Received August 14, 2019, accepted August 28, 2019, date of publication September 6, 2019,
date of current version October 17, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2939945

UTSP: User-Based Two-Step Recommendation With Popularity Normalization Towards Diversity and Novelty

KE NIU[1,2], XIANGYU ZHAO[3,4], FANGFANG LI[5], NING LI[1], XUEPING PENG[2], AND WEI CHEN[6]

[1]Computer School, Beijing Information Science and Technology University, Beijing 100101, China
[2]CAI, School of Computer Science, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia
[3]National Engineering Research Center for Information Technology in Agriculture, Beijing 100097, China
[4]Beijing Research Center for Information Technology in Agriculture, Beijing 100097, China
[5]oOh!Media Limited, North Sydney, NSW 2060, Australia
[6]Agricultural Information Institute, Chinese Academy of Agricultural Sciences, Beijing 100081, China

Corresponding authors: Ke Niu (niuke@bistu.edu.cn), Xiangyu Zhao (zhaoxy@nercita.org.cn), and Xueping Peng (xueping.peng@uts.edu.au)

This work was supported in part by the Beijing Education Commission Research Project of China under Grant KM201911232004, in part by the National Natural Science Foundation of China under Grant 61672105, and in part by the National Key Research and Development Program of China under Grant 2018YFB1004100.

ABSTRACT Information technologies such as e-commerce and e-news bring overloaded information as well as convenience to users, cooperatives and companies. Recommender system is a significant technology in solving this information overload problem. Due to the outstanding accuracy performance in top-N recommendation tasks, two-step recommendation algorithms are suitable to generate recommendations. However, their recommendation lists are biased towards popular items. In this paper, we propose a user based two-step recommendation algorithm with popularity normalization to improve recommendation diversity and novelty, as well as two evaluation metrics to measure diverse and novel performance. Experimental results demonstrate that our proposed approach significantly improves the diversity and novelty performance while still inheriting the advantage of two-step recommendation approaches on accuracy metrics.

INDEX TERMS Top-N recommendation, collaborative filtering, popularity normalization, two-step recommendation algorithm.

I. INTRODUCTION

Internet and online services, such as e-commerce, e-news, et al., bring great convenience as well as information overload problem. Many people have been often surrounded with a large number of products when shopping or reading news online, which sometimes even confuse people to decide which one to buy or read. Recommender system becomes inevitable and essential to help people choose wisely from the huge number of products using personalized recommendation technologies [1]–[5].

One popular approach of recommender system is collaborative filtering (CF) [6], [7]. The key of CF is to analyze the past interactions between users and items, and hence can be readily applied in various domains, without additional information required such as item features. Because of its simplicity, CF based recommendation has been widely applied in different applications and industries. Conventional CF approaches often consider recommendation as a rating prediction problem by analyzing user's explicit rating feedback, then recommend users the items with highest predicted rating score. For the cold start items that have not been rated, CF might predict a score with a heuristic learning method [8]–[10] or a machine learning model [11]–[14].

Intuitionally, more accurate rating prediction algorithm will produce better recommendation outcomes, that is why many researchers are working so hard to improve the rating prediction accuracy [12], [13]. However, what people really want from recommender system is actually the items they need [15] rather than the higher rated items. In addition, some studies demonstrate that the ratings actually are coupled together with very complicated relations rather than just linearly, as the accuracy of rating prediction is not always consistent with the ranking effectiveness [15]–[17].

The associate editor coordinating the review of this manuscript and approving it for publication was Min Jia.

Therefore, alternatively some researchers directly consider recommendation as a ranking problem [16], [18], [19] by modelling user preferences to rank items rather than predicting rating scores on individual items. It has been demonstrated that the ranking stream models outperform the rating prediction ones as reported if recommendation is considered as a ranking problem [16], [18]–[21].

It is arguable that the recommendation problem is considered as ranking or prediction challenge, as the most frequently used rating data is not able to fully capture user behaviors. Typically a rating actually embeds two sorts of user behaviors: (1) a user selects an item to rate, and (2) rate the item with a value. It won't be effective enough by simply using rating or ranking prediction to generate recommendations for this circumstance, because a user may simply rate an item with any values predicted by recommendation algorithms.

To fully capture the above two user behaviors, Hofmann [22] decomposes the recommendation problem into two steps: (1) predict the items to select, (2) predict the rating given the selected items. This two step prediction process actually mimics a scenario where users are free to select items out of their interests and rate them accordingly. We follow the same two-step recommendation strategy since it simulates the generation of user behaviors, and have proposed a few inter two-step recommendation approaches, which are different from the Hofmann's intra two-step recommendation approach by combining two separate models to process each step [17], [20], [21]. Because of better simulation of user behaviors, this two-step recommendation strategy improves the accuracy of the recommendation towards conventional ones. However, the two-step methods are still not innovative and diverse enough, because these models are possibly biased on well-known items to users. In this case, these recommendation results mean little to users although they are accurate. People generally need the information that they did not know, which will be really valuable to them.

Let's take a toy example in agricultural e-commerce domain to illustrate our motivation. A compound fertilizer is a well-known item as a generic fertilizer for all crops, and it may be recommended to vineyard owners though they may have already known about it. However, the bordeaux mixture is a better recommendation since it is a fungicide to prevent grapes from infestations of downy mildew, powdery mildew and other fungi. This kind of recommendations is more acceptable since it is not only an accurate item but also a novel one. As a result, diversity and novelty factors are also important to recommender system in addition to accuracy. Some studies have pointed out that one goal of recommender system is to provide users with highly idiosyncratic or personalized items, and more diverse recommendations will be likely to recommend more satisfied items to users [23]. More and more attentions have been paid on recommendation diversity and novelty [24]–[31]. The ACM conference on Recommender Systems actually held an independent session ''Diversity, Novelty and Serendipity'' in 2014.

