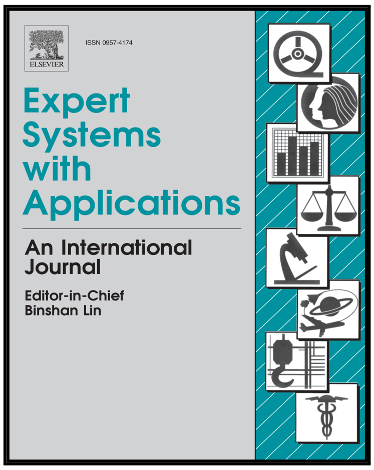
Accepted Manuscript

An Intuitionistic Fuzzy Set Based Hybrid Similarity Model for Recommender System

Junpeng Guo , Jiangzhou Deng , Yong Wang



PII: S0957-4174(19)30408-7
DOI: https://doi.org/10.1016/j.eswa.2019.06.008
Reference: ESWA 12720

To appear in: *Expert Systems With Applications*

Received date: 13 January 2019
Revised date: 28 March 2019
Accepted date: 5 June 2019

Please cite this article as: Junpeng Guo , Jiangzhou Deng , Yong Wang , An Intuitionistic Fuzzy Set Based Hybrid Similarity Model for Recommender System, *Expert Systems With Applications* (2019), doi: https://doi.org/10.1016/j.eswa.2019.06.008

Highlights

- An adjusted Google similarity is proposed in condition of sufficient co-rated items
- A fuzzy set based KL similarity is proposed in condition of rare co-rated items
- Proposed schemes are integrated in a certain range of co-rated items
- Results reveal that our system has a favorable efficiency and accuracy

An Intuitionistic Fuzzy Set Based Hybrid Similarity Model for Recommender System

Junpeng Guo [a] Jiangzhou Deng [a*] Yong Wang [b]

[a] College of Management and Economics, Tianjin University, Tianjin 30072, China
[b] Key Laboratory of Electronic Commerce and Logistics of Chongqing, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Abstract

In general, a practical online recommendation system does not rely on only one algorithm but adopts different types of algorithms to predict user preferences. Although most of similarity measures can rapidly calculate the similarity on the basis of co-rated items, their prediction accuracy is not satisfactory in the case of sparse datasets. Making full use of all the rating information can effectively improve the recommendation quality, but it reduces the system efficiency because all the ratings need to be calculated. To recommend items for target users rapidly and accurately, this paper designs a hybrid item similarity model that achieves a trade-off between prediction accuracy and efficiency by combining the advantages of the two above-mentioned methods. First, we introduce an adjusted Google similarity to rapidly and precisely calculate the item similarity in the condition of enough co-rated items. Subsequently, an intuitionistic fuzzy set (IFS) based Kullback-Leibler (KL) similarity is presented from the perspective of user preference probability to effectively compute the item similarity in the condition of rare co-rated items. Finally, the two proposed schemes are integrated by an adjusted variable to comprehensively evaluate the similarity values when the number of co-rated items lies in a certain range of value. The proposed model is implemented and tested on some benchmark datasets with different thresholds of co-rated items. The experimental results indication that the proposed system has a favorable efficiency and guarantees the quality of recommendations.

Keywords: Recommender system; Collaborative filtering; Normalized Google distance; Intuitionistic fuzzy set; Kullback–Leibler divergence

---

* **Corresponding author**
E-mail address: guojp@tju.edu.cn (J. Guo), dengjz114@163.com (J. Deng), wangyong1@cqupt.edu.cn (Y. Wang)

1. Introduction

The rapid growth of the Internet has tremendously boosted online enterprises, especially e-commerce, providing consumers with a wide variety of choices in books (Amazon), videos (YouTube) and photos (Flickr) (Baluja, Seth, Sivakumar, Jing, Yagnik, Kumar, Ravichandran, & Aly, 2008; Brynjolfsson, Hu, & Smith, 2003; Zheng, Li, Liao, & Zhang, 2010), etc. However, the massive amount of information on the Internet usually overwhelms users and makes them indecisive. (Liu, Hu, Mian, Tian, & Zhu, 2014). Recommender systems (RS) have been successfully deployed to provide information, make recommendations, and facilitate decision-making on products of interest for active users (Davidson, Livingston, Sampath, Liebald, Liu, & Nandy, 2010; Shahabi, Banaei-Kashani, Chen, & Mcleod, 2001; Takeuchi, & Sugimoto, 2006). They can match users' expectations and points of interest by analyzing their previous preference behaviors of users, thereby addressing the information overload problem effectively.

As one of the best-known recommendation techniques, collaborative filtering (CF) (Breese, Heckerman, & Kadie, 2013) has been adopted by numerous e-commerce websites. It provides unknown items to the target users by learning the potential interests of the users. The general recommendation process of CF involves three main steps. The first step calculates the similarity degree among users. The second step selects the most similar users with the target users as the nearest neighbors. Finally, the third step predicts the preferences of users and recommends items for them. In general, the co-rated items (Arwar, Karypls, Konstan, & Riedl, 2001; Patra, Launonen, Ollikainen, & Nandi, 2014) play a decisive role in similarity calculation. The classic similarity methods based CF, such as cosine similarity and the Pearson correlation coefficient, can quickly return recommendation results because they only depend on co-rated items and do not consider other non-co-rated items, thereby improving the operating efficiency of recommendation systems significantly. However, these methods ignore the influence of the number of co-rated items that directly affect the accuracy and reliability of the similarity results. In particular, in the case of sparse data, most of similarity measures always suffer from the problem of insufficient co-rated items and they will not work if there are no co-rated items in a given dataset. Moreover, it is not reasonable to ignore the effect of the non-co-rated items that are likely to contain the potentially valuable information for the similarity calculation. To address the dependence on co-rated items, Patra, Launonen, Ollikainen, and Nandi (2015) and Wang, Deng, Gao, and Zhang (2017) proposed a linear similarity model and a non-linear similarity model, respectively, to fully exploit all the rating information from the perspective of item probability distribution. Although these two schemes improve the prediction accuracy effectively, their time complexity is higher than that of classic methods.

It is known that recommendation systems aim to recommend items rapidly and accurately for target users. In practical applications, a recommendation system is not likely to depend on only one recommendation algorithm to predict user preferences. It often selects different algorithms to provide decision-making schemes for users according to different conditions.

In this paper, we design a hybrid item similarity model for a recommendation system to achieve a trade-off between prediction accuracy and efficiency. The concept of the hybrid similarity model and its specific framework are described in detail in the next section. In our hybrid similarity model, an adjusted Google similarity is firstly used to calculate item similarity in the condition of sufficient co-rated items. This method consists of four functions, i.e., a positive function and three negative functions, and it considers the number of ratings, the user preferences, and the proportion of co-rated items to provide more accurate and reliable results compared to other similarity models that depends on the co-rated items. Nevertheless, our model still suffers from the problem of co-rated items. Therefore, to compute item similarity in the condition of rare co-rated items, we present a Kullback-Leiblber (KL) similarity based on intuitionistic fuzzy set (IFS) from the perspective of user preference probability. The proposed algorithm fully exploits all the rating information regardless of the number of co-rated items. It can evaluate the similarity even in the absence of co-rated items, which enhances the utility of the calculation results. Moreover, the IFS based KL method considers the rating preferences (like or dislike) of users as well as the rating uncertainty of users faced with unknown items. However, its computation time is relatively long because it calculates all the ratings. Finally, to comprehensively evaluate the similarity values, the two above-mentioned schemes are integrated by an adjusted variable when the number of co-rated items lies in a certain range of value. The experimental results on different datasets indicate that our proposed system has a favorable efficiency and guarantees the quality of recommendations.

