) MiniFilm				
	PCC	COS	PIP	BS	RES	PCC	COS	PIP	BS	RES	PCC	COS	PIP	BS	RES
1	0.7466	0.7466	0.7466	0.7466	0.7466	1.1420	1.1420	1.1420	1.1420	1.1420	0.8333	0.8333	0.8333	0.8333	0.8333
2	0.7128	0.6810	0.6741	0.6804	0.6753*	1.0670*	1.0692	1.0692	1.0675	1.0692	0.5000	0.5000	0.5000	0.5000	0.5000
3	0.7087*	0.7276	0.7288	0.7337	0.7159	1.0204	1.0206	1.0207	1.0210	1.0189*	0.6667	0.6667	0.6667	0.6667	0.6667
4	0.7481	0.7260	0.7416	0.7547	0.7207*	0.9869	0.9861	0.9858	0.9861	0.9854*	1.3330	1.1077	1.1637	1.3021	0.9003*
5	0.6086	0.5961	0.5975	0.6353	0.5809*	0.9684	0.9651	0.9651	0.9577*	0.9647	0.6533	0.5867	0.5867	0.5400	0.5067*
6	0.6410*	0.6554	0.6554	0.6967	0.6490	0.9862	0.9732	0.9733	0.9725	0.9717*	0.7503	0.6495	0.5972	0.4583	0.3463*
7	0.6890	0.6357*	0.6513	0.6543	0.6406	0.9742	0.9630	0.9631	0.9639	0.9628*	1.1801	1.1550	1.0809	1.1786	0.9611*
8	0.7043	0.6604	0.6734	0.7183	0.6593*	0.9750	0.9561	0.9566	0.9586	0.9557*	0.4682	0.3414	0.3416	0.4204	0.3139*
9	0.6922	0.6674	0.6574	0.7136	0.6605*	0.9839	0.9685*	0.9694	0.9702	0.9686	0.9222	0.9909	0.9965	0.9533*	0.9812
10	0.6516	0.6180	0.6209	0.6716	0.6107*	0.9764	0.9668	0.9689	0.9637	0.9662*	NaN	NaN	NaN	NaN	NaN
11	0.6579	0.6472	0.6438	0.6869	0.6426*	0.9727	0.9530	0.9554	0.9562	0.9521*	NaN	NaN	NaN	NaN	NaN
12	0.5535	0.5539	0.5558	0.5779	0.5477*	0.9486	0.9253	0.9261	0.9210*	0.9249	NaN	NaN	NaN	NaN	NaN
13	0.6289	0.6055*	0.6203	0.6617	0.6107*	0.9181	0.9006	0.9023	0.8995*	0.9006	0.5860	0.6629	0.6607	0.6530	0.3752*
14	0.7113	0.6444	0.6478	0.7052	0.6444*	0.9465	0.9237	0.9265	0.9212*	0.9236	1.0305	0.9278	0.9096	1.2125	0.6085*
15	0.6998	0.6801	0.6807	0.7316	0.6758*	0.9636	0.9434	0.9429	0.9444	0.9418*	0.8195	1.0189	0.9980	0.8421	0.7714*
16	0.6040	0.5934	0.6000	0.6342	0.5933*	0.9340	0.9294	0.9317	0.9141*	0.9293	NaN	NaN	NaN	NaN	NaN
17	0.6351	0.5954*	0.5999	0.6154	0.5969*	0.9489	0.9360	0.9340	0.9222*	0.9337	1.0000	1.0000	1.0000	1.0000	1.0000
18	0.6606	0.6518	0.6499	0.6878	0.6481*	0.9623	0.9421	0.9433	0.9530	0.9414*	NaN	NaN	NaN	NaN	NaN
19	0.6128	0.5994	0.5990	0.6488	0.5938*	0.9664	0.9295	0.9356	0.9258*	0.9314	1.0460	0.9466	0.9418	1.0000	0.6250*
20	0.6527	0.6376	0.6384	0.6534	0.6354*	1.0148	0.9863	0.9853	0.9889	0.9825*	NaN	NaN	NaN	NaN	NaN