Due to the poor performance of the previous two-step recommendation approaches on diversity and novelty, this paper aims to solve this problem and recommend more diverse and novel items while maintaining the advantages on accuracy metric. In this paper, we propose a user-based two-step recommendation algorithm with popularity normalization (UTSP) to consider item importance according to their popularity with both similarity calculation and probability prediction. In addition, there are two other innovations in this paper. Firstly, to evaluate the effectiveness of recommendation diversity and novelty, we propose two new evaluation metrics (*HitCOV* and *HitCIL*) based on two typical metrics: coverage and coverage in long tail. Secondly, we propose an improved Jaccard similarity function (IJ) combined with popularity normalization to further improve the model performance, especially on *HitCOV* and *HitCIL*. The improved IJ function is actually helpful to recommend more diverse items meeting user's real interests.

The remainder of the paper is organized as follows. We first introduce diversity and novelty challenges in recommender system, and propose two metrics to measure them. In the Recommendation algorithm Section, two-step recommendation algorithms, user-based two-step recommendation algorithms and similarity functions are introduced step by step, which are the key parts in UTSP method. Experiments are conducted on MovieLens dataset to compare the proposed approach with baselines, as well as the discussion of experiment results in Experiment and discussion Section, followed by the Conclusion Section.

II. DIVERSITY AND NOVELTY

Diversity and novelty have been grabbing more and more attention in the Recommender System community as key recommendation quality factors beyond accuracy in real recommendation scenarios [23]–[34]. Many different diversity and novelty metrics have already been proposed in these studies.

In [32], the novelty of recommendations is considered that how different it is with respect to ''what has been previously seen'', by a specific user. This means that whether the recommendations are novel or not depends on individual opinions which are difficult to be measured. Diversity generally applies to a set of items, and is related to how different the items are with each other. A diverse recommendation set is also related to novelty, for example, each item is ''novel'' with respect to the rest of the set.

There are two kinds of diversity measures, individual diversity and aggregate diversity [23]. Individual diversity is defined as the diversity of recommendation lists for a given user, which are often measured by an average dissimilarity between all pairs of recommended items. On the contrary, aggregate diversity considers recommendations across all users. Therefore, it can be easily measured by the coverage of recommendations across all users. It should be noticed that there is no trivial relationship between individual diversity and aggregate diversity. For example, if the system recommends to all users the same five best-selling items that are not similar to each other, the recommendation list for each user is diverse (i.e., high individual diversity), but only five distinct items are recommended to all users (i.e., resulting in low aggregate diversity) [23]. Based on the analysis, aggregate diversity is a more important problem in our opinion, though significant amount of work has been done on improving individual diversity [25], [30], [31], [33], [34]. Therefore, this paper mainly focuses on improving aggregate diversity (which we will simply refer to as diversity throughout the paper, unless explicitly specified otherwise) which has been largely untouched.

In addition, making a diverse or novel set of recommendations is easy. However, it is difficult to ensure that this set contains many items that are relevant to the user preference. The diverse, novel and accurate recommendation list will be more reasonable, since the purpose of recommender system is inherently linked to a notion of discovery. This is exactly the purpose of this paper—improving the diversity and novelty performance which are the weakness of the two-step recommendation approaches while maintaining their high accuracy advantages.

To evaluate modelling performance, we proposed two new evaluation metrics *HitCOV* and *HitCIL* based on two typical metrics: coverage (*COV*) and coverage in long tail (*CIL*). *COV* is one of the most popular diversity metrics. It measures the coverage or percentage of the recommended items across the entire items. The N-dependent *COV* is defined as:

$$COV(N) = \frac{|\bigcup_u TopN(u)|}{|I|} \tag{1}$$

where I represents the entire item set, and $TopN(u)$ is the recommendation result for user u in top-N recommendation task. In addition to *COV*, *CIL* indicates novelty to a certain degree by measuring recommendation coverage in the long tail of the items. It is defined as:

$$CIL(N) = \frac{|Long \cap \bigcup_u TopN(u)|}{|I|} \tag{2}$$

where *Long* represents the long tail item set. In this paper, the long tail item set consists of the rest of top 20% popular items.

It can be easily found that *COV* and *CIL* cannot evaluate whether the recommendations are effective, which just indicates how many different items can be shown to users. In order to measure the effectiveness of recommendation results, the distinct item set which contains all the items that are recommended to a user and meet the user's preference in top-N recommendation task is defined as *Hit*, it can be written as:

$$Hit(N) = \bigcup_u (Pre(u) \cap TopN(u)) \tag{3}$$

where $Pre(u)$ is the item set that meets the preference of user u. Based on $Hit(N)$, *HitCOV* and *HitCIL* are proposed to evaluate the effectiveness of recommendation results on diversity and novelty. They are defined as:

$$HitCOV(N) = \frac{|Hit(N)|}{|I|} \tag{4}$$

$$HitCIL(N) = \frac{|Hit(N) \cap Long|}{|I|} \tag{5}$$

These four metrics (*COV*, *CIL*, *HitCOV*, and *HitCIL*) will be used to evaluate the performance of our proposed recommendation approaches comparing with benchmark ones in the experiment section.

III. RECOMMENDATION ALGORITHM

A. TWO-STEP RECOMMENDATION ALGORITHM

The typical CF recommendation algorithm is based on user's ratings. As mentioned in our previous work [17], [20], [21], user ratings data actually embed two sorts of user behaviors: (1) user selects an item to rate, and (2) rate the selected item. However, the traditional recommendation algorithms normally try to predict or rank ratings directly on rating values or rating ordinal relation, but ignore the first item selection behavior. These algorithms normally assume that if users rate an item, the rating value could be predicted. Unfortunately, this assumption may not be always true as some users may not tend to rate an item which is out of their interest.