The remainder of this paper is organized as follows: we briefly introduce the related works about CF, Google similarity, and intuitionistic fuzzy set in Section 2. In Section 3, we describe the research framework in detail. The proposed model is introduced in Section 4. Section 5 discusses the experimental results in various evaluation indictors. Conclusions and further work are presented in Section 6.

2. Literature review

Section 2.1 introduces the recommendation mechanism of collaborative filtering (CF), including identifying the user rating matrix, calculating the similarity, finding the nearest neighbors and generating the final recommendation list. Sections 2.2 and 2.3 elaborate the use of Google distance and intuitionistic fuzzy set, respectively, in the proposed system.

2.1 *Collaborative filtering*

The concept of collaborative filtering (CF) was originally proposed by Goldberg, Nichols, Oki, and Terry (1992). Figure 1 shows that the entire recommendation process based on CF. Suppose that m users and n items are collected as a set of users and a set of items, $U = \{u_1, u_2, \ldots, u_m\}$ and $I = \{i_1, i_2, \ldots, i_n\}$, respectively. All the rating data are represented by a user rating matrix $[r_{ui}]^{m\times n}$ (see fig. 1), where r_{ui} is a rating score made by the u^{th} user on the i^{th} item. Note that most of the ratings in the rating matrix are unknown, which leads to the sparsity problem for similarity calculation.

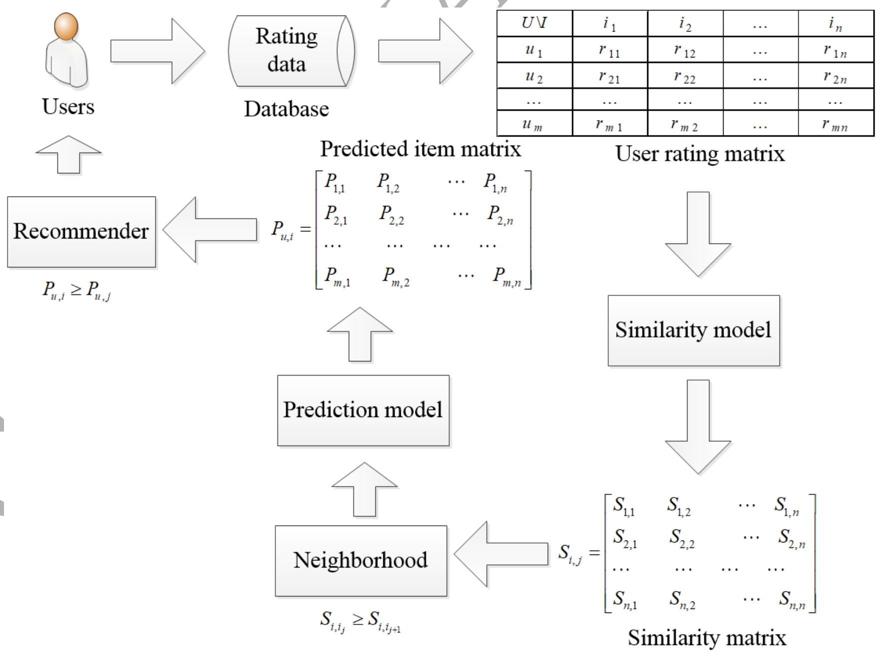


Figure 1 Recommendation mechanism of CF

The cosine similarity (COS) (Arwar, Karypls, Konstan, & Riedl, 2001; Chowdhury, 1983) and Pearson correlation coefficient (PCC) (Arwar, Karypls, Konstan, & Riedl, 2001; Ekstrand, Riedl, & Konstan, 2007) are as the most widely used measures of similarity between two users or items. Their formulas based on item similarity are expressed as follows (Arwar, Karypls, Konstan, & Riedl, 2001):

$$sim(i,j)_{COS} = \frac{\sum_{u \in U_i \cap U_j} (r_{ui})(r_{uj})}{\sqrt{\sum_{u \in U_i \cap U_j} r_{ui}^2}\sqrt{\sum_{u \in U_i \cap U_j} r_{uj}^2}} \quad (1)$$

$$sim(i,j)_{PCC} = \frac{\sum_{u \in U_i \cap U_j} (r_{ui}-\overline{r}_i)(r_{uj}-\overline{r}_j)}{\sqrt{\sum_{u \in U_i \cap U_j} (r_{ui}-\overline{r}_i)^2}\sqrt{\sum_{u \in U_i \cap U_j} (r_{uj}-\overline{r}_j)^2}} \quad (2)$$

where U_i is a set of users rating item i, and $\overline{r}_i$ is the average rating value of item i. Althrough these measures have achieved considerable success, with the advent of the big data era, they have been found to be unsuitable for sparse envorinments mainly because they have to depend on the co-rated items when computing similarity values. In sparse data, co-rated items are extremely rare , which leads to the inaccurate calculation results. To alleviate the sparsity problem, researchers have proposed several well-known memory-based solutions in recent years.

To "punish" the bad similarity and "reward" the good similarity, Liu, Hu, Mian, Tian, and Zhu (2014) introduced a new heuristic similarity model (NHSM) based on a non-linear function to overcome the shortcoming of the initial heuristic similarity measure, i.e., proximity-impact-popularity (PIP) (Ahn, 2008). This approach improves the prediction accuracy and alleviates the cold-start problem to some extent. In general, the similarity value in previous studies is symmetric: the similarity between objects *a* and *b* is the same as the similarity between objects *b* and *a*, i.e., $sim(a, b) = sim(b, a)$. To break the symmetric mode, Pirasteh, Hwang, and Jung (2015) exploited two weighted factors, namely the compromise factor and the accordance factor, to show the asymmetric relationship between users in some classic similarity methods and thus improved the recommendation quality effectively. However, all the above-mentioned similarity methods still suffer from the effect of rare co-rated items. To eliminate the dependency on co-rated items, a Bhattacharyya coefficient based linear similarity measure (BCF) (Patra, Launonen, Ollikainen, & Nandi, 2015) and a Kullback–Leibler (KL) based non-linear model (KL-NHSM) (Wang, Deng, Gao, & Zhang, 2017) were proposed from the perspective of item probability distribution, respectively. These two schemes can make full use of all the rating information, and they can even calculate the similarity values when there is no co-rated item in the system. These methods enhance the sphere of similarity computation and can effectively deal with the sparsity problem. However, their computational efficiency is relatively lower than that of other methods because they consider all the ratings.

The similarity matrix is obtained by calculating similarity value between any two items (see fig. 1). Then, the K the nearest neighbors of a target item i in the matrix are found according to the selected rule that satisfies the similarity value $S_{i,i_j} \geq S_{i,i_{j+1}}$, $j = 1, 2, \ldots, n$, $i \neq j$. Thus, unknown rating $P_{u,i}$ of an active user u on an item i is computed by the prediction model (Karypis, 2001; Patra, Launonen, Ollikainen, & Nandi, 2014) as follows:

$$P_{u,i} = \overline{r}_i + \frac{\sum_{k=1}^{K} sim(i,j)(r_{uj} - \overline{r}_j)}{\sum_{k=1}^{K} sim(i,j)} \tag{3}$$

Finally, N items with the highest prediction values are selected as a recommended list from the predicted rating set of the active user u on all the unrated items. Thus, the online top-N recommendation is determined.

