

## A Novel Recommendation Algorithm Based on Time-aware

Liang Zhang<sup>1</sup>, Jing Yi<sup>1</sup>, Ping Lv<sup>1,\*</sup> and Xue Zhou<sup>2</sup>

<sup>1</sup>School of economics and management, Guizhou Normal University, Guiyang 550001, China

<sup>2</sup>School of management, University of Chicago, Chicago, IL 60637, United States

Received 12 October 2020; Accepted 29 December 2020

### Abstract

The primary key in novel recommendation is to judge whether an item is “new” for target users. The novelty of recommended items to users changes with time. Therefore, the effects of time factors on user preference, as well as the novelty and accuracy of the recommended items, have become key to novelty recommendation research. To increase novelty based on recommendation accuracy, a novel recommendation algorithm based on time-aware was proposed. First, a time attenuation parameter was introduced into a traditional algorithm to strengthen the weights of recently adopted items. Next, the calculation of innovation adoption activism was adjusted through time information. Differences among recommended items were calculated by combining the time of items that have been adopted by users and the quantity of relevant categories. Lastly, the novelty of recommended items was calculated by the result fusion strategy for Top-N recommendation. Meanwhile, the effectiveness of the proposed algorithm was verified by conducting an offline experiment. Results demonstrate that the novelty of the recommendation list by the proposed algorithm is significantly higher than that of traditional algorithms. Novelty when the number of items in the candidate set reaches 300-400 reaches the peak. The recall rate of unknown items increases by 25%, whereas the recall rate of known items decreases by 60%. The average popularity of the recommendation list decreases by 50%, and the coverage rate increases by 200%. The proposed algorithm improves the ability of the recommendation system (RS) to explore novel items. This study provides references for improving the satisfaction of users with RSs.

*Keywords:* time-aware, novelty, recommendation algorithm, accuracy

### 1. Introduction

With the rapid development of information technology and Internet computation, Web 2.0 technology makes human beings consumers and producers of information. Not only does it bring significant convenience to human beings, it also intensifies the problem of information overload [1]. Recommendation systems (RSs) establish a user preference model by analyzing the historical behavior of users rather than asking users to provide explicit demands; thus, users obtain information that meet their interests and needs positively. Almost all large-sized e-commerce systems use different forms of RSs to different extents, and personalized recommendation technologies can help users make wise choices from numerous products [2]. In most cases, RSs provide a list of many similar items to assure accuracy. Moreover, RSs consider the historical consumption records or behavioral similarity of users, thus resulting in redundancy and serious homogeneous problems of recommendation results. As a result, users hardly encounter fresh and diversified content, thus decreasing user satisfaction [3]. Since the development of RSs, most associated studies have focused on the accuracy metric. In addition to accuracy, the influencing factors of user satisfaction include validity, novelty, and usability. Therefore, many studies on the potential defects of RSs have been conducted [4]. Pursuing high accuracy in RSs decreases diversity and novelty; thus, it may cause some

problems, such as homogeneity, polarization, echo chamber, and wrong information [5]. Given that the novelty metric is close to user satisfaction, it has attracted considerable attention and thus is being studied further. However, most novel recommendation algorithms are challenged with decreasing accuracy [6].

Substantial situational information facilitates the establishment of an accurate user preference model, thus improving the performance of RSs [7]. Such information mainly includes location, time, weather, equipment, and emotions. Among them, time is considered one of the most useful information. Using time information is beneficial to analyzing the evolution of user preference and recognizing the periodicity of user habits and interests. Therefore, time information becomes a key input variable of context-aware RSs. Using time information can accurately establish a user preference model [8] and effectively improve the accuracy of RSs [9]. However, how to improve novelty of RS based on time information is rarely studied.

For this reason, a novel recommendation algorithm was designed by studying the temporal characteristics of preference, unknown, and difference of novelty items, aiming to establish an improved user preference model in terms of accuracy. Moreover, items that conform to user novelty-aware at the current moment were recommended.

### 2. State of the art

With respect to the problem of accurate recommendation, novel recommendation has been highly considered in recent

\*E-mail address: ftygz@vip.sina.com

ISSN: 1791-2377 © 2020 School of Science, IHU. All rights reserved.

doi:10.25103/jestr.136.05

years. To determine relevant definitions, some studies have attempted to explain novelty by using the existing cold boot method. When solving the novelty problem of new items, Yu found that items that are kept unknown from most users and have low popularity often appear novel to users; he also measured the novelty of recommendation results in accordance with the proportion of new items in the recommendation list [10]. Chou et al. defined the percentage of artists that users know in the music recommendation as novelty [8]. However, these studies only evaluated the overall novelty of recommendation results and did not define novelty of items. Chen et al. considered scoring time and introduced the concept of “innovator” and “potential followers.” They believed that the scoring terms of the innovator are novel to target users [11]. Kapoor et al. believed that novel items refer to newly added unevaluated items of the system and items that users do not know or may have forgotten [12]. Given that item popularity is convenient for calculation, commodities with low popularity are believed to be novel [13, 14]. However, global value cannot reflect user interest and cannot express the novelty of the same item to different users. On the basis of a user–item bipartite graph, Wu et al. combined random walk and heat conduction theory by parameters, thus increasing accuracy and novelty [15]. Wang et al. proposed a recommendation model based on the discovered relations of user items; this model increased the novelty and diversity of recommendation list effectively compared with the traditional k-nearest neighbor (KNN) algorithm and the nonimproved classical algorithm based on MMR [16]. Yu et al. proposed the correlations between users and community in terms of adjacent domains and subjects and developed a measurement method of user–community distance. Later, the novelty of candidate communities was calculated on the basis of the measured user–community distance, and novel communities were recommended, with consideration to the accuracy of recommendation results [17]. Pietro et al. proposed the concept of adjacent possibility, which can further meet user needs, to the redesigning of RSs [18]. None of the abovementioned algorithms have considered the differences of novelty. Hence, novel recommendation must establish a user preference model according to items that are adopted by users.