and Epinions. We also take the RES with parameter vector $\mathbb{K} = [1, 1, 1, 1]$ for comparisons in this experiment (denoted as $\mathrm{RES}^{\mathrm{k=1}}$). As discussed in the previous experiments, the predictive accuracy of $\mathrm{RES}^{\mathrm{k=1}}$ is worse than RES with parameters optimized. However, its predictive accuracies are 1.78%, 0.29% and 1.29% higher than COS on datasets Movielens-100K, Movielens-latest-small and Movielens-1M, respectively; and 0.73%, 0.47%, 0.71% and 5.96% higher than PCC on datasets Movielens-100K, MiniFilm, FilmTrust and Epinions, respectively. By comparison with PIP and BS, which are considered relatively state-of-the-art methods, RES shows superior performance on all datasets. Experiments also prove that UserMean and ItemMean are not reliably suitable for prediction, even though ItemMean returns the best accuracy on Epinions.

This experiment proves that RES has good performance in terms of predictive accuracy when the similarity is computed based on all users with full ratings, and it is superior to current typical similarity measures.

E. EXPERIMENT (4): PERFORMANCE COMPARISON ON GROUPED USERS

The purpose of the experiment is to evaluate the predictive performance of RES on users with various numbers of ratings. We divide users into several groups according to the user distribution in the given datasets, as discussed in the section on datasets. Table 7 shows the user groups divided according to the number of ratings. Let's take users in Movielens-100K as an example. Since most users have 20∼200 related items in dataset Movielens-100K (see Figure 5(a)), we divide the users into 10 groups in which each group covers the preceding group: the first user group in Movielens-100K includes users related to [0, 20] items, the second group includes users related to [0, 40] items, and the last user group includes users related to [0, 200] items. For the other five datasets, we use similar methods to divide the users into groups. Then, we evaluate the performance (MAE) using different similarity measures on each user group. Similar to experiment (1), the MAE of RES is optimized on each user group. The results of all similarity measures are listed in Table 7, and Figure 9 shows the performance comparisons visually.

According to the results, RES has excellent performance on all datasets. On datasets Movielens-100K, Movielens-1M, MiniFilm and FilmTrust, RES has the best predictive accuracy, where we mark MAE of RES with '$*$'. On datasets Movielens-latest-small and Epinions, the performances of RES rank in the top two and the corresponding results are marked with '$*$' or '$**$'. The predictive accuracies of all similarity measures vary for different user groups, and the MAEs decrease continually on most datasets, except on FilmTrust, where the MAE falls first and rises later. We also observed that the RES has relatively high performance on the first user group of each dataset, where the number of related ratings is less than 20 or 40. Users with less than 20 ratings are considered cold-start users; thus, we perform an experiment to further investigate the performance of RES on cold-start users in the next section.