2.2 *Normalized Google distance*

The normalized Google distance (NGD) is first introduced by Cilibrasi and Vitanyi (2007) to evaluate the correlation between two words or phrases from the World-Wide-Web (WWW) using the Google search engine. The NGD is a semantic measurement derived from the number of hits returned in a query for a given set of keywords, and it calculates the logical distance and reflects the semantic similarity between two words directly. The formula for NGD is given by

$$NGD(x,y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x,y)}{\log N - \min\{\log f(x), \log f(y)\}} \tag{4}$$

where $f(x)$ denotes the number of pages containing term x, and $f(x, y)$ represents the number of hits returned containing both terms x and y, and N is the maximum number of web pages returned by the Google search engine. It can be seen from Eq. (4) that the NGD value roughly lies in between 0 and ∞, a low NGD value implies a high similarity between a pair of keywords, and vice versa. In addition, if the NGD value is greater than or equal to 1, then x and y are regarded as very dissimilar terms.

By considering a specific example, we attempt to assess the similarity degree between two keywords “recommendation system” and “collaborative filtering” using Eq. (4). A Google search for “recommendation system”, returned 195,000,000 hits, and the number of hits for inputting term “collaborative filtering” into Google is 12,600,000. Furthermore, a searching for pages in which both keywords “recommendation system” and “collaborative filtering" occur returned 7,230,000 hits. The maximum number of pages N is approximately 10^{10}. Therefore, we use Eq. (4) to calculate the similarity between these two terms on the basis of the aforementioned numbers, the computation result is approximately 0.493.

According to the introduction above, the NGD method can be applied to recommendation systems to measure item similarity. Huang, Chen, and Chen (2016) proposed a new idea from the WWW-based concept by using the Google search engine to access online information on items and computed their Google similarity based on NGD. This model can deal with the cold-start problem more effectively because it does not rely on the rating matrix data at all but only focuses on the online resource. Although this method effectively alleviates the sparsity problem, it first searches the online resources of items using Google, and then calculates the similarity between two items; hence, its computational efficiency is lower than that of other similarity methods that use the local rating information. Moreover, it cannot calculate the similarity in the offline condition, which causes the system to respond slowly to users.

In this study, we mainly focus on the local data to evaluate the similarity. To use the NGD method to handle the user ratings in the offline environment, we consider the entire recommendation system as a large search engine. Each item is regarded as a keyword searched by the system. The number of hits for a search term denotes the number of users who rated that item. In addition, we also consider the effects of three factors, namely user preference, absolute rating value, and the proportion of co-rated items, to adjust the NGD method. The adjusted model will be described in detail in Section 4.1.

2.3 *Intuitionistic fuzzy set*

During a decision-making process, people are usually uncertain when providing judgements on objective matters that are vague and complex. The concept of a fuzzy set was originally proposed by Zadeh (1965) to express people's uncertain decision information. However, the proposed fuzzy set considers only the membership degree and ignores the hesitation and the indeterminacy often involved in decision making (Xu & Zhao, 2016). To comprehensively reflect the three characteristics of human cognitive performance, namely affirmation, negation, and hesitation, Atanassov (1986) extended the fuzzy set concept by introducing an intuitionistic fuzzy set (IFS) in 1986. Let X be a fixed set; then, an intuitionistic fuzzy set (IFS) I on X is defined as:

$$I=\{\langle x,u_I(x),v_I(x)\rangle | x\in X\} \tag{5}$$

with the condition:

$$u_I(x)+v_I(x)\leq 1 \tag{6}$$

where $u_I(x)$ and $v_I(x)$ represent the membership degree and the non-membership degree of an element x, respectively. Usually, $\pi_I(x)=1-u_I(x)-v_I(x)$ is called the hesitancy (indeterminacy) degree of x to I (Xu & Yager, 2006), and $\alpha = (u_\alpha, v_\alpha)$ is an intuitionistic fuzzy number or value (IFN or IFV (Xu, 2013)) whose physical interpretation is defined as follows: For example, consider a class election with α = (0.6, 0.3), which implies that 60% of the votes are in favor of a monitor candidate, 30% are against him/her, and 10% abstain from voting. In addition, Atanassov & Gargov (1989) pointed out that IFSs are equipotent to interval-valued fuzzy sets (IVFS) (Zadeh, 1975), and that they are two types of equivalent extensions of fuzzy sets. They aim to make it possible to add uncertainty about the degree of truth. Stachowiak and Dyczkowski (2013) proposed an interval-valued similarity measure for incompletely known fuzzy sets to preserves information about the operands' uncertainty. Subsequently, Stachowiak, Zywica, and Dyczkowski et al. (2015) introduced an interval-valued fuzzy classifier based on an uncertainty-aware similarity measure to emphasize the role and importance of the uncertainty factor in making well-informed decisions.

Recommendation systems also provide imprecise decisions on unrated items for users according to their preferences on rated items. Therefore, the IFS can be applied to recommendation systems to make decisions for users. For instance, if (u_α, v_α) = (0.4, 0.3) can be explained as follows: in a movie evaluation system with a rating scale of 1-10, 40% of the user ratings of users on a movie are greater than 8, 30% ratings are less than 3, and the remaining 30% of the users did not provide a score. In IFS based recommendation systems, the membership degree denotes the percentage of users who like an item i, the non-membership degree represents the percentage of users who dislike an item i, and the hesitancy degree, reflecting the percentage of users who are not sure whether they like item i because, e.g., they did not watch the movie, represents an unknown.

In consideration of the importance of the membership degree and non-membership degree in IFS based recommendation systems, we expect to establish an appropriate item-based similarity method that calculates the similarity between two items from the perspective of user preference (like or dislike) percentage on rated items. Wang and Deng et al. (2017) first used the KL divergence (Kullback & Leibler, 1951) in information theory, i.e., the KL similarity, to measure item similarity by calculating the rating probability of an item. Their scheme makes full use of all the rating information instead of considering only the co-rated items. Moreover, it can distinguish different items effectively, especially when it is so difficult for geometrical distances to distinguish those overlapping items (Kullback & Leibler, 1951). Inspired by their approach, we propose an intuitionistic fuzzy set based KL similarity in Section 4.2 to evaluate the similarity between two items from the perspective of user preference probability.