Clustering algorithms are often used to user preference modeling. Han et al. clustered users and items into several subsets according to user–item evaluation matrices and item attributes derived by field experts. Later, candidate items were explored by selecting item attributes. The mean absolute error and root mean square error were increased significantly [19]. Jiang et al. introduced information entropy and biclustering into collaborative filtering, which was superior to the benchmark algorithm in terms of accuracy and calculation cost [20]. Raja et al. proposed an RS of personalized mobile multimedia applications on the basis of user evaluation, comment text, application description, and application popularity. This RS analyzes user comments and describes applications by using the latent Dirichlet allocation (LDA) subject model, implements k-means clustering of all extracted application features in principal component analysis, and leaves optimal potential features. This algorithm assures the correlation and diversity of the recommendation list [21]. Das et al. clustered or divided user spaces of the system in accordance with user positions by using two hierarchical space division data structures, namely, k-d tree and quadtree, and then predicted the grade of target users with these clusters. Operation time

was decreased significantly while maintaining the acceptable recommendation quality [22]. Mezni et al. mixed the k-means clustering method and multipopulation variants of the famous particle swarm optimization algorithm to eliminate users who share a few public web services with the activity user in a specific context [23]. Selvi et al. proposed a supervised adaptive genetic neural network based on an improved k-means clustering algorithm to position the favorite data points of users and provide effective recommendations [24]. Devarajan et al. generated limited clusters by using four different clustering optimization algorithms based on swarm intelligence [25]. These studies can describe user preferences accurately by combining clustering technology with LDA, genetic neural networks, and other algorithms and maximizing spatial and context information. However, the above algorithms ignore the timeliness of novel recommendations. Given that user preference changes with time, the novelty of items also changes with time.

The fuzzy clustering algorithm can establish user interest models and can be combined with situational information effectively. Liu et al. introduced a model of users’ stable and current interests and solved problems of traditional collaborative filtering recommendation algorithms, such as data sparsity and equal consideration to user interests at different time, based on the fuzzy clustering algorithm [26]. Mohanty et al. proposed a recommendation framework of user access behavior based on rough fuzzy clustering technology [27]. Zeng et al. clustered users in accordance with scenario similarities, recognized the nearest neighbor that had the most target users according to the principle of similarity propagation, and gave recommendations according to item scores by predicted target users [28]. These studies mainly use scenario information in fuzzy clustering algorithms and analyze user preference dynamically; these approaches are beneficial to increase recommendation quality.

Zhao et al. designed hierarchical and time-aware sampling and extracted representative data from new and historical data to solve conceptual drifting and capture long-term user preference [8]. To meet the requirements of recommendation reliability and user flavor variation, Noulapeu proposed a realistic temporal service recommendation method based on the reasoning of implicit trust relations and integrated time features into the recommendation process, which considered the timeliness of user demands [9]. Aliannejadi et al. proposed a two-stage CR algorithm, which punishes users who are sensitive to time in the past and recommend their point of interest (POI) by combining influencing factors of POI recommendation and variation of POI popularity variance with time, aiming to explain long-term behavioral mode of users [29]. Li et al. established a user similarity model and a commodity similarity model by using emotional analysis technology in accordance with description information and comment data of items. They proposed the similarity matrix decomposition method for items and users. The consistency between implicit and dominant factors was naturally enlisted into the learning of implicit factors of users and commodities, thus enabling an accurate prediction of user preference to different objects [30]. By capturing user preference changes, Chen et al. integrated dynamic time attenuation to distinguish grades (instantaneous, short term, or long term) of user interests and make recommendations according to different interest levels [31]. Zhang et al. divided users into several clusters according to the attributes of the

configuration files of users. A virtual opinion leader was designed for each cluster as a representative of the whole cluster; this design could remarkably decrease the dimensions of the original user-item matrix and the time complexity of RSs under the premise of accuracy [32]. Zhang et al. introduced personalized time weight and combined the forgetting curve and good similarity to decrease influences of outdated information and maintain influences of user stable preferences [33]. Zhao et al. proposed a dynamic DTree2Vec scheme for serialized books and established a model for texts with different completion degrees. This model realizes uniform semantic feature expressions to measure the semantic correlations of books and assures recommendation quality by tracking the updated state of serialized books and real-time adding follow-up chapters. This model also effectively relieves dynamic capture problems and real-time recommendation problems in book recommendation [34].

The above analyses mainly focus on user preference modeling and the improvement of user accuracy and novelty. However, no study has fully considered the temporal changes of user preference, novelty-aware ranking, and item features. In this study, the accuracy of RS was improved by using the result fusion strategy and time information. On this basis, novel recommendation was performed by combining the time characteristics of user preference and the time point of users adopting new items, thus improving the novelty of RS.