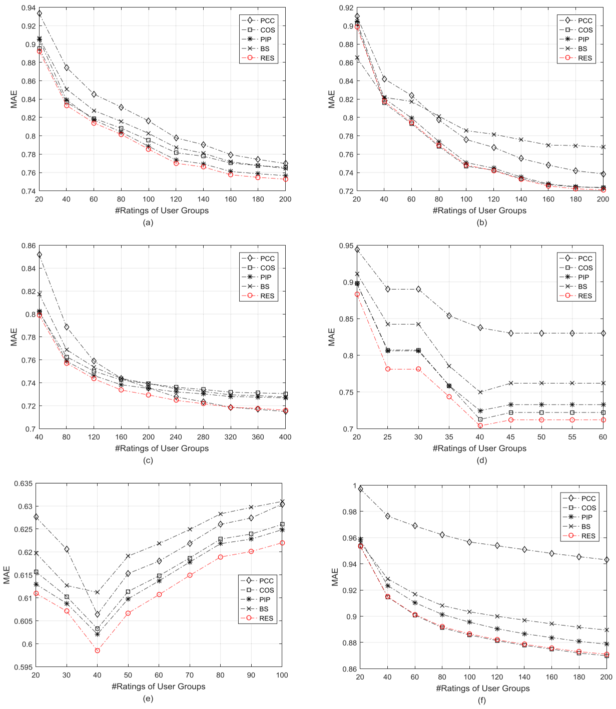


FIGURE 9. **Performance comparisons on users grouped by different #ratings, on datasets Movielens-100K (a), Movielens-latest-small (b),Movielens-1M (c),MiniFilm (d),FilmTrust (e) and Epinions (f).**

This experiment proves that RES has superior predictive accuracy when similarity is computed on users grouped by different numbers of ratings. Thus, RES is suitable for measuring users' similarity on a subset of a given dataset.

F. EXPERIMENT (5): PERFORMANCE COMPARISON ON COLD-START USERS

In user-based CF recommender systems, the cold-start problem, in which users do not have enough ratings for similarity measurement, always degrades the predictive accuracy of the recommendation. This experiment is to evaluate the predictive performance of RES on cold-start users, and compare it with those of other similarity measures. Based on the observation that more than half of the users have less than 20 related items in datasets FilmTrust, MiniFilm and Epinions, we perform this experiment on cold-start users based on these three datasets. All cold-start users related to 1∼20 items are selected for computation and cold-start users with same number of ratings are evaluated by similarity measures PCC, COS, PIP, BS and RES. Table 8 shows the MAE results, and Figure 10 shows them visually. 'NaN' means there is no user related to the corresponding number of ratings. Since there are no users related to {10,11,12,16,18,20} ratings in dataset MiniFilm, the MAE results for all the corresponding similarity measures are marked as 'NaN'.

In the above table, we mark the best performance with '$*$'. The results show that RES outperforms the other similarity measures in most cases. On dataset FilmTrust, the performances of RES are obviously superior to others, except when #ratings={3,6,7} where the performance of RES is also ranked top two. On dataset Epinions, the RES is ranked in the top two, and its performance is superior to that of PCC and similar to those of PIP and COS, but sometimes inferior to that of BS. On dataset MiniFilm, the performances of RES are superior especially when the number of ratings per user is larger than 5. When the number of ratings per user is less than 5, the performance of RES has a relatively