3. Research framework

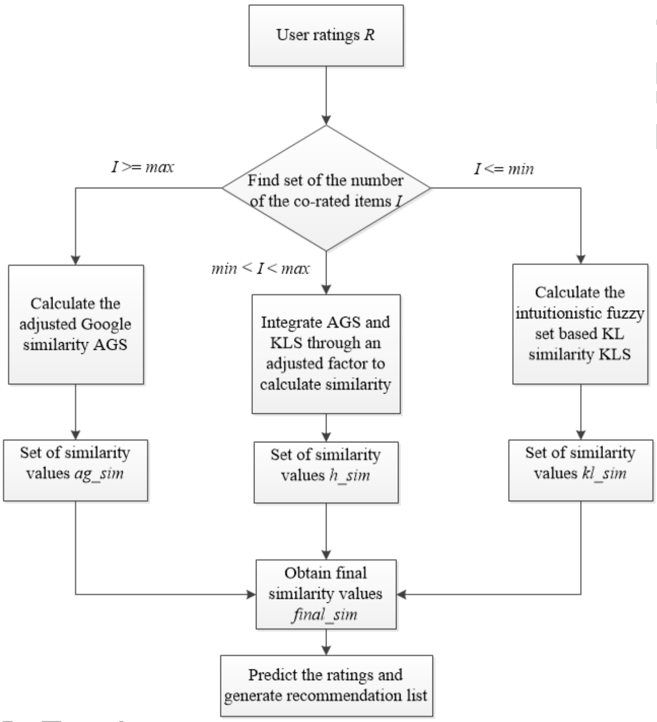


Figure 2. Framework of recommendation system procedure

Figure 2 shows the framework of recommendation system procedure using the proposed similarity model. As shown in Figure 2, the recommendation system first obtains a set of the number of co-rated items *I* from user ratings *R*. Then, our system selects different algorithms to evaluate item similarity under different numbers of co-rated items *I*, and the final similarity results are obtained. The selection of algorithms is as follows: 1) an adjusted Google similarity (AGS) is used to calculate the similarity values between two items when the number of co-rated items *I* is greater than or equal to a minimum threshold *max* that satisfies the condition of sufficient co-rated items; 2) an intuitionistic fuzzy set based KL similarity (KLS) is deployed to compute the similarity results when *I* is less than or equal to a maximum threshold *min* that leads to the cold-start problem; 3) AGS and KLS are integrated by an adjusted variable to comprehensively evaluate the similarity when I lies in the interval (min, max). The final similarity values are used to predict the ratings of active users on unrated items, and recommended lists are generated using top-N recommendation. In our system, the number of co-rated items is considered as a discriminant standard because it has a major impact on the similarity results. It is known that most of item similarity measures, such as cosine similarity, use only the co-rated items to compute the similarity and do not consider the non-co-rated items, which could be beneficial for increasing the operating efficiency. However, in sparse data, co-rated items are usually rare, which seriously affects the accuracy of the similarity results. Making full use of all the rating information can alleviate the data sparsity problem and improve the prediction accuracy effectively, but it ignores the problem of low operating efficiency of the system when all the ratings are used. Therefore, we propose a hybrid similarity model for the recommender system to achieve a trade-off between accuracy and efficiency, and achieving better recommendation results.

4. Proposed model

In this section, we first introduce an adjusted Google similarity (AGS) to rapidly and precisely calculate item similarity in the condition of sufficient co-rated items. Then, we present an intuitionistic fuzzy set based KL similarity (KLS) to effectively evaluate sthe imilarity results in the condition of rare co-rated items. Finally, we propose a hybrid similarity model through integrating AGS and KLS to comprehensively compute the similarity values when the number of co-rated items lies in a certain range.

4.1 *Adjusted Google similarity*

According to the introduction and analysis of the NGD method in Section 2.2, we apply it to the local database to measure the item similarity. The NGD between two items in CF is defined as

$$NGD(i,j)=\frac{\max\{\log f(i),\log f(j)\}-\log f(i,j)}{\log R-\min\{\log f(i),\log f(j)\}} \tag{7}$$

where function $f(i)$ returns the number of users who rated item i, and function $f(i, j)$ returns the number of users who rated items i and j; and R is the number of all possible ratings in the system.

However, Eq. (7) considers only the number of rated items; it ignores the preference of each user and the effect of the absolute rating value, which is highly likely to lead to the inaccurate similarity results. For example, let I = {1, 1, 1, 1, 1} and J = {5, 5, 5, 5, 5} be two rating vectors of items i and j, respectively. The rating scale is 1 through 5 and the average rating value of each user is 3. In this paper, we assume that if a rating of user u on item i is greater than his/her average rating, it means that user u likes item i or the user u has a positive preference on item i; otherwise, user u dislikes item i or user u has a negative preference on item i. From Eq. (7), the NGD distance between item i and j is 0. Thus, items i and j are exactly the same. Obviously, the result is unreliable and inaccurate because it ignores the influences of user preference and absolute rating values. Therefore, we divide the NGD method into four functions to overcome these problems. Each function reflects a positive (like) or negative (dislike) preference behavior of users on items. These functions are introduced as follows:

$$\begin{cases} NGD_1(i,j) = \dfrac{\max\{\log f_1(i), \log f_1(j)\} - \log f_1(i,j)}{\log R - \min\{\log f_1(i), \log f_1(j)\}} \\ NGD_2(i,j) = \dfrac{\max\{\log f_2(i), \log f_2(j)\} - \log f_2(i,j)}{\log R - \min\{\log f_2(i), \log f_2(j)\}} \\ NGD_3(i,j) = \dfrac{\max\{\log f_1(i), \log f_2(j)\} - \log f_3(i,j)}{\log R - \min\{\log f_1(i), \log f_2(j)\}} \\ NGD_4(i,j) = \dfrac{\max\{\log f_2(i), \log f_1(j)\} - \log f_4(i,j)}{\log R - \min\{\log f_2(i), \log f_1(j)\}} \end{cases} \tag{8}$$

where:

$$\begin{cases} f_1(p) = \sum_{u \in U_i} |r_{up} > \bar{r}_u|,\ f_2(p) = \sum_{u \in U_i} |r_{up} < \bar{r}_u|,\ p \in \{i, j\}, \\ f_1(i,j) = \sum_{u \in U_i \cap U_j} |r_{ui} > \bar{r}_u \ and\ r_{uj} > \bar{r}_u|,\ f_2(i,j) = \sum_{u \in U_i \cap U_j} |r_{ui} \le \bar{r}_u \ and\ r_{uj} \le \bar{r}_u|, \\ f_3(i,j) = \sum_{u \in U_i \cap U_j} |r_{ui} > \bar{r}_u \ and\ r_{uj} \le \bar{r}_u|,\ f_4(i,j) = \sum_{u \in U_i \cap U_j} |r_{ui} \le \bar{r}_u \ and\ r_{uj} > \bar{r}_u| \\ and \quad |\cdot| = 1 \end{cases}$$

where $f_1(i)$ is the number of rated item i satisfying the condition that rating r_{ui} of user u on item i is greater than the average rating $\bar{r}_u$ of user u, $f_1(i, j)$ is the number of co-rated items satisfying the condition that both r_{ui} and r_{uj} are greater than $\bar{r}_u$, and $\sum_{k=1}^{4} f_k(i,j) = f(i,j)$. It can be found from Eq. (8) that only the first function NGD_1 is a positive function that indicates a completely positive preference of each user on item i or/and item j, i.e., each user likes item i or/and item j in NGD_1, which has a positive influence on the similarity calculation. The other functions (NGD_2, NGD_3 and NGD_4) are negative functions because there exists at least one negative preference of users on items in these functions, which reduces the accuracy and reliability of the similarity results.

To eliminate the adverse impact of negative preference functions, we add a penalty function e^{-P} in Eq. (8), where P is a penalty factor. Therefore, the adjusted Google similarity is defined as

$$ag\_sim(i,j)=\left[\frac{1}{1+NGD_1(i,j)}\right]+e^{-P}\left[\sum_{k=2}^{4}\frac{1}{1+NGD_k}\right] \tag{9}$$

Eq. (9) involves only a positive function NGD_1 when P = 0. As the P value increases, it will gradually weaken the effect of negative functions. conversely, as the P value decreases, the effect of positive function will be gradually strengthened.