To solve the above issue, we have proposed the two-step recommendation algorithms by considering the two user behaviors embedded in ratings in our previous work [17], [20], [21], as shown in Fig. 1. In a two-step recommendation algorithm, the unknown user behaviors can be predicted as the two steps are actually a simulation of user ratings. The first step of selecting items can be predicted by the probability $\hat{P}(u, i)$ that user u rates item i, then the second step of rating the selected item is to predict the value $\hat{r}(u, i)$ which u may rate item i. After that, the ranking score can be computed as:

$$ranking(u, i) = \hat{P}(u, i)\hat{r}(u, i) \tag{6}$$

The goal of the first step is to predict the rating behaviors. Intuitively, historical rating behaviors are relevant to it, whereas rating values are not. Therefore, the probability is predicted using only rating behaviors in the first step of our proposed framework. In the second step, all users' historical rating data (both rating behaviors and rating values) are used to predict unknown ratings. As this is a classic rating prediction problem, therefore, existing techniques focusing on rating prediction can be used in this step. After the two-step calculation, the ranking score can be computed with Eq (6). The recommendation results can be generated based on the rankings, that is, the items with top-N ranking values will be recommended to the target user.

B. USER-BASED TWO-STEP RECOMMENDATION ALGORITHM

It has been demonstrated that these two-step recommendation algorithms gained good performance in top-N recommendation task. However, these algorithms may reduce

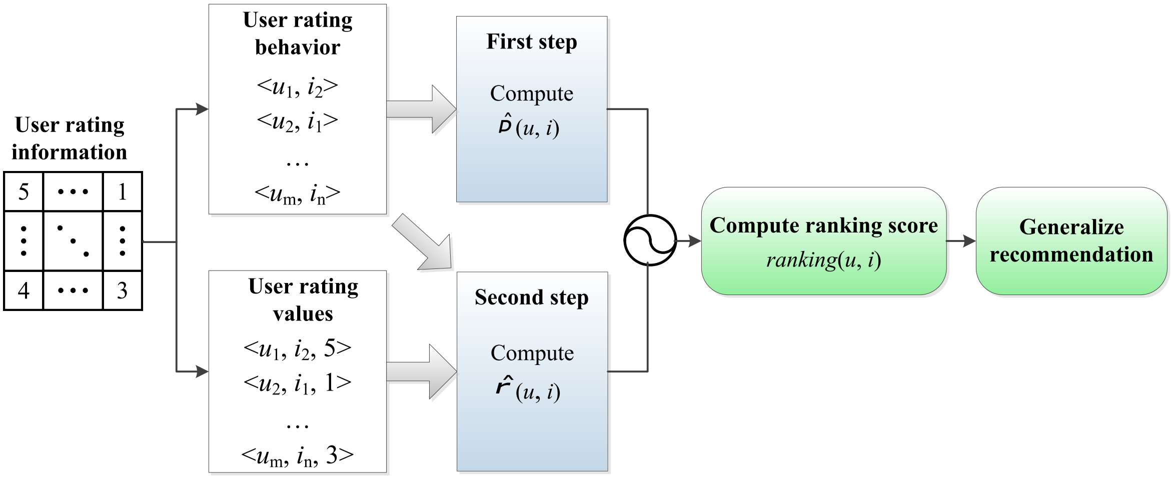


FIGURE 1. The two-step recommendation framework.

recommendation aggregate diversity [21], which mismatch the original purpose of recommender systems since they are explored to solve information overload problem for users. This problem is called "Harry Potter Problem" [35], [36]. During many years, Harry Potter is a runaway bestseller, thus too frequently been recommended to users whenever or whatever books they are reading. This "Harry Potter Problem" clearly indicates that the recommended items are biased on popular and well-known items. Furthermore, one fact is that more popular items typically have more ratings, but idiosyncratic items might have limited ratings. An item with more ratings actually means easier to be recommended to more users, which can partly explain the problem. In order to solve the problem in two-step recommendation algorithms, we propose a user-based two-step recommendation algorithm with popularity normalization (UTSP) in this section.

The target of UTSP's **first step** is to predict the probability that a user rates an item with user's historical rating behaviors. The rating behaviors are binary data, hence a user can be described as an n-dimensional vector in which 1 represents rated items and 0 represents unrated ones, which can be written as:

$$V_U(u) = (v_1, v_2, \cdots, v_n) \tag{7}$$

$$v_i = \begin{cases} 1, & i \in I(u) \\ 0, & i \notin I(u) \end{cases} \quad (i \in [1, n]) \tag{8}$$

where $I(u)$ represents all the items rated by user u.

Conventional user-based two-step recommendation algorithm (UTS) directly use this method to predict the probability that a user would like to rate an item. If we don't consider user similarity, the probability can be easily calculated as:

$$\hat{P}(u, i) = \frac{1}{|N(u)|} \sum_{a \in N(u)} V_U(a)[i] \tag{9}$$

where $V_U(a)[i]$ is the i^{th} element of the binary user model for user a, and $N(u)$ consists of the most similar neighbor users of user u.

This probability represents how likely the neighbors rated an item for a given user. Intuitively, this approach is biased on popular items with more ratings. Let's take a toy example in book domain to illustrate this bias: Harry Potter verse Data Mining [21]. Harry Potter is a very popular book, more than 20% users actually have bought this book, while less than 0.3% users only bought the professional computer science book Data Mining. Therefore in this bias situation, for a given user a, 10 users out of the 50 neighbors actually have bought the popular book Harry Potter, but only 5 neighbors bought the book Data Mining. If directly applying Eq (9) for recommendation, user a will get a recommendation book Harry Potter. However, the book Data Mining might be a better recommendation because the neighbor' purchase rate across all users on this book are actually much higher than book Harry Potter and the overall rate, which implies that this user might be a computer science researcher. This toy example indicates that the increment of the purchase rate in a user's neighborhood is significant for a good recommendation, which can be calculated as:

$$\hat{P}(u, i) = \frac{\sum_{a \in N(u)} V_U(a)[i]/|N(u)|}{|U(i)|/|U|} \tag{10}$$

where U represents the entire user set, and $U(i)$ represents the subset of users who have rated item i. Note that from this equation, the increased purchase rate might be greater than 1, which means we'll need a normalized step. In Eq. 10, $|N(u)|$ and $|U|$ are just constants for a given user, therefore these two constants actually will not affect the item ranking if we delete them from the equation. Thus this equation can be further simplified to a normalized version as:

$$\hat{P}(u, i) = \frac{\sum_{a \in N(u)} V_U(a)[i]}{|U(i)|} \tag{11}$$

Eq (11) is actually a normalized version with popularity as $U(i)$ is the popularity for item i. In addition to normalization, the user attributes from neighbors are also very important for a recommendation. Therefore, the equation can be further improved by including user similarities as:

$$\hat{P}(u,i) = \frac{\sum_{a \in N(u)} sim(u,a) \cdot V_U(a)[i]}{|U(i)| \cdot \sum_{a \in N(u)} sim(u,a)} \tag{12}$$

where $sim(u, a)$ is the similarity between user u and user a.