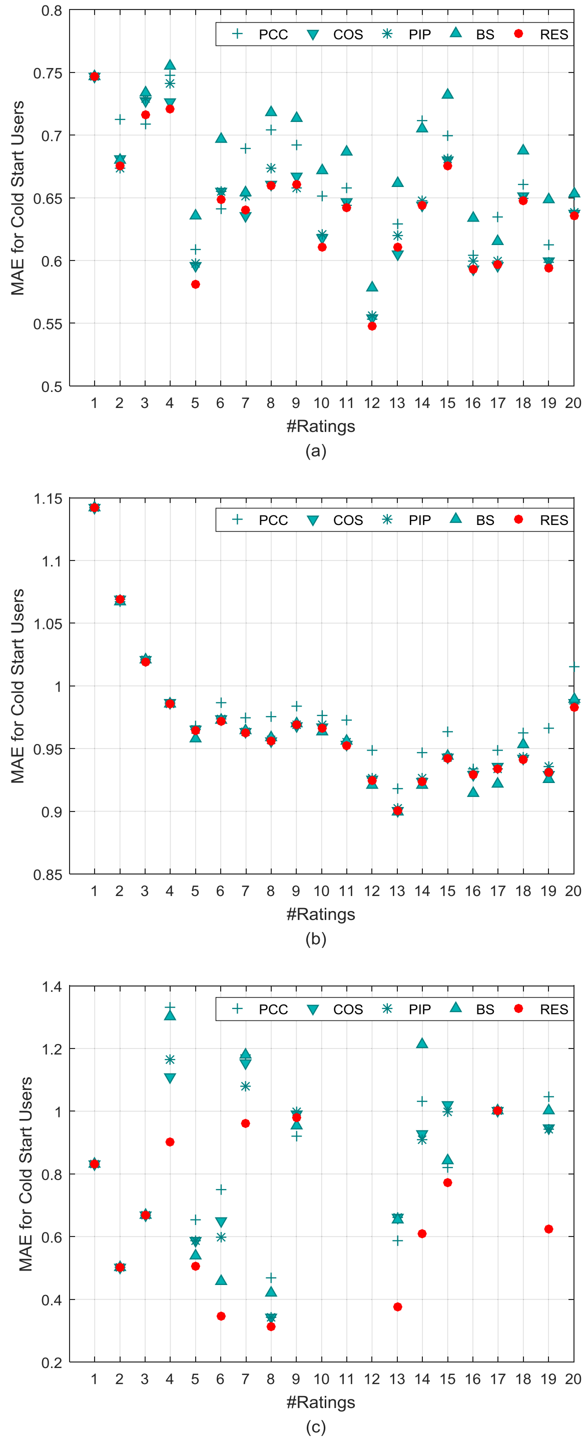


FIGURE 10. **Performance comparisons on cold-start users, on datasets FilmTrust (a), Epinions (b) and MiniFilm (c).**

slow advance. For example, RES has same performance as the others when #ratings={1,2,3}. However, when #ratings=4 in MiniFilm, the result of RES is obviously superior, where the MAE of RES is 0.9003, but those of the others are all larger than 1.1017.

This experiment proves that the RES has good performance against the cold-start problem; thus, it has ability to measure the similarity for cold-start users in user-based CF recommender systems when there are not enough ratings to calculate the standard similarity measures.

VI. CONCLUSION

Inspired by the physical resonance phenomenon, this paper proposed a novel similarity measure named RES for user-based CF recommender systems. During the mathematical modeling, we fully consider the personalization of user ratings in different cases. RES consists of the *consistency* component, *distance* factor and *Jaccard* factor. Users' rating behaviors in a CF recommender system are regarded as simple harmonic vibrational motions in a virtual resonance system, where the critical *consistency* component in RES measures two users' rating consistency by modeling their initial phase angles, the *distance* factor complements the *consistency* component by calculating the distance of users' rating opinions of the same co-related item, and the *Jaccard* factor weights the *consistency* component by both co-related ratings and *non-related* ratings. We also propose an optimization method to improve the performance of RES by learning parameters from the given dataset iteratively. The predictive accuracy of RES is evaluated by MAE on six real datasets. According to comparisons with traditional similarity measures and some state-of-the-art measures, RES is robust against the observed problems of traditional measures and exhibits superior performance on full users', grouped users', and cold-start users' evaluations.

REFERENCES

[1] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending and evaluating choices in a virtual community of use," in *Proc. SIGCHI*, Denver, CO, USA, 1995, pp. 194–201, doi: 10.1145/223904.223929.

[2] U. Shardanand and P. Maes, "Social information filtering: Algorithms for automating 'word of mouth,'" in *Proc. SIGCHI*, Denver, CO, USA, 1995, pp. 210–217, doi: 10.1145/223904.223931.

[3] R. Sharma, D. Gopalani, and Y. Meena, "Collaborative filtering-based recommender system: Approaches and research challenges," in *Proc. CICT*, Ghaziabad, India, Feb. 2017, pp. 1–6, doi: 10.1109/CIACT.2017.7977363.

[4] Y.-Y. Shih and D.-R. Liu, "Product recommendation approaches: Collaborative filtering via customer lifetime value and customer demands," *Expert Syst. Appl.*, vol. 35, nos. 1–2, pp. 350–360, Jul. 2008, doi: 10.1016/j.eswa.2007.07.055.