To emphasize the importance of the proportion of co-rated items, the penalty factor P is designed as follows:

$$P=\frac{\sum_{k=2}^{4} f_k(i,j)}{f(i)+f(j)}=\frac{f(i,j)-f_1(i,j)}{f(i)+f(j)} \tag{10}$$

4.2 *Intuitionistic fuzzy set based KL similarity*

Based on the discussion on IFS in Section 2.3, the concept of IFS can be applied to recommender systems to provide an uncertain decision-making for active users according to their previous preference behaviors on rated items. In the proposed system, the IFS focuses on three degrees of user rating preference: like, dislike, and unknown. The membership degree and the non-membership degree represent the percentage of users who like an item and the percentage of users who dislike an item, respectively. Therefore, we improve the KL similarity (Wang, Deng, Gao, & Zhang, 2017) to evaluate the similarity between two items from the perspective of user preference probability. Suppose that all the ratings on item i and j are two sequences. The IFS based KL distance is computed as

$$D(i\,\|\,j)_{IFS}=D(\hat{\rho}_i\,\|\,\hat{\rho}_j)=\sum_{pre\in\{l,d\}}\rho_{i,pre}\log_2\frac{\rho_{i,pre}}{\rho_{j,pre}} \tag{11}$$

where pre represents a user preference on an item, i.e., like l or dislike d, and $\rho_{i,pre}$ is the probability of user preference for item i, its formula is given by

$$\rho_{i,pre}=\begin{cases}\dfrac{\#l}{\#U} & \text{as } \rho_{i,l}\text{, if } r_{ui}>\bar{r}_u\\[2ex] \dfrac{\#d}{\#U} & \text{as } \rho_{i,d}\text{, otherwise}\end{cases} \tag{12}$$

where $\rho_{i,l}$ and $\rho_{i,d}$ represent the probability of users who like item i and the probability of users who dislike item i, respectively; $\#U$ is the number of users in the system, $\#l$ is the number of positive users, and $\#d$ is the number of negative users.

Thus, the IFS based KL similarity is used to compute the similarity between item i and j:

$$kl\_sim(i,j) = \frac{1}{1 + D(i \parallel j)_{IFS}} \tag{13}$$

In addition, it can be found from Eq. (11) that the KL distance between items i and j is asymmetric, i.e., $D(i \parallel j)_{IFS} \neq D(j \parallel i)_{IFS}$, which is useful for emphasizing the effects of asymmetry between a pair of items. For example, 80% of users like item i, and 20% of users like item j indicates that item j does not have a significant impact on item i, i.e., item j will probably not be a neighbor of item i when selecting the nearest neighbors of item i. By contrast, item i has a significant impact on item j. Therefore, the asymmetric mode of KL distance can effectively identify different effects between items and improve the prediction accuracy.

4.3 *Hybrid similarity model*

To fully integrate the advantages of both the adjusted Google similarity and the KL similarity, an adjusted variable λ is introduced to construct a hybrid similarity model. The proposed scheme can comprehensively calculate the item similarity. The hybrid similarity (h_sim) between item i and j is calculated as follows:

$$h\_sim(i,j) = \lambda * ag\_sim(i,j) + (1-\lambda) * kl\_sim(i,j) \tag{14}$$

with the condition that

$$\lambda = \begin{cases} 0 & , \quad I \le min \\ \dfrac{I - min}{max - min}, & min < I < max \\ 1 & , \quad I \ge max \end{cases} \tag{15}$$

where I is the number of co-rated items, min is the maximum threshold that results in the cold-start problem, max is the minimum threshold that satisfies the condition of sufficient co-rated items that we set in the system.

Further, variable λ is established to comprehensively consider the extent to which the two types of similarity methods affect the final similarity results for items i and j. When λ=1, the system uses only the adjusted Google similarity to recommend items. When λ=0, the system uses only the IFS based KL similarity to recommend items.

5. Experiments

5.1 *Operating environment*

The operating environment of our system is summarized below:

- Operating system: Windows 10
- CPU: Intel® Core^{TM} i5-8400
- Primary memory: 16GB RAM
- Development platform: PyCharm
- Development language: Python

5.2 *Dataset setup*

Two experimental datasets, MovieLens 1M (ML-1M) and Yahoo Music (YM), were employed to verify the performance of our proposed model. A brief description of these two datasets is provided in Table 1.

The sparsity level is defined as the percentage of all possible ratings available in the user rating matrix. A low sparsity level implies that only a few ratings are used in the system, providing little information for similarity calculation. To test the effectiveness of our proposed model, the following methodology was adopted to test the recommendation system (Cremonesi, Koren, & Turrin, 2010). Each of the datasets was divided into two parts: we randomly selected 80% of ratings from the dataset as the training set, and the remaining 20% of the data was used as the testing set.

Table 1. Description of the datasets used in the experiments

Name	**#Users** (m)	**#Items** (n)	**#Ratings** (r)	**Sparsity level** ($\kappa = \frac{r \times 100\%}{m \times n}$)	**Rating scale**
ML-1M	6,040	3,952	1,000,209	4.2%	{1,2,3,4,5}
YM	15,400	1,000	365,704	2.3%	{1,2,3,4,5}

5.3 *Evaluation indicators*

We employed two t evaluation indicators (Deng, Wang, Guo, Deng, Gao, & Park, 2018), namely prediction accuracy and recommendation accuracy to assess the performance of our system under different numbers of co-rated items.

The most commonly measures of prediction accuracy, namely *Mean Absolute Error* (*MAE*) and *Root Mean Squared Error* (*RMSE*), were deployed to indicate the difference between the actual ratings and the predicted ratings.

$$MAE = \frac{1}{m}\sum_{i=1}^{m}\left|r_i - p_i\right| \tag{16}$$

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(r_i - p_i)^2} \tag{17}$$

where m represents the number of predicted items and r_i and p_i are the actual rating in the testing set and the predicted rating of an active user on an item i, respectively. A lower prediction error indicates better prediction accuracy.

The recommendation accuracy involves three indicators, namely *Precision*, *Recall* and *F*1-*value*, given by

$$Precision = \frac{n(I_a \cap I_p)}{n(I_p)} \tag{18}$$

$$Recall = \frac{n(I_a \cap I_p)}{n(I_a)} \tag{19}$$

$$F1 - value = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{20}$$

where I_a and I_p represent the number of actual recommendation items and the number of predicted recommendation items, respectively. The *F*1-*value* is a comprehensive evaluation index that integrates *Precision* and *Recall* together. A larger *F*1 value indicates a better recommended result. In our experiments, we assumed that the recommended item should satisfy the condition that its rating value is greater than the average rating of the target user.

5.4 *Experimental results and analysis*

In our experiments, the number of co-rated items was treated as a baseline to select different similarity models. The variable λ in Eq.(15) is determined by the number of co-rated items I belonging to the range of some threshold to modify how the two types of similarity model affect the final similarity results. This enables us to comprehend the effects of different similarity methods on the performance of our proposed system. Table 2 summarizes the threshold settings on different datasets to assess system efficiency, where MAX, MIN and AVG denote the maximum value, minimum value, and average value in a set of I, respectively, R_I is the ratio of I to the total sum of I under a certain threshold. In each dataset, five numbers of co-rated items are presented as the minimum threshold and maximum threshold for the average co-rated items to estimate whether our system can work effectively under different sets of thresholds. A large threshold value implies that an enormous amount of information is available for computing the similarity results and vice versa. In addition, it is known that the number of the nearest neighbors *K*, has a significant influence on the prediction results. Therefore, the experimental results are investigated and compared under the conditions of different *K* values.