The remainder of this paper is organized as follows: Section 3 mainly describes the specific idea of novel recommendation algorithm based on time-aware. Section 4 verifies the performance of the algorithm through an offline experiment and analyzes the key influencing factors of performance improvement. Section 5 summarizes the conclusions.

### 3. Methodology

In view of definition of novelty, time factors were embedded into the definition and measurement of item novelty. Novel items have three feature aspects (i.e., preference, unknown, and difference) for target users. These three aspects change with time for users. Therefore, the situational information of time must be considered to measure the novelty of an item and thus target users accurately. As shown in Eq. (1), the time variable is embedded into the definition of item novelty:

$$Novelty(i, u, t) = p(i|like, u, t) \times p(i|unknown, u, t) \times dis(i, I_u, t) \quad (1)$$

where  $Novelty(i, u, t)$  is the novelty of item  $i$  to user  $u$  at time  $t$ ;  $p(i|like, u, t)$  is the possibility of user  $u$  to like item  $i$  at time  $t$ ;  $p(i|unknown, u, t)$  is the possibility of user  $u$  not knowing item  $i$  at time  $t$ ;  $dis(i, I_u, t)$  is the difference between item  $i$  and the item that user evaluates at time  $t$ , and it refers to the set of evaluated items of the user. In the following text, how to involve the time variable to increase the possibility of novel recommendation was analyzed thoroughly based on these three aspects.

#### 3.1 Time-aware preference

Preference means that an item must first conform to the interests of users. The variation of user interest with time and the timeliness of items must be fully considered to increase the accuracy of RSs. Based on the above analysis,

many modeling methods of time information are available, but the used recommendation algorithm varies and shows different effects, even showing the opposite effect. In this study, collaborative filtering and content-based recommendation analyzed the effects of time information on accuracy and the effects of accuracy variation on novel recommendation.

##### 3.1.1 Time-aware collaborative filtering

The collaborative filtering (CF) algorithm is one of the most common algorithms in commercial RSs. The CF algorithm is mainly divided into two core steps. First, it calculates the similarity of different items offline based on the data on user interaction behavior of the system, thereby forming an item similarity matrix. Next, it predicts the scores of items that users have not evaluated, according to the interaction behavior of target users and the item similarity matrix; it also recommends  $N$  items with the highest predicted scores. Time information can change greatly in these two core steps.

The first core is similarity of objects. Traditional algorithms only consider the item scores given by users. If users' scores to two items are close, then these two items are considerably similar. This method ignores the scoring time of users to an item; two items that have similar scores in a short period obtain a high similarity. If the time span is too large, then the probability for changes of user preference becomes higher, and the score similarity of items might be less correlated with item similarity. For example, the song that a user likes today might be similar with a song he/she liked yesterday, but it may be remarkably different from the song that he/she liked five years ago. Based on this idea, item similarity cannot be calculated by the traditional cosine distance. After the time factor is added, item similarity is calculated by Eq. (2).

$$sim(i, j) = \frac{\sum_{u \in U} \frac{u_i \times u_j}{1 + \gamma |t_{ui} - t_{uj}|}}{\sqrt{\sum_{u \in U} u_i^2} \sqrt{\sum_{u \in U} u_j^2}} \quad (2)$$

where  $t_{ui}$  and  $t_{uj}$  are the scoring time of user  $u$  to items  $i$  and  $j$ , respectively;  $\gamma$  is the time attenuation parameter;  $u_i$  and  $u_j$  are the user scores to item  $i$  and  $j$ , respectively;  $U$  is the user set.

The second core is online recommendation. Online recommendation is mainly based on the predicted scores of items. The predicted scores of items are related not only with the similarity of goods but also with the interaction behavior of target users. The recent interaction behavior of users can reflect the current preference of users more than the interaction behavior from long ago. Therefore, increasing the weights of the recent interaction behavior of users is suggested when predicting scores. The predicted score function of items is shown in Eq. (3).

$$predict(i, u, t) = \frac{\sum_{j \in I_u} \frac{sim(i, j) u_j}{1 + \gamma |t - t_{uj}|}}{\sum_{j \in I_u} sim(i, j)} \quad (3)$$

where  $t$  is the recommendation time. The weights of interaction behavior with great time intervals are decreased when calculating item similarity, whereas the weights of the previous interaction behavior of target users are decreased when predicting scores. The reduction amplitude is

positively related with value, which is determined in accordance with the change frequency of user interests.

### 3.1.2 Time-aware content-based recommendation

The traditional content-based recommendation (CBR) algorithm assumes that user interests are constant. The CBR algorithm is mainly applied to the recommendation of news, documents, and webpages. This algorithm first calculates item similarity in accordance with the eigenvectors of items and then recommends items that mostly conform to user preference. The CBR algorithm does not need substantial user and historical score data, and it can provide recommendations only when there is one user. The CBR algorithm is mainly composed of three steps. First, the eigenvalues of items are extracted. Second, a preference model is established in accordance with the historical interests of target users. Lastly, user preference to unevaluated items is predicted by the preference model, and N items with the highest predicted scores are recommended to target users.