[5] B. Xie, P. Han, F. Yang, R.-M. Shen, H.-J. Zeng, and Z. Chen, "DCFLA: A distributed collaborative-filtering neighbor-locating algorithm," *Inf. Sci.*, vol. 177, no. 6, pp. 1349–1363, Mar. 2007, doi: 10.1016/j.ins.2006.09.005.

[6] G. Badaro, H. Hajj, W. El-Hajj, and L. Nachman, "A hybrid approach with collaborative filtering for recommender systems," in *Proc. IWCMC*, Sardinia, Italy, Jul. 2013, pp. 349–354, doi: 10.1109/IWCMC.2013.6583584.

[7] L. Candillier, F. Meyer, and F. Fessant, "Designing specific weighted similarity measures to improve collaborative filtering systems," in *Proc. ICDM*, Leipzig, Germany, 2008, pp. 242–255, doi: 10.1007/978-3-540-70720-2_19.

[8] Z. Tan, L. He, H. Li, and X. Wang, "Rating personalization improves accuracy: A proportion-based baseline estimate model for collaborative recommendation," in *Proc. CollaborateCOM*, Beijing, China, 2016, pp. 104–114, doi: 10.1007/978-3-319-59288-6_10.

[9] G. Cao and L. Kuang, "Identifying core users based on trust relationships and interest similarity in recommender system," in *Proc. ICWS*, San Francisco, CA, USA, Jun. 2016, pp. 284–291, doi: 10.1109/ICWS.2016.44.

[10] I. Bartolini, Z. Zhang, and D. Papadias, "Collaborative filtering with personalized skylines," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 2, pp. 190–203, Feb. 2011, doi: 10.1109/TKDE.2010.86.

[11] W. Li, H. Xu, M. Ji, Z. Xu, and H. Fang, "A hierarchy weighting similarity measure to improve user-based collaborative filtering algorithm," in *Proc. ICCC*, Chengdu, China, Oct. 2016, pp. 843–846, doi: 10.1109/CompComm.2016.7924821.

[12] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel Bayesian similarity measure for recommender systems," in *Proc. IJCAI*, Beijing China, 2013, pp. 2619–2625. [Online]. Available: http://www.ijcai.org/Proceedings/13/Papers/386.pdf

[13] 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, doi: 10.1145/2856037.

[14] 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, doi: 10.1016/j.ins.2007.07.024.

[15] 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. HiPCW*, Hyderabad, India, Dec. 2016, pp. 32–38, doi: 10.1109/HiPCW.2016.013.

[16] S. Wei, N. Ye, S. Zhang, X. Huang, and J. Zhu, "Collaborative filtering recommendation algorithm based on item clustering and global similarity," in *Proc. BIFE*, Lanzhou, China, Aug. 2012, pp. 69–72, doi: 10.1109/BIFE.2012.23.

[17] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proc. UAI*, Madison, WI, USA, 1998, pp. 43–52.

[18] N. Lathia, S. Hailes, and L. Capra, "The effect of correlation coefficients on communities of recommenders," in *Proc. SAC*, Fortaleza, Brazil, 2008, pp. 2000–2005, doi: 10.1145/1363686.1364172.

[19] M. Schwarz, M. Lobur, and Y. Stekh, "Analysis of the effectiveness of similarity measures for recommender systems," in *Proc. CADSM*, Lviv, Ukraine, Feb. 2017, pp. 275–277, doi: 10.1109/CADSM.2017.7916133.

[20] H. Ma, I. King, and M. R. Lyu, "Effective missing data prediction for collaborative filtering," in *Proc. SIGIR*, Amsterdam, The Netherlands, 2007, pp. 39–46, doi: 10.1145/1277741.1277751.

[21] Z. Tan, X. Wang, and X. Wang, "A novel iterative and dynamic trust computing model for large scaled P2P networks," *Mobile Inf. Syst.*, vol. 2016, Jan. 2016, Art. no. 3610157, doi: 10.1155/2016/3610157.