Table 2. Threshold settings for different datasets

Dataset	**MAX**	**MIN**	**AVG**	**Minimum threshold** (max)	**Ratio** (R_I)	**Maximum threshold** (min)	**Ratio** (R_I)
ML-1M	1507	1	23	40	16.0%	10	57.2%
				50	12.6%	20	71.9%
				60	10.1%		
YM	1640	1	9	20	9.0%	5	53.1%
				25	6.4%	10	78.8%
				30	4.8%		

5.4.1 *MovieLens* 1*M dataset*

Figure 3 illustrates the prediction accuracy of the proposed system on the ML-1M dataset. Note that the errors (*MAE* and *RMSE*) of the proposed scheme decreased as the number of the nearest neighbors, *K*, increased. The system achieved the lowest prediction errors of 0.7 (*MAE*; K = 160) and 0.893 (*RMSE*; K = 200) when min = 10 and max = 40, respectively. In addition, we found that the performance of our system is slightly degraded as the two thresholds are increased. The worst results of the proposed system were less than 0.714 (*MAE*) and 0.92 (*RMSE*), which indicates that our system is more suitable for the sparse environments.

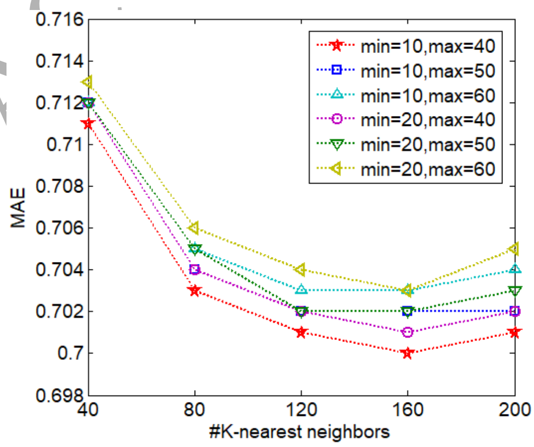
Figure 4 shows the recommendation accuracy of our system on the ML-1M dataset. It can be seen from Figure 4(a) that the difference in *Precision* between all the schemes was small, i.e., around 0.04. The lowest precision of the system was more than 0.697, and the best result was 0.701 when K = 160. As shown in Figure 4(b), the *Recall* of our scheme was in the interval (0.69, 0.708), which indicates better recommendation quality. An increase in *K* did not have a significant impact on the *Recall* of the system when $K \geq 120$. The $F1$ was calculated as shown in Figure 4(c). We can see that the $F1$ value of the system attained the highest accuracy of 0.705 when min = 10 and max = 40 at K = 120. Moreover, all the $F1$

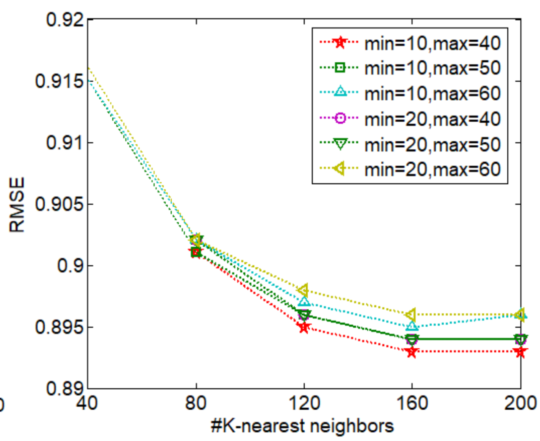
values exceeded 0.7 when K ≥ 120. Similarly, the system accuracy decreased as the thresholds increased, in particular, when the two types of thresholds reached the maximum values set by the system, the *F*1 value was the smallest.

5.4.2 *Yahoo Music dataset*

Similarly, all the schemes were executed on the YM dataset. The experimental results for the two types of evaluation indicators are shown in Figures 5 and 6. As shown in Figure 5, the prediction errors (*MAE* and *RMSE*) decreased as the number of nearest neighbors increased. The two plots clearly illustrate that the proposed scheme had the best *MAE* (0.986) and *RMSE* (1.251) when *min* = 5 and *max* = 20 at *K* = 200. Furthermore, the range of error fluctuation was rather small, indicating that our system is relatively stable and reliable for recommending items. Figure 6 shows the recommendation quality of the proposed system. It can be seen that the best results were obtained in the first scenario (*min* = 5 and *max* = 20). Moreover, the fluctuation of all the schemes in terms of *Precision* was greater than that in terms of *Recall* and *F*1. Although the recommendation accuracy on the YM dataset declined compared to that on the ML-1M dataset because of the sparser ratings of the former, the *F*1 value of the best plot was greater than 0.6 when K ≥ 60, indicating that our system shows satisfactory performance even under a relatively lower sparsity level.

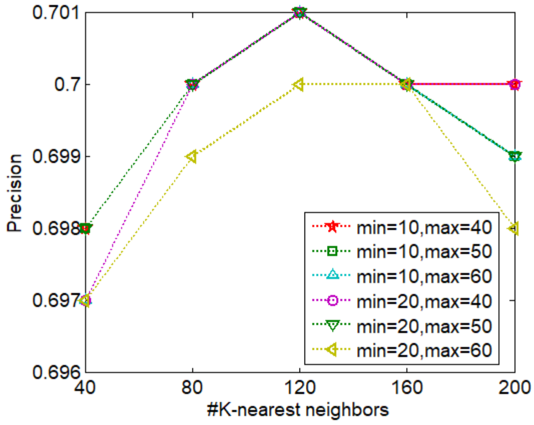
From our experimental analysis of the two above-mentioned datasets with different sparsity levels, we found that the proposed system can attain better results when it sets two types of thresholds (*min* and *max*) for the number of co-rated items; thus, our system is more sensitive to the sparse environments. In addition, the fluctuation of the system in terms of each evaluation indicator was small, indicating that our system is stable under various scenarios. Therefore, the experimental results verified that our proposed system performs well and provides high-quality recommendations.



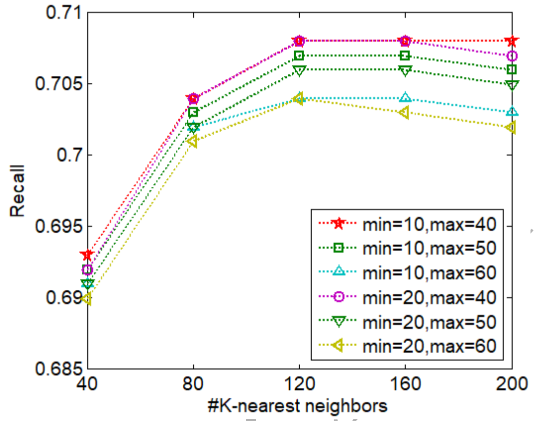


(a) (b)