In theory, Eq. 12 should be effective to estimate how likely a user rate an item. However, according to our previous experiments, the recommendations from Eq (12) might be biased towards long tail items. Let's review the book domain toy example Harry Potter vs Data Mining again. Assume only one neighbor for a given user has bought book Data Mining, Eq. 12 is likely to recommend this book to this user since this book is less popular than Harry Potter. This recommendation result is actually biased and experiences individual long tail interest, rather than considering the common interests of the whole neighbor set. In order to decrease the long tail interest bias, the prediction equation can be further updated through an improved popularity normalization, which can be revised as:

$$\hat{P}(u,i) = \frac{\sum_{a \in N(u)} sim(u,a) \cdot V_U(a)[i]}{\beta \cdot \sqrt{|U(i)|} \cdot \sum_{a \in N(u)} sim(u,a)} \tag{13}$$

where β is a small constant to make sure the probability is between 0 and 1.

The **second step** is considered as a classic rating prediction problem. It can be done by making use of existing techniques. In UTSP, we use SVD++ [12] in the second step.

As a popular matrix factorization approach, SVD++ is capable to consider explicit rating and implicit feedbacks for a superior recommendation model by optimizing a pre-defined objective function. The prediction model and training strategy of SVD++ is detailed in [12], which won't be further explained in this paper.

Based on the above models, UTSP can predict $\hat{P}(u, i)$ according to Eq (13), then predict $\hat{r}(u, i)$ using SVD++ [12], [21], followed by ranking the unrated items for users according to Eq (6) to produce recommendation outcomes.

C. SIMILARITY FUNCTIONS

Similarity function is an important part in collaborative filtering approaches, which has not been discussed yet. There are two typical similarity functions, correlation and relevance. According to the classic rating-based recommendation task, some studies [6], [9], [11], [17] believe that different rating scores represent different degrees of user's attitude towards items. Therefore, users with similar rating values to the same item are often considered to be similar. This type of similarity functions is called correlation. Another type of similarity function is called relevance [37], which considers the users who often rate the same items are similar.

Arguably, correlation is often considered as a better similarity function since it utilizes more information. However, relevance has also been demonstrated as a great similarity function than correlation especially in the first step of two-step recommendation algorithms [17], [38]. Among the relevance similarity methods, Jaccard is a popular similarity function which can be directly applied in the UTSP algorithm. The Jaccard similarity function can be defined as:

$$sim_{\text{Jaccard}}(u,a) = \frac{|I(u) \cap I(a)|}{|I(u) \cup I(a)|} = \frac{\sum_i V_U(u)[i] \wedge V_U(a)[i]}{\sum_i V_U(u)[i] \vee V_U(a)[i]} \tag{14}$$

This Jaccard similarity function treats all items equally. However, as we mentioned before, user's behaviors on rating items are biased according to the item popularity. Therefore, we incorporate popularity normalization to the Jaccard function to increase recommendation diversity. The improved Jaccard (IJ) similarity function can be defined as:

$$sim_{\text{IJ}}(u,a) = \frac{\sum_i (V_U(u)[i] \wedge V_U(a)[i])/|U(i)|}{\sum_i (V_U(u)[i] \vee V_U(a)[i])/|U(i)|} \tag{15}$$

From this improved version, it is clearly seen that items with different popularity would play different roles in similarity calculation, the impact of less popular items would be bigger than the popular ones. The effectiveness of this improved Jaccard similarity function will be discussed in Experiment and discussion Section.

IV. EXPERIMENT AND DISCUSSION

A. EXPERIMENT SETUP

In the experiment, we aim to evaluate modelling performance in terms of accuracy, diversity and novelty for our proposed model in top-N recommendation task using 6 metrics. The Normalized Discounted Cumulative Gain ($NDCG$) [39] and 1-*call* [40] are used as accuracy metrics, whereas *COV* and *HitCOV* are for diversity evaluation, and *CIL* and *HitCIL* are mainly for novelty.

The data sets to evaluate our proposed recommendation approach are MovieLens 100K and 1M.[1] MovieLens 100K includes 100,000 ratings with 1-to-5 star scale assigned by 943 users on 1,682 movies, and MovieLens 1M includes 1,000,209 ratings with 1-to-5 star scale assigned by 6,040 users on 3,900 movies. To make sure the stable experiment result, we also apply 5-fold cross validation for our evaluation. Basically, we first split the initial data set to 5 equal sized subset, then randomly assign 4-fold as training set and the rest fold as test set. The recommendation algorithms will apply user's rating behaviors in the training set to train models, then to test their accuracy, diversity and novelty metrics based on test data set.