[22] A. Said, B. J. Jain, and S. Albayrak, "Analyzing weighting schemes in collaborative filtering: Cold start, post cold start and power users," in *Proc. SAC*, 2012, pp. 2035–2040, doi: 10.1145/2245276.2232114.

[23] J. Bobadilla, F. Ortega, and A. Hernando, "A collaborative filtering similarity measure based on singularities," *Inf. Process. Manage.*, vol. 48, no. 2, pp. 204–217, Mar. 2012, doi: 10.1016/j.ipm.2011.03.007.

[24] S. Aygün and S. Okyay, "Improving the pearson similarity equation for recommender systems by age parameter," in *Proc. AIEEE*, Riga, Latvia, Nov. 2015, pp. 1–6, doi: 10.1109/AIEEE.2015.7367282.

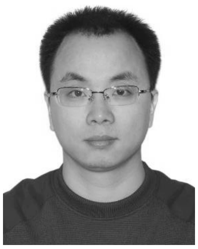
[25] R. Bell, Y. Koren, and C. Volinsky, "Modeling relationships at multiple scales to improve accuracy of large recommender systems," in *Proc. SIGKDD*, San Jose, CA, USA, 2017, pp. 95–104, doi: 10.1145/1281192.1281206.

[26] F. Ghaznavi and S. H. Alizadeh, "Assessing usage of negative similarity and distrust information in CF-based recommender system," in *Proc. IRANOPEN*, Qazvin, Iran, Apr. 2017, pp. 132–138, doi: 10.1109/RIOS.2017.7956456.

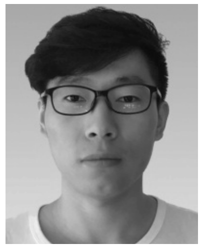
[27] B. H. Huang and B. R. Dai, "A weighted distance similarity model to improve the accuracy of collaborative recommender system," in *Proc. MDM*, Pittsburgh, PA, USA, Jun. 2015, pp. 104–109, doi: 10.1109/MDM.2015.43.

[28] R. Reshma, G. Ambikesh, and P. S. Thilagam, "Alleviating data sparsity and cold start in recommender systems using social behaviour," in *Proc. ICRTIT*, Chennai, India, Apr. 2016, pp. 1–8, doi: 10.1109/ICRTIT.2016.7569532.

[29] G. Lekakos and G. M. Giaglis, "Improving the prediction accuracy of recommendation algorithms: Approaches anchored on human factors," *Interacting Comput.*, vol. 18, no. 3, pp. 410–431, 2006, doi: 10.1016/j.intcom.2005.11.004.
[30] G. Lekakos and G. M. Giaglis, "A hybrid approach for improving predictive accuracy of collaborative filtering algorithms," *User Model. User-Adapt. Interact.*, vol. 17, nos. 1–2, pp. 5–40, 2007, doi: 10.1007/s11257-006-9019-0.
[31] P. Winoto and T. Y. Tang, "The role of user mood in movie recommendations," *Expert Syst. Appl.*, vol.37, no. 8, pp. 6086–6092, 2010, doi: 10.1016/j.eswa.2010.02.117.



ZHENHUA TAN (M'17) was born in Hu Nan, China, in 1980. He received the B.S., M.S., and Ph.D. degrees in computer science from Northeastern University, Shenyang, China, in 2003, 2006, and 2009, respectively. He is currently an Associate Professor with the College of Software, Northeastern University. He holds three U.S. patents about networking and security. He has published over 30 journal articles, book chapters, and refereed conference papers. His current research interests include networking behaviors analysis and information security.



LIANGLIANG HE received the B.S. degree from Xi'an Technological University, Xi'an, China, in 2015. He is currently pursuing the M.S. degree with the Software College of Northeastern University, China. His research interests include CF recommender system and Networking behaviors analysis.