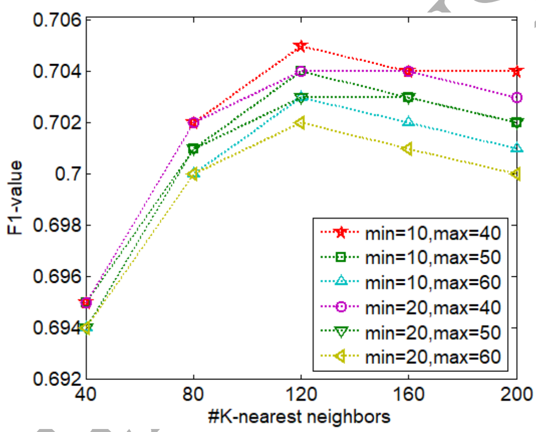
Figure 3. System efficiency on ML-1M (*MAE* and *RMSE*)



(a)

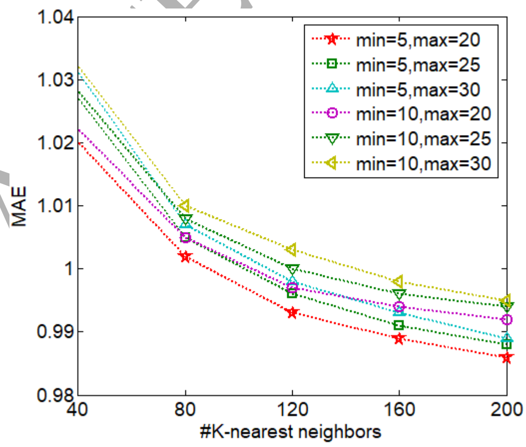


(b)

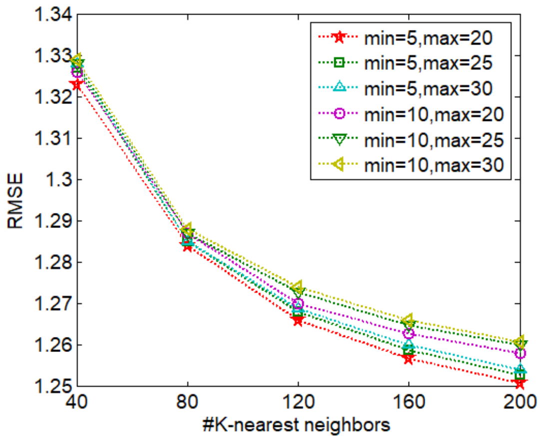


(c)

Figure 4. System efficiency on ML-1M (*Precision*, *Recall* and *F1- value*)

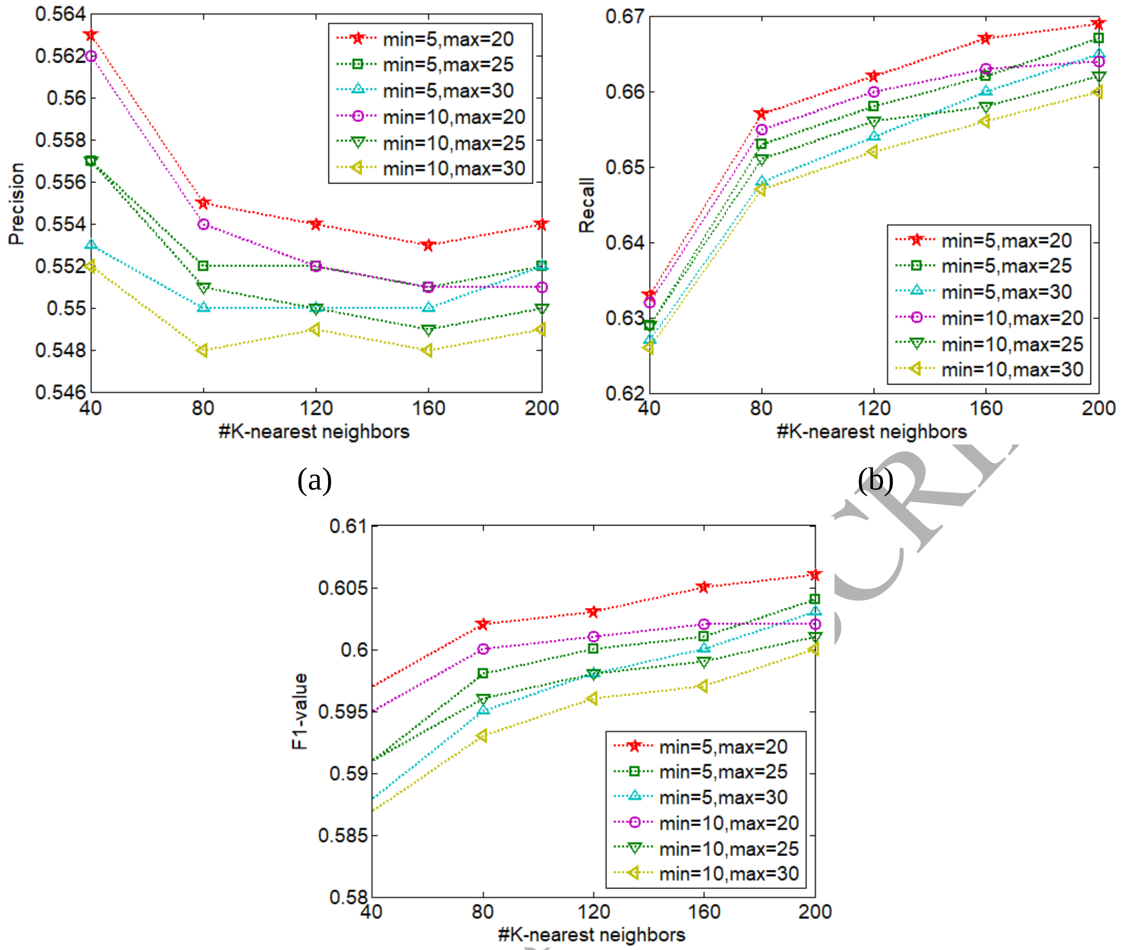


(a)



(b)

Figure 5. System efficiency on YM (*MAE* and *RMSE*)



(a) (b)

(c)

Figure 6. System efficiency on YM (*Precision*, *Recall* and *F*1-*value*)

5.5 *Comparative analysis*

In this section, we compared several state-of-the-art similarity measures to further verify the effectiveness of our proposed scheme in terms of the evaluation indicators described in Section 5.3 as well as the time complexity. The compared methods included NHSM (Liu, Hu, Mian, Tian, & Zhu, 2014), BCF (Patra, Launonen, Ollikainen, & Nandi, 2015) and KL-NHSM (Wang, Deng, Gao, & Zhang, 2017). The comparison experiments were implemented on the same datasets. The results are shown in Table 3. It can be seen from Table 3 that the proposed scheme obviously outperformed the compared methods in terms of *MAE* and *F*1-*value*. Furthermore, for *m* users and *n* items in a rating matrix, we can see that the time complexity of NHSM was the lowest because it considers only the co-rated items; however, it had the worst accuracy among all the methods. Although the BCF and KL-NHSM obtained relatively good recommendation results, they required a long time for similarity calculation because they use all the rating information. The proposed method not only achieved better accuracy but also exhibited acceptable complexity compared with the other methods. In summary, our scheme can effectively achieve a trade-off between accuracy and efficiency and thus improve the performance of recommendation systems.