The conducted experiments include two parts. One is to compare the performance between conventional user-based two-step recommendation algorithm (UTS) and the proposed improved algorithms UTSP. The differences among the approaches are similarity functions and prediction methods to estimate the probability of a user rating an item. Three UTSP variants with different similarity functions will be discussed. The approach using Eq (12) and Eq (15) is

[1] https://grouplens.org/datasets/movielens/

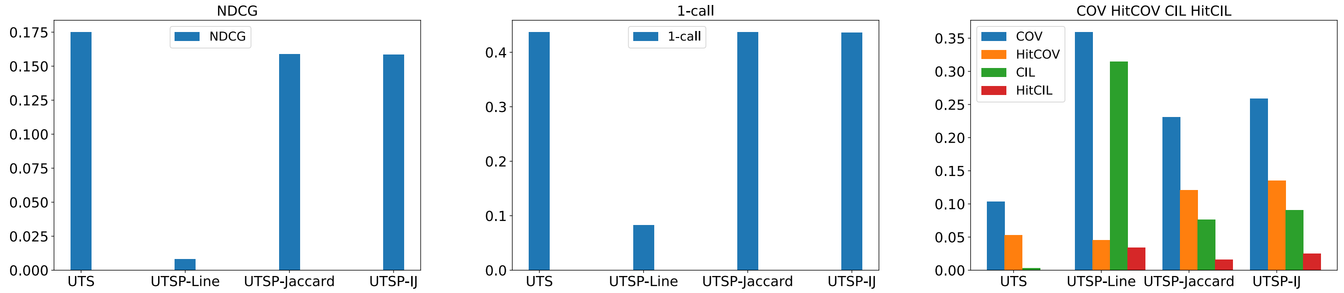


FIGURE 2. Performance of two-step recommendation approaches.

TABLE 1. Performance of two-step recommendation approaches.

	NDCG	1-call	COV	HitCOV	CIL	HitCIL
UTS	**0.1750**	**0.4369**	0.1034	0.0529	0.0036	0.0000
UTSP-Line	0.0084	0.0827	**0.3591**	0.0458	**0.3151**	**0.0339**
UTSP-Jaccard	0.1589	**0.4369**	0.2313	0.1207	0.0767	0.0161
UTSP-IJ	0.1587	0.4358	0.2592	**0.1356**	0.0910	0.0250

marked as UTSP-Line, the one using Eq (13) and Eq (14) is marked as UTSP-Jaccard, while the one using Eq (13) and Eq (15) is marked as UTSP-IJ. These three approaches will be compared with UTS to demonstrate the effectiveness of our proposed approach. The second is to compare our methods with the benchmark models on both rating and ranking prediction, for example, UserCF [9] and SVD++ [12] for rating prediction purpose, and pLPA [16] for ranking prediction purpose. Among these models, UserCF is a user-based CF with Jaccard similarity, and SVD++ is a state-of-the-art rating prediction model. For ranking prediction methods, pLPA [16] is a probabilistic latent preference analysis approach directly optimizing ranking target based on a pairwise ordinal model.

To easily reproduce the evaluations, we also detail the model parameters used in this paper, which include:

- the size of nearest neighbors for UserCF is 50;
- SVD++ model with 50 features and 25 iterations with $\lambda_6 = \lambda_7 = 0.05$, and $\gamma_1 = \gamma_2 = 0.002$;
- pLPA has 6 latent preferences and 30 iterations [16];
- UTS and UTSPs have the same neighbor size parameter setting as UserCF model for first setp, and same settings as SVD++ for second step.

All the experiments conducted in this Section are evaluated by metrics *NDCG*, 1-*call*, *COV*, *HitCOV*, *CIL* and *HitCIL*.

B. RESULTS AND DISCUSSION

1) COMPARISON WITH TWO-STEP RECOMMENDATION APPROACHES

Firstly, we present a performance comparison among two-step recommendation approaches. For each approach, we report *NDCG*, 1-*call*, *COV*, *HitCOV*, *CIL* and *HitCIL* at the 5*th* position in the recommendation list. Table 1 illustrates the results based on MovieLens 100K dataset. The bold cells indicate the best results for the corresponding metrics.

As can be seen from Table 1, UTS gets the best accuracy and the worst diversity. All the UTSP approaches gain better diversity than UTS. This indicates that the popularity normalization can lead to significant diversity improvement. However, UTSP-Line does not maintain the accuracy advantage of two-step recommendation algorithms. It is because that directly using Eq (12) to predict probability that a user rates an item causes the recommendation list to be biased towards long tail interests from individual neighbors. This can be further demonstrated by the evidence that most recommended items (about 88%) of UTSP-Line are long tail ones.

Focusing on the conventional diversity metrics *COV* and *CIL*, both USTP-Jaccard and UTSP-IJ gain significant improvement of diversity with at least 124% on *COV* and 2050% on *CIL*, while maintaining the accuracy advantage of UTS with no more than 10% loss on *NDCG* and 1% on 1-*call*. Moreover, as shown in Fig. 2, the accuracy performance of UTSP-Jaccard and UTSP-IJ are almost the same, while IJ similarity function can further lead to about 12% improvement compared to Jaccard on *COV* and 19% on *CIL*, which demonstrates the effectiveness of the proposed popularity normalization on similarity calculation. *HitCOV* and *HitCIL* are two novel diversity metrics which can evaluate whether the diverse recommendation is effective. Comparing the performance on *COV* and *HitCOV*, UTSP-IJ and UTSP-Jaccard gain better *HitCOV* and worse *COV* than UTSP-Line. This indicates that high *COV* is not always effective. Though UTSP-Line can generate more diverse recommendations, users may hardly like them. On the contrary, the diverse recommendations from UTSP-IJ and UTSP-Jaccard are much more effective. UTSP-IJ gains the best performance on *HitCOV*, and gets good performance on *HitCIL* close to UTSP-Line, which is biased towards long tail interests.

Generally speaking, UTSP-IJ outperforms other three two-step recommendation approaches if considering both accuracy and diversity performance comprehensively, which demonstrates the effectiveness of our proposed UTSP algorithm on both probability prediction and similarity calculation.