For an unevaluated item, the KNN algorithm first recognizes k items, which users have scored and are the most similar with the unevaluated item. Next, it calculates the weighted average of target users to these k items to judge user preference to the item. The prediction formula for user preference to the new item is shown in Eq. (4).

$$predict(i,u,t) = \frac{\sum_{j \in Knn_u} sim(i,j)u_j}{\sum_{j \in Knn_u} sim(i,j)} \quad (4)$$

where  $Knn_i$  is the set of new items the most similar k items in item set which the target user has been scored. Given that  $sim(i, j)$  is calculated by the eigenvectors in the CBR algorithm and the eigenvalues of items generally remain constant as time goes, adding the time factor is difficult. In the CBR algorithm, only the variations of user interaction behavior with time are considered when predicting the scores of target users. The weights of recent interaction behavior are increased by the time factor.

$$predict(i,u,t) = \frac{\sum_{j \in Knn_u} \frac{sim(i,j)u_j}{1 + \gamma |t - t_{ij}|}}{\sum_{j \in Knn_u} sim(i,j)} \quad (5)$$

### 3.2 Particle representation

Unknown is an important factor in novel recommendation. However, judging the unknown of an item is a difficult problem. Users can understand an item from various channels. However, such information cannot be reflected in a specific dataset. Hence, many novel recommendation algorithms assume that more popular items are more likely to be known by users [35]. However, popularity represents the “wisdom of crowds” [36], and most RS algorithms recommend items that conform to the preference of the target users in accordance with the “wisdom of crowds.” Decreasing the accuracy of RS to increase the novelty of RS and eliminate high-popularity items is inevitable. Popularity is related with the accuracy and novelty of RS. Therefore, the key of novel recommendation is how to use the popularity metric reasonably in recommendation algorithms.

In novel recommendation algorithms based on diffusion of innovation (DI) theory [37], the positivity of innovation adoption (PIA) of target users to each item type is calculated and used to measure their initiatives of adopting such new

items. A high PIA indicates that the target users might choose the new items at an early time. In novel recommendation algorithms based on DI, the PIA of users to a type of items is calculated in accordance with the average evaluation time of the user to similarly scored items. However, such PIA may change with time. For example, users prefer to read some classical introduction books and papers when they just contact the recommendation algorithms. These papers are popular products in this type. However, users may prefer new papers and books of RS gradually as they become increasingly familiar with RS. Hence, the PIA of this type of books and papers increases. Therefore, assuming that the recent innovation adoption behavior of users can reflect the PIA of the user is reasonable. After the time factor is added, the PIA of target users to an item set can be calculated by Eq. (6).

$$PIA(C_j, u, t) = \frac{1}{|I_u \cap C_j|} \sum_{i \in I_u \cap C_j} \frac{PIA_{ui}}{1 + \gamma |t - t_{ui}|} \quad (6)$$

where  $C_j$  is the category in which item j belongs to,  $PIA_{ui}$  is PIA of user u to the item j,  $t_{ui}$  is the time that user u adopts item j, and  $I_u$  is the set of items that user u has adopted.

However, the scored items of a user might not cover all categories. Hence, the mean PIA of all scored items by the user is used, and it is redefined as Eq. (7). For most users, a limited number of categories is covered by scored items. This idea is too coarse.

$$PIA(C_j, u, t) = \begin{cases} \frac{1}{|I_u \cap C_j|} \sum_{i \in I_u \cap C_j} \frac{PIA_{ui}}{1 + \gamma |t - t_{ui}|} & I_u \cap C_j \neq \emptyset \\ \frac{1}{|I_u|} \sum_{i \in I_u} \frac{PIA_{ui}}{1 + \gamma |t - t_{ui}|} & I_u \cap C_j = \emptyset \end{cases} \quad (7)$$

To address this problem, clusters with dendritical structures are generated by using cohesion hierarchical clustering algorithm. If a cluster does not cover the scored items of users, then a search upward is conducted to find the father cluster until the cluster with the scored items of the user is recognized.

### 3.3 Time-aware difference

The third characteristic of novel items is difference. In other words, the recommended item must be different from the item that the target user is familiar with. Difference mainly covers two aspects: difference measurement among items and searching familiar items of users.

Items that users are familiar with are determined by analyzing the adopted item set of users. The rated-item clustering (RC) algorithm of the DBSCAN algorithm can establish a model of familiar items well [38]. However, DBSCAN establishes the user preference model after noise points are eliminated, without consideration to the time of clustering. For instance, a type of items that users have interacted with recently can represent the current POI of users. If the recommended items have great differences with this type of items, then the accuracy of the RS can be influenced seriously. Therefore, the calculation of difference of this type of items decreases the weights of similarity. Weights are related with the number of included items in this type except for time. If more items are included, then users become more familiar with this type of items; otherwise, users do not become familiar. Therefore, the

difference between the recommended item and user  $i$  is calculated by Eq. (8).

$$dis(i,u,t) = \min_{C_j \in C_u} (1 - w_{u,C_j}(t, |C_j|) sim(i, C_j)) \quad (8)$$

where  $sim(i, C_j)$  is the similarity between item  $i$  and the category of item  $j$ .  $w_{u,C_j}(t, |C_j|)$  is the weight function (Eq. (9)). When all the adopted items of the user belong to the category and the interaction behavior is made long ago, the weight is 1. If a type only has one item and the interaction time is considerably close to the recommendation time, then the weight is  $1/|I_u|$ . In this manner, the weight of similarity is decreased.

$$w_{u,C_j}(t, |C_j|) = \frac{1}{2} \left( \frac{|C_j|}{|I_u|} + \left( 1 - \frac{\sum_{i \in C_j} t_{ui}}{t |C_j|} \right) \right) \quad (9)$$

Based on the above three features of novel items, the time factor is added into preference, unknown, and difference. The offline experiment adds the above time-aware characteristics into the above-mentioned algorithms to study the influences of time information on novel recommendation.