Table 3. Comparison between results of our scheme and those of state-of-the-art methods

Dataset	**Method**	**MAE**	**F1-value**
ML-1M	NHSM	0.798	0.648
	BCF	0.778	0.667
	KL-NHSM	0.764	0.676
	Our scheme	**0.701**	**0.704**
YM	NHSM	1.097	0.527
	BCF	1.041	0.553
	KL-NHSM	1.028	0.577
	Our scheme	**0.986**	**0.606**
Time Complexity	NHSM	$O(3mn+2m)$	
	BCF	$O(16mn^2)$	
	KL-NHSM	$O(24mn^2+2m)$	
	Our scheme	$O(8mn+7n)$	

6. Conclusions and future work

To recommend items for target users rapidly and accurately, we design a hybrid item similarity model that achieves a trade-off between prediction accuracy and efficiency. Our proposed model involves three steps: (i) An adjusted Google similarity is introduced to rapidly and precisely calculate item similarity in the condition of sufficient co-rated items. This method considers not only the number of ratings but also the effects of user preference and the proportion of co-rated items into account; (ii) An intuitionistic fuzzy set based KL similarity is presented from the perspective of user preference probability to effectively compute item similarity in the condition of rare co-rated items. This approach can fully exploit all the rating information regardless of the number of co-rated items. In addition, it considers the rating preferences (like or dislike) of users as well as the rating uncertainty of users on unknown items; (iii) The two above-mentioned schemes are integrated by an adjusted variable to comprehensively evaluate the similarity values when the number of co-rated items lies in a certain range of value. Finally, the proposed system is implemented and tested under different thresholds of co-rated items. The experimental results on different sparse datasets indicate that the proposed system provides high-quality recommendations and achieves favorable system efficiency.

We explore two directions for future work. First, our proposed model only focuses on the rating values and does not consider the characteristics of items for calculating similarity. In general, the item properties have an important influence on decision-making when a target user chooses the items. To comprehensively assess the performance of the system, we will apply the item properties to our algorithm to further improve the recommendation quality in the future. Secondly, in this study, we emphasize the integration of similarity methods to evaluate the similarity according to their advantages. To recommend items for target users under various sparse environments, we will deploy a competitive algorithm that selects different similarity measures

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 71671121, 61472464), which is gratefully acknowledged.

References

Ahn, H. J. (2008). A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences, 178*(1), 37-51.

Arwar, B., Karypls, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International World Wide Web Conference* (pp. 285-295).

Atanassov, K. T. (1986). Intuitionistic fuzzy sets. *Fuzzy Sets and Systems, 20*(1), 87-96.

Atanassov, K. T., Gargov G. (1989). Interval-valued intuitionistic fuzzy sets [J]. *Fuzzy Sets and Systems*, 31, 343-349.

Baluja, S., Seth, R., Sivakumar, D., Jing, Y., Yagnik, J., Kumar, S., Ravichandran, D., & Aly, M. (2008). Video suggestion and discovery for YouTube: taking random walks through the view graph. In *Proceedings of the 17th International World Wide Web conference* (pp. 895–904).

Breese, J. S., Heckerman, D., & Kadie, C. (2013). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence* (pp. 43–52).

Brynjolfsson, E., Hu, Y., & Smith, M. D. (2003). Consumer surplus in the digital economy: estimating the value of increased product variety at online booksellers. *Management Science*, *49*(11), 1580-1596.

Chowdhury, G. (1983). Introduction to modern information retrieval. *McGrawpHill*.

Cilibrasi, R. L., & Vitanyi, P. M. B. (2007). The Google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, *19*(3), 370-383.

Cremonesi, P., Koren, Y., & Turrin, R. (2010). Performance of recommender algorithms on top-N recommendation tasks. In *Proceedings of the 4th ACM Conference on Recommender Systems* (pp. 39-46). ACM.

Davidson, J., Livingston, B., Sampath, D., Liebald, B., Liu, J., & Nandy, P. (2010). The YouTube video recommendation system. In *Proceedings of the 4th ACM Conference on Recommender Systems* (pp. 293). ACM.

Deng, J., Wang, Y., Guo, J., Deng, Y., Gao, J., & Park, Y. (2018). A similarity measure based on Kullback–Leibler divergence for collaborative filtering in sparse data. *Journal of Information Science*. doi:10.1177/0165551518808188.

Ekstrand, M. D., Riedl, J. T., & Konstan, J. A. (2007). Collaborative filtering recommender systems. *ACM Transactions on Information Systems*, *22*(1), 5-53.

Goldberg, D., Nichols, D., Oki, B.M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, *35*(12), 61–70.

Huang, C. K., Chen, Y. L., & Chen, M. C. (2016). A novel recommendation model with Google similarity. *Decision Support Systems*, *89*, 17-27.

Karypis, G. (2001). Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the 10th International Conference on Information and Knowledge Management Evaluation of Item-Based Top-N Recommendation Algorithm* (pp. 247-254).

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, *22*(1), 79-86.

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

Patra, B. K., Launonen, R., Ollikainen, V., & Nandi, S. (2014). Exploiting Bhattacharyya similarity measure to diminish user cold-start problem in sparse data. In *Proceedings of the 17th International Conference on Discovery Science* (pp. 252–263).

Patra, B. K., Launonen, R., Ollikainen, V., & Nandi, S. (2015). A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data. *Knowledge-Based Systems*, *82*(C), 163-177.

Pirasteh, P., Hwang, D., & Jung, J. E. (2015). Weighted similarity schemes for high scalability in user-based collaborative filtering. *Mobile Networks and Applications*, *20*(4), 497-507.

Shahabi, C., Banaei-Kashani, F., Chen, Y. S., & Mcleod, D. (2001). Yoda: An accurate and scalable web-based recommendation system. In *Proceedings of the 9th International Conference on Cooperative Information Systems* (pp. 418-432).

Stachowiak, A., Dyczkowski, K. (2013). A similarity measure with uncertainty for incompletely known fuzzy Sets. In *Proceedings of the IFSA World Congress NAFIPS Annual Meeting* (pp. 390-395).

Stachowiak, A., Żywica, P., Dyczkowski, K., Wójtowicz, A. (2015). An interval-valued fuzzy classifier based on an uncertainty-aware similarity measure. In *Intelligent Systems'2014* (pp. 741-752).

Takeuchi, Y., & Sugimoto, M. (2006). CityVoyager: An Outdoor Recommendation System Based on User Location History. *Ubiquitous Intelligence and Computing*. Springer Berlin Heidelberg.

Wang, Y., Deng, J., Gao, J., & Zhang, P. (2017). A hybrid user similarity model for collaborative filtering. *Information Sciences*, *418-419*, 102-118.

Xu, Z. (2013). Intuitionistic fuzzy aggregation operators. *IEEE Transactions on Fuzzy Systems*, *14*(1), 108-116.

Xu, Z., & Yager, R. R. (2006). Some geometric aggregation operators based on intuitionistic fuzzy sets. *International Journal of General Systems*, *35*(4), 417-433.

Xu, Z., & Zhao, N. (2016). Information fusion for intuitionistic fuzzy decision making: an overview. *Information Fusion*, *28*, 10-23.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, *8*(3), 338-353.

Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning–I. *Information Sciences*, 8(3), 199–249.

Zheng, N., Li, Q., Liao, S., & Zhang, L. (2010). Which photo groups should I choose? a comparative study of recommendation algorithms in Flickr. *Journal of Information Science*, *36*(6), 733-750.

Credit Author Statement

Guo Junpeng, Deng Jiangzhou, and Wang Yong participated in the design of this study. Guo contributed significantly to analysis and manuscript preparation. Deng conducted the experiments and drafted the manuscript. Wang performed the analysis with constructive discussions. All authors have read and approved the final manuscript.