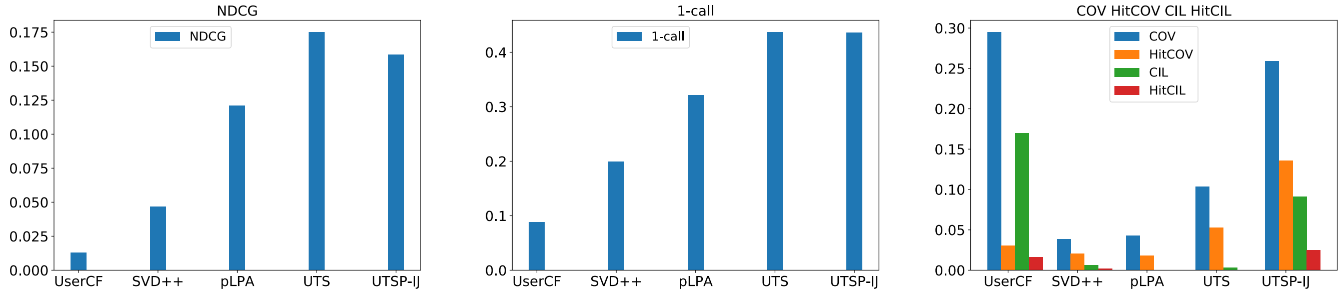


FIGURE 3. Performance compared with benchmark recommendation approaches (MovieLens 100K).

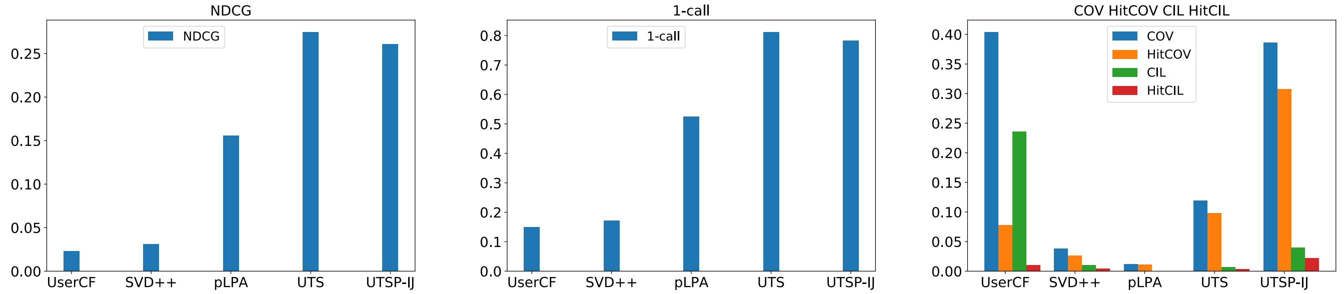


FIGURE 4. Performance compared with benchmark recommendation approaches (MovieLens 1M).

TABLE 2. Performance compared with benchmark recommendation approaches (MovieLens 100K).

	NDCG	1-call	COV	HitCOV	CIL	HitCIL
UserCF	0.0131	0.0880	**0.2949**	0.0303	**0.1700**	**0.0166**
SVD++	0.0468	0.1994	0.0386	0.0208	0.0065	0.0018
pLPA	0.1211	0.3213	0.0428	0.0184	0.0000	0.0000
UTS	**0.1750**	**0.4369**	0.1034	**0.0529**	0.0036	0.0000
UTSP-IJ	**0.1587**	**0.4358**	**0.2592**	**0.1356**	**0.0910**	**0.0250**

TABLE 3. Performance compared with benchmark recommendation approaches (MovieLens 1M).

	NDCG	1-call	COV	HitCOV	CIL	HitCIL
UserCF	0.0233	0.1492	**0.4040**	0.0783	**0.2363**	**0.0103**
SVD++	0.0311	0.1724	0.0385	0.0263	0.0100	0.0045
pLPA	0.1559	0.5253	0.0120	0.0115	0.0000	0.0000
UTS	**0.2747**	**0.8118**	0.1195	**0.0980**	0.0073	0.0038
UTSP-IJ	**0.2610**	**0.7833**	**0.3866**	**0.3075**	**0.0402**	**0.0219**

2) COMPARISON WITH BENCHMARK RECOMMENDATION

To further demonstrate the effectiveness and robustness, UTSP-IJ is also compared with the benchmark models such as UserCF, SVD++ and pLPA with metrics *NDCG*, 1-*call*, *COV*, *HitCOV*, *CIL* and *HitCIL* on both data sets MovieLens 100K and 1M. We detail the comparison results in Table 2 and Table 3, where we can clearly see the highlighted top 2 best performed methods.

As depicted from Fig. 3 and Fig. 4, the two rating prediction models UserCF and SVD++ perform worse than our proposed models in terms of accuracy evaluation metric. The result indicates that the metric accuracy of rating prediction is probably not closely relevant to the quality of top-*N* recommendation. While the ranking prediction recommendation approach pLPA indeed improves the recommendation accuracy, which proves the statement of recommendation challenge is more likely to be a ranking prediction issue. In addition, our proposed two-step recommendation approaches UTS and UTSP-IJ further improve the recommendation accuracy to outperform the benchmark ones, which shows that the two-step strategy is feasible for top-*N* recommendation task.

In terms of diversity metrics, model UTS is almost the worst on *COV* but with the 2^{nd} best performance on *HitCOV*. It means that although the recommendation diversity of UTS is not good, the diverse recommendations can always meet user interests. In addition, the popularity normalized model UTSP-IJ significantly improves the diversity performance. In terms of metrics *HitCOV* and *HitCIL*, UTSP-IJ is able to recommend the most diverse items. All the above experimental comparisons clearly outline the superiority of our proposed model UTSP-IJ which actually outperforms all the benchmark models both on accuracy and diversity metrics.

3) ABLATION ANALYSIS

We performed a detailed ablation study to examine the contributions of the proposed model components for recommendation performance. There are three replaceable components in this algorithm:

- **UserCF:** one step recommendation algorithm which map to the first step in UTS model;
- **SVD++:** one step recommendation algorithm which map to the second step in UTS model;
- **UTS:** a conventional user-based two-step recommendation algorithm;
- **UTSP-IJ:** our proposed two-step recommendation algorithm.

From Table 2 and Table 3, we find that the UTSP-IJ algorithm obtains the balanced performance of accuracy, diversity and novelty compared to the ablated models on two data sets. Moreover, we note that UTS can effectively improve the recommendation accuracy, but it reduces diversity. For this problem, we introduce an improved user-based two-step recommendation algorithm with popularity normalization UTSP-IJ, which not only maintains high accuracy but also significantly improves diversity.