## 4 Result analysis and discussions

### 4.1 Experiment and evaluation metric design

The offline experiment used two datasets, namely, MovieLens and LastFM. The traditional CF and CBR algorithms discussed the effects of time information on novel recommendation. The offline experimental process of novel recommendation is shown in Fig. 1 [39]. Experimental results were comprehensively evaluated by the novelty (Eq. (10)), average popularity (Eq. (11)), and coverage (Eq. (12)) of the recommendation list. First, the effects of time-aware preference on novel recommendation were studied. Differences of time-aware collaborative filtering (TCF), time-aware content-based recommendation (TCBR), and traditional algorithms in evaluation metric were analyzed. Next, the TCF and TCBR algorithms were applied to the DI and RC-DI algorithms to discuss the variations of novel recommendation performance. Lastly, time-aware unknown and time-aware differences were added. For distinction, the time-aware algorithms were denoted with an additional letter "T" before the names of the traditional algorithms.

$$\begin{aligned} Novelty &= \frac{1}{|U|} \sum_{u \in U} (2Recall_{au} - Recall_{bu}) \\ &= \frac{1}{|U|} \sum_{u \in U} \left( 2 \frac{|Rec_{cu} \cap Hide_{au}|}{|Hide_{au}|} - \frac{|Rec_{cu} \cap Hide_{bu}|}{|Hide_{bu}|} \right) \end{aligned} \quad (10)$$

$$Avg\_pop = \frac{1}{|U|} \sum_{u \in U} \frac{\sum_{i \in Rec_{cu}} RN_i}{|Rec_u|} \quad (11)$$

$$Coverage = \frac{|\cup_{u \in U} Rec_u|}{|I|} \quad (12)$$

where  $Hide_{au}$  and  $Hide_{bu}$  are the sets of items hidden by user  $u$  in  $R_a$  and  $R_b$ , respectively,  $Rec_u$  is the set of recommended items for user  $u$ ,  $RN_i$  represents the

number of evaluation times of item  $i$ , and  $U$  and  $I$  represent the user and item sets, respectively. As shown in the Eq. (6),  $Recall_b$  is the recall rate in the traditional experiment, and  $Recall_a$  is the accuracy measurement for the prediction of future user demand by the recommendation system. The quantity of recommendation lists is extremely limited. Thus, each recommendation list is expected to contain items meeting future user demands and reduce the items already known by the user, thereby accurately defining the meaning of novelty metrics. Average popularity and coverage measure the ability of recommendation algorithms to explore long-tail products. These measures are also a manifestation of the novel recommendation ability of the algorithm.

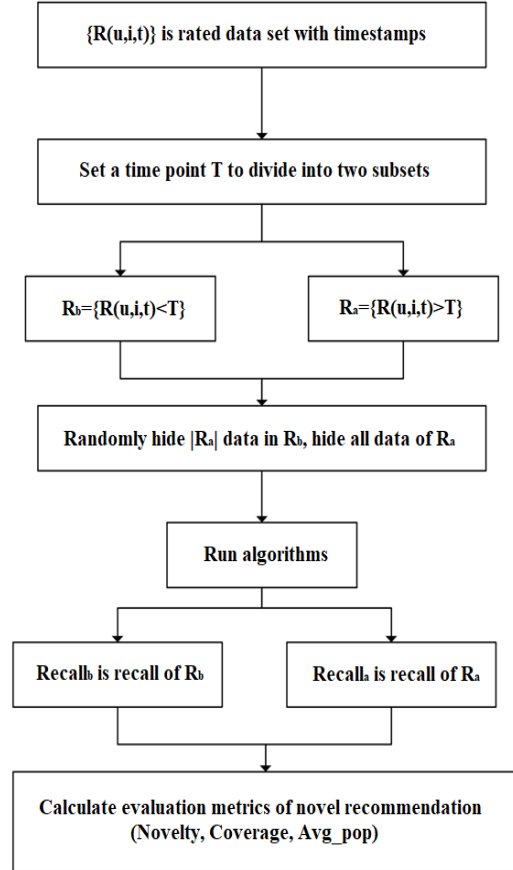


Fig. 1. Offline experimental scheme of novel recommendation

### 4.2 Time-aware preference

An increase in the accuracy of novel recommendation algorithms is a key influencing factor of novel recommendation. Table 1 and Table 2 show that the TCF and TCBR algorithms are superior to traditional algorithms in accuracy.  $Recall_a$  is improved in comparison with  $Recall_b$ , thus slightly improving novelty. TCF considers the time series of user preference changes when measuring similarity among users, and it can recognize neighbors of users accurately, thus improving  $Recall_a$  and  $Recall_b$  simultaneously. The TCF and TCBR algorithms increase the weights of the recent interaction behavioral items of users during the calculation of predicted scores; thus, items in the recommendation list are close to the recent preference items of users. As a result,  $Recall_a$  is improved more than  $Recall_b$ . According to an analysis on  $Avg\_pop$ ,  $Avg\_pop$  of time-aware algorithms is improved significantly, whereas Coverage is slightly reduced. When the recommendation

quantity is increased, *Recall\_a* and *Recall\_b* approach those of traditional algorithms continuously although they are improved. However, *Avg\_pop* and Coverage are improved to some extent. In Table 3, the improvements of *Recall\_a*, *Recall\_b*, *Avg\_pop*, and Coverage of TCF decrease continuously with the increase in the neighbor number of users. A recall rate higher than 80% even degrades slightly. The “wisdom of crowds” can relieve the effects of time information on the recommendation results of the TCF algorithm.

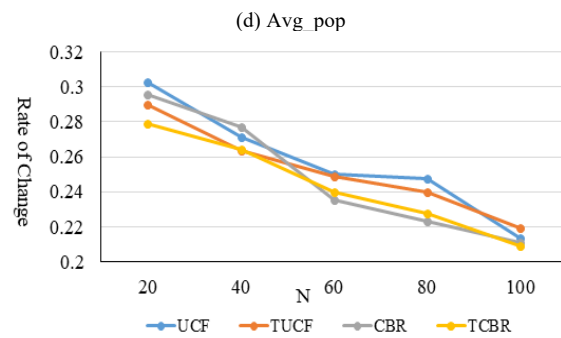
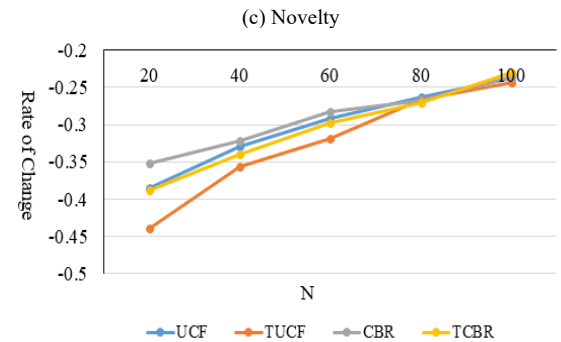
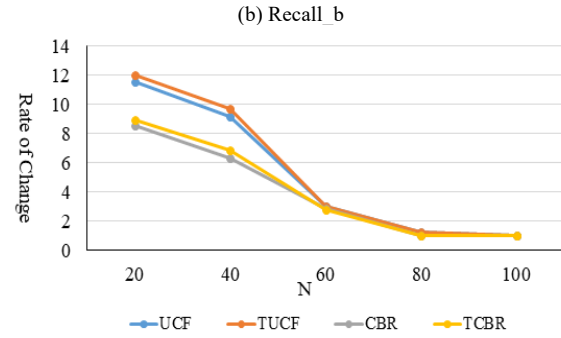
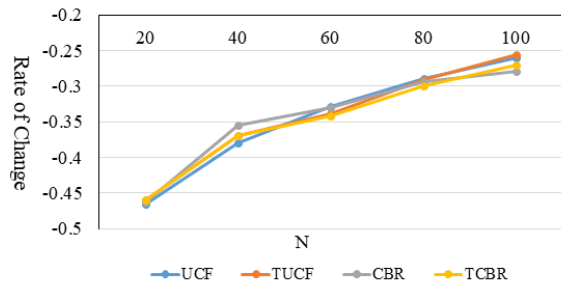
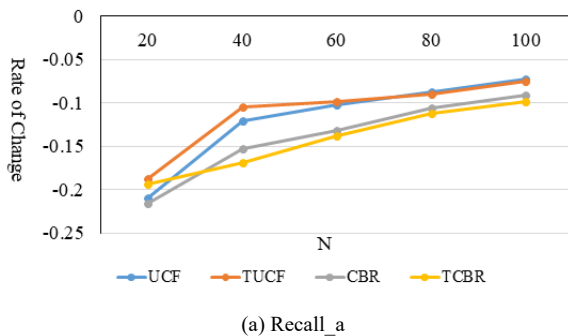
**Table 1.** Experimental results of the TUCF algorithm

DataSet	N	Recall a	Recall b	Novelty	Avg_pop	Coverage
Movie Lens	20	0.0532	0.0382	0.6267	-0.2043	-0.0123
	40	0.0453	0.0301	0.4362	-0.1801	-0.0102
	60	0.0383	0.0211	0.1913	-0.1522	-0.0143
	80	0.0224	0.0132	0.0661	-0.1023	-0.0141
	100	0.0101	0.0087	0.0144	-0.0543	-0.0115
Last FM	20	0.0632	0.0401	0.9256	-0.2109	-0.0032
	40	0.0561	0.0324	0.6562	-0.1978	-0.0098
	60	0.0314	0.0251	0.0874	-0.1732	-0.0102
	80	0.0198	0.0102	0.0655	-0.1201	-0.0089
	100	0.0093	0.0098	0.0055	-0.0912	-0.0035

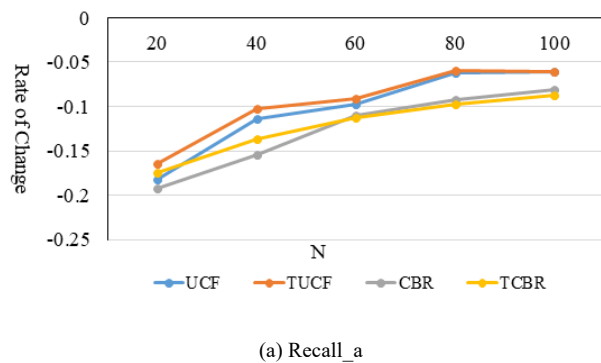
**Table 2.** Experimental results of the TCBR algorithm