In particular, we can see that UTS has the best accuracy performance on both data sets. Compared with UserCF, UTS gains significant improvement of accuracy with at least 1079% on *NDCG* and 396% on 1-*call*. Compared with SVD++, UTS improves accuracy with at least 274% on *NDCG* and 119% on 1-*call*. However, UTS obtains worse performance in terms of diversity (*COV*) and novelty (*CIL* and *HitCIL*) than that of UserCF on both data sets. In diversity, UTS is up to 70% lower than UserCF on *COV*. The accuracy performance of UTSP-IJ is comparative to that of UTS, but the performance of diversity and novelty increases by at least 151% on *COV*, 156% on *HitCOV*, 451% on *CIL*, and 476% on *HitCIL* on both data sets.

V. CONCLUSION

This paper proposes a user based two-step recommendation model with popularity normalization UTSP which analyses user behaviors embedded in rating data and recommends items integrating user ratings, user similarity and item popularity. We first propose an improved Jaccard similarity function combined with popularity normalization to improve modelling performance. We then integrate the improved Jaccard function to the proposed user-based two step UTSP model. The proposed model variant UTSP-IJ actually overcomes the recommendation bias on popular items and significantly result in a more diverse and accurate recommendation. In addition to modelling contribution in recommender system area, we also propose two new metrics (*HitCOV* and *HitCIL*) to evaluate diversity and novelty of recommendation methods. Last but not least, the conducted comprehensive experiments also demonstrate the outstanding performance of the proposed model in terms of recommendation accuracy, diversity and novelty, compared with benchmark models UserCF, SVD++, pLPA, and previous two-step recommendation approach UTS.

REFERENCES

[1] J. Wang, Z. Wang, and J. Li, "A recommendation method for social collaboration tasks based on personal social preferences," *IEEE Access*, vol. 6, pp. 45206–45216, 2018.

[2] D. Zhou, X. Wu, W. Zhao, S. Lawless, J. Liu, "Query expansion with enriched user profiles for personalized search utilizing folksonomy data," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 7, pp. 1536–1548, Jul. 2017.

[3] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Feb. 2018, pp. 565–573.

[4] L. Zhang, T. Luo, F. Zhang, and Y. Wu, "A recommendation model based on deep neural network," *IEEE Access*, vol. 6, pp. 9454–9463, 2018.

[5] F. Zhang, "A personalized time-sequence-based book recommendation algorithm for digital libraries," *IEEE Access*, vol. 4, pp. 2714–2720, 2016.

[6] B. Sarwar, G. Karypis, G. Karypis, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, vol. 1, May 2001, pp. 285–295.

[7] D. Kluver, M. D. Ekstrand, and J. A. Konstan, "Rating-based collaborative filtering: Algorithms and evaluation," in *Social Information Access—Systems and Technologies* (Lecture Notes in Computer Science) vol. 10100, P. Brusilovsky and D. He, Eds. Springer, 2018, pp. 344–390. doi: 10.1007/978-3-319-90092-6.

[8] W. Chen, Z. Niu, X. Zhao, and Y. Li, "A hybrid recommendation algorithm adapted in e-learning environments," *World Wide Web*, vol. 17, no. 2, pp. 271–284, 2014.

[9] Y. Tay, L. A. Tuan, and S. C. Hui, "Latent relational metric learning via memory-based attention for collaborative ranking," in *Proc. World Wide Web Conf.*, Apr. 2018, pp. 729–739.

[10] J. Li, J.-J. Yang, Y. Zhao, B. Liu, M. Zhou, J. Bi, and Q. Wang, "Enforcing differential privacy for shared collaborative filtering," *IEEE Access*, vol. 5, pp. 35–49, 2017.

[11] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, Apr. 2017, pp. 173–182.

[12] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2008, pp. 426–434.

[13] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, "Large-scale parallel collaborative filtering for the netflix prize," in *Proc. Int. Conf. Algorithmic Appl. Manage.* Springer, 2008, pp. 337–348.

[14] X. Guan, C. T. Li, and Y. Guan, "Matrix factorization with rating completion: An enhanced SVD model for collaborative filtering recommender systems," *IEEE Access*, vol. 5, pp. 27668–27678, 2017.

[15] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proc. 4th ACM Conf. Recommender Syst.*, Sep. 2010, pp. 39–46.

[16] N. N. Liu, M. Zhao, and Q. Yang, "Probabilistic latent preference analysis for collaborative filtering," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, Nov. 2009, pp. 759–766.

[17] X. Zhao, Z. Niu, and W. Chen, "Interest before liking: Two-step recommendation approaches," *Knowl.-Based Syst.*, vol. 48, p. 46– 56, Aug. 2013.

[18] Y. Koren and J. Sill, "Ordrec: An ordinal model for predicting personalized item rating distributions," in *Proc. 5th ACM Conf. Recommender Syst.*, Oct. 2011, pp. 117–124.

[19] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, "Personalized ranking metric embedding for next new poi recommendation," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, Jun. 2015, pp. 2062–2068.

[20] X. Zhao, Z. Niu, W. Chen, C. Shi, K. Niu, and D. Liu, "A hybrid approach of topic model and matrix factorization based on two-step recommendation framework," *J. Intell. Inf. Syst.*, vol. 44, no. 3, pp. 335–353, Jun. 2014.

[21] X. Zhao, W. Chen, F. Yang, and Z. Liu, "Improving diversity of user-based two-step recommendation algorithm with popularity normalization," in *Proc. Int. Workshops, BDMS, BDQM, MoI, SeCoP Database Syst. Adv. Appl. (DAS-FAA)*, Dallas, TX, USA, Apr. 2016, pp. 15–26. doi: 10.1007/978-3-319-32055-7_2.

[22] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 89–115, Jan. 2004.

[23] G. Adomavicius and Y. Kwon, "Improving aggregate recommendation diversity using ranking-based techniques," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 5, pp. 896–911, May 2012.

[24] P. Adamopoulos and A. Tuzhilin, "On over-specialization and concentration bias of recommendations: Probabilistic neighborhood selection in collaborative filtering systems," in *Proc. 8th ACM Conf. Recommender Syst.*, Oct. 2014, pp. 153–160.

[25] M. Kunaver and T. Požrl, "Diversity in recommender systems—A survey," *Knowl.-Based Syst.*, vol. 123, pp. 154–162, May 2017.

[26] M. D. Ekstrand, F. M. Harper, M. C. Willemsen, and J. A. Konstan, "User perception of differences in recommender algorithms," in *Proc. 8th ACM Conf. Recommender Syst.*, Oct. 2014, pp. 161–168.

[27] F. Garcin, B. Faltings, O. Donatsch, A. Alazzawi, C. Bruttin, and A. Huber, "Offline and online evaluation of news recommender systems at swissinfo.ch," in *Proc. 8th ACM Conf. Recommender Syst.*, Oct. 2014, pp. 169–176.

[28] K. Kapoor, V. Kumar, L. Terveen, J. A. Konstan, and P. Schrater, "'I like to explore sometimes': Adapting to dynamic user novelty preferences," in *Proc. 9th ACM Conf. Recommender Syst.*, Sep. 2015, pp. 19–26.

[29] S. Vargas and Pablo Castells, "Improving sales diversity by recommending users to items," in *Proc. 8th ACM Conf. Recommender Syst.*, Oct. 2014, pp. 145–152.

[30] S. A. P. Parambath, N. Usunier, and Y. Grandvalet, "A coverage-based approach to recommendation diversity on similarity graph," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 15–22.

[31] A. Javari and M. Jalili, "A probabilistic model to resolve diversity–accuracy challenge of recommendation systems," *Knowl. Inf. Syst.*, vol. 44, no. 3, pp. 609–627, Sep. 2015.

[32] S. Vargas and P. Castells, "Rank and relevance in novelty and diversity metrics for recommender systems," in *Proc. 5th ACM Conf. Recommender Syst.*, Oct. 2011, pp. 109–116.

[33] P. Castells, N. J. Hurley, and S. Vargas, "Novelty and diversity in recommender systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds. Springer, 2015, pp. 881–918. doi: 10.1007/978-1-4899-7637-6.

[34] V. Vijayakumar, S. Vairavasundaram, R. Logesh, and A. Sivapathi, "Effective knowledge based recommender system for tailored multiple point of interest recommendation," *Int. J. Web Portals*, vol. 11, no. 1, pp. 1–18, Jan. 2019.

[35] D. Fleder and K. Hosanagar, "Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity," *Manage. Sci.*, vol. 55, no. 5, pp. 697–712, Mar. 2009.

[36] X. Zhao, Z. Niu, and W. Chen, "Opinion-based collaborative filtering to solve popularity bias in recommender systems," in *Proc. 24th Int. Conf. Database Expert Syst. Appl. (DEXA)*, Prague, Czech Republic, Aug. 2013, pp. 426–433. doi: 10.1007/978-3-642-40173-2_35.

[37] S.-B. Sun, Z.-H. Zhang, X.-L. Dong, H.-R. Zhang, T.-J. Li, L. Zhang, and F. Min, "Integrating triangle and jaccard similarities for recommendation," *PLoS ONE*, vol. 12, no. 8, pp. 1–16, Aug. 2017.

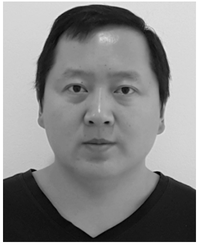
[38] M.-S. Shang and L. Lü, W. Zeng, Y.-C. Zhang, and T. Zhou, "Relevance is more significant than correlation: Information filtering on sparse data," *EPL (Europhys. Lett.)*, vol. 88, no. 6, Jan. 2010, Art. no. 68008.

[39] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, Oct. 2002.

[40] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver, and A. Hanjalic, "CLiMF: Learning to maximize reciprocal rank with collaborative less-is-more filtering," in *Proc. 6th ACM Conf. Recommender Syst.*, Sep. 2012, pp. 139–146.

KE NIU received the M.S. degree in software engineering and the Ph.D. degree in computer software and theory from the Beijing Institute of Technology (BIT). He is currently a Lecturer with the Computer School, Beijing Information Science and Technology University (BISTU), China. His research interests include artificial intelligence, data mining, and intelligent tutoring systems.

XIANGYU ZHAO received the B.S. degree from Beijing Jiaotong University, China, in 2007, and the M.S. and Ph.D. degrees from the Beijing Institute of Technology, China, in 2009 and 2014, respectively. He is currently an Assistant Professor with the Beijing Research Center for Information Technology in Agriculture. His current research interests include recommender systems, data mining, and agricultural information.



FANGFANG LI received the Ph.D. degree in computer software and theory from the Beijing Institute of Technology, in 2014, and the Ph.D. degree in data analytics from the University of Technology Sydney (UTS), in 2016. He is currently with oOh!media Limited, as a Data Scientist, focusing on data science research and applications of recommender system, natural language processing, optimization, and out-of-home advertising.

NING LI was born in 1964. He received the Ph.D. degree from the University of Kent, Canterbury, U.K., in 1994. He is a Professor and Dean of Computer School, Beijing Information Science and Technology University. His research interests include documents information processing, XML, and multimedia.

XUEPING PENG received the B.S. degree in automation from Hefei University, and the M.S. degree in software engineering from the Beijing Institute of Technology (BIT), China, and the joint Ph.D. degrees in computer software and theory from BIT and in computer science from the University of Technology Sydney (UTS), where he is currently a Lecturer with the Centre for Artificial Intelligence (CAI), School of Computer Science (SoCS), Faculty of Engineering and Information Technology (FEIT). His current research interests focus on data mining, artificial intelligence, and healthcare.

WEI CHEN received the Ph.D. degree in computer science from the City University of Hong Kong, in 2009. He is currently an Associate Professor with the Agricultural Information Institute, Chinese Academy of Agricultural Sciences. His research interests include agricultural data analysis and user modeling.