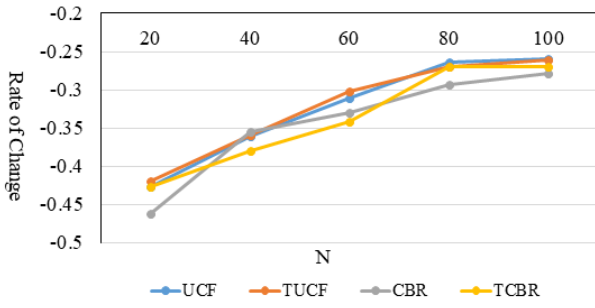
DataSet	N	Recall a	Recall b	Novelty	Avg_pop	Coverage
Movie Lens	20	0.0354	0.0267	0.3208	-0.1897	-0.0098
	40	0.0298	0.0198	0.2785	-0.1768	-0.0121
	60	0.0219	0.0123	0.0978	-0.1438	-0.0101
	80	0.0178	0.0097	0.0333	-0.1098	-0.0097
	100	0.0076	0.0065	0.0268	-0.0768	-0.0114
Last FM	20	0.0387	0.0221	0.6197	-0.2097	0.0021
	40	0.0301	0.0242	0.1370	-0.2001	-0.0065
	60	0.0225	0.0165	0.0556	-0.1694	-0.0045
	80	0.0154	0.0108	0.0378	-0.1298	0.0015
	100	0.0101	0.0096	0.0114	-0.1004	-0.0065

Based on the above experimental results, time-aware algorithms can improve accuracy, but they cannot improve novelty significantly. In the following, novel recommendation was performed on the basis of the TUCF and TCBR algorithms by using the result fusion strategy. The number of candidate sets was chosen in accordance with the maximization of the novelty metric. The number of neighbors in the TCF algorithm was set to 20. The experimental results are shown in Figs. 2 and 3.

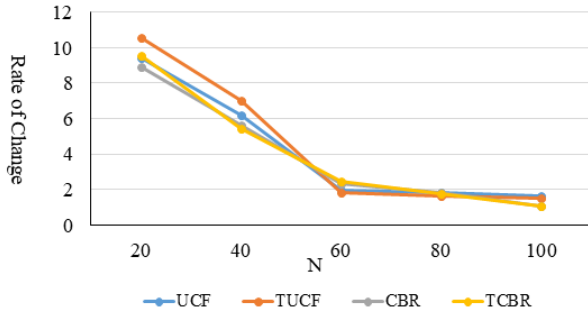


**Fig. 2.** Experimental results of the TUCF and TCBR algorithms using the result fusion strategy based on MovieLens

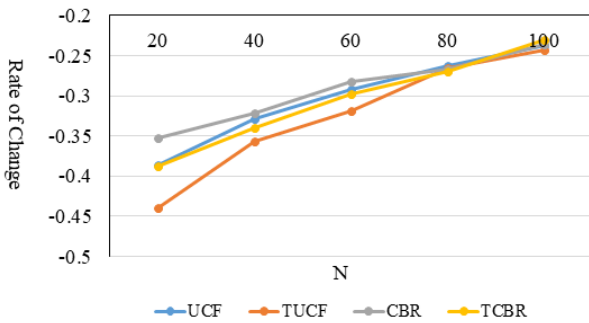




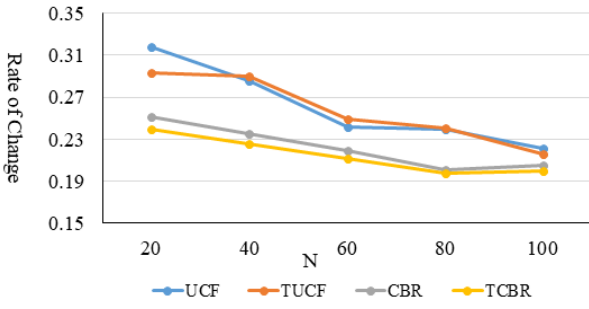
(b) Recall\_b



(c) Novelty

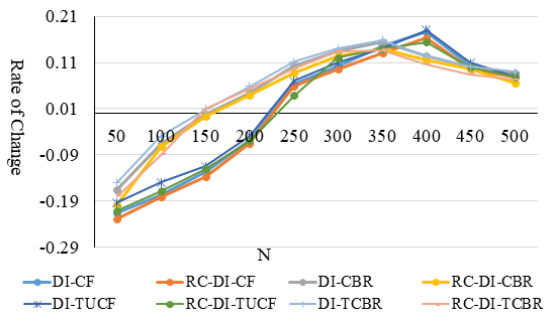


(d) Avg\_pop

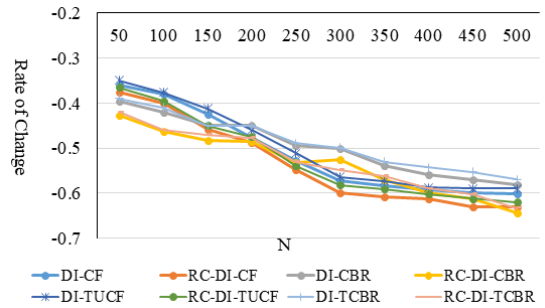


(e) Coverage

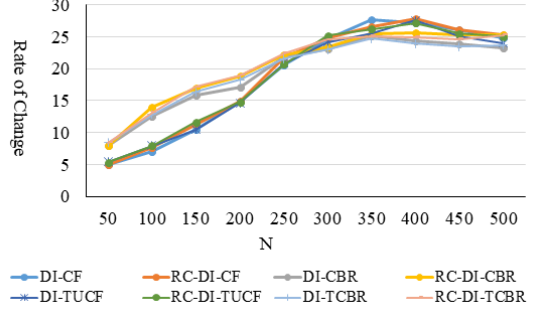
Fig. 3. Experimental results of the TUCF and TCBR algorithms using the result fusion strategy based on LastFM



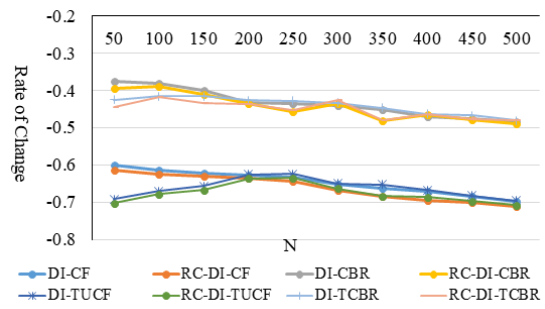
(a) Recall\_a



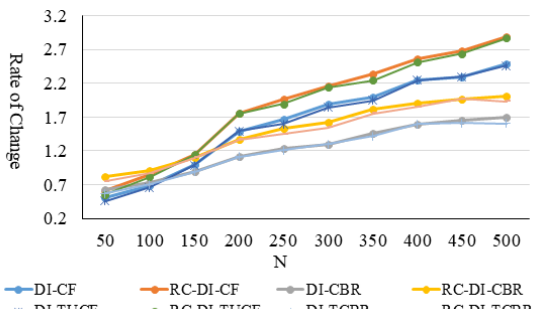
(b) Recall\_b



(c) Novelty

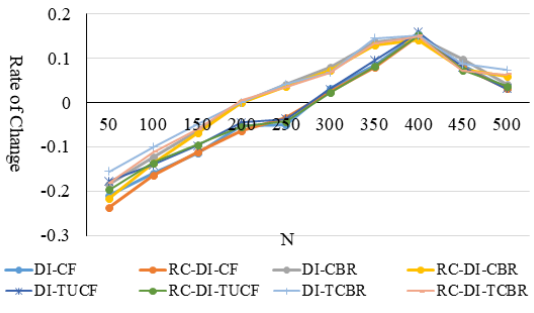


(d) Avg\_pop



(e) Coverage

Fig. 4. Experimental results of the TUCF and TCBR algorithms using the result fusion strategy based on MovieLens



(a) Recall\_a

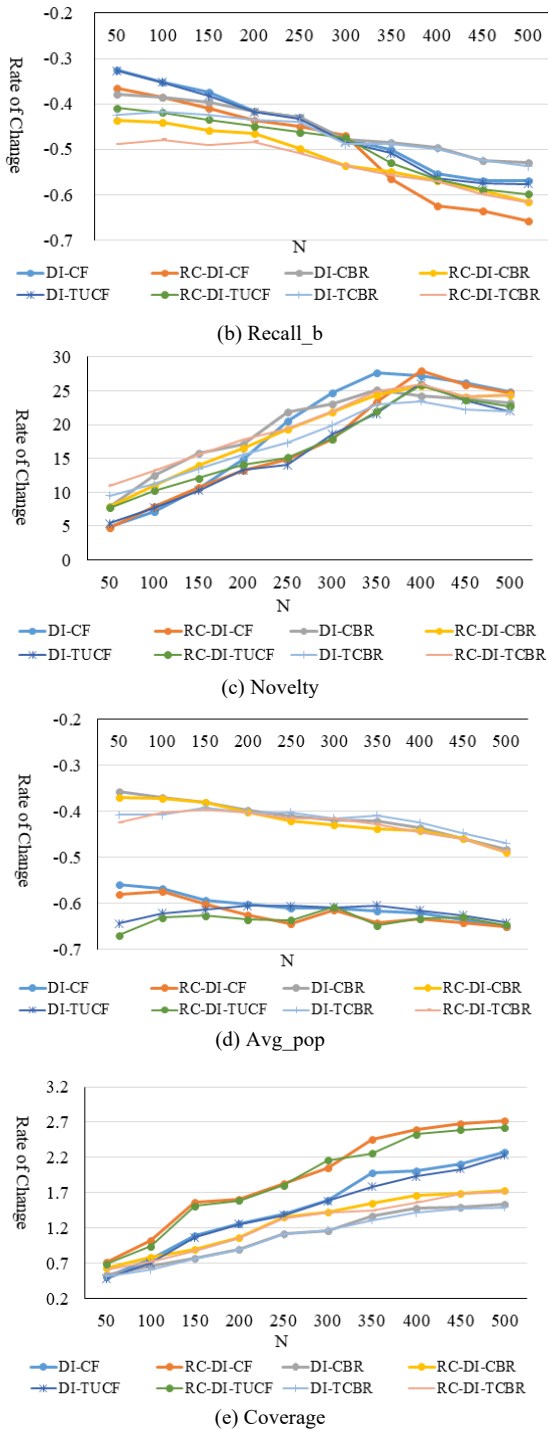


Fig. 5. Experimental results of the TUCF and TCBR algorithms using the result fusion strategy based on LastFM

After using the result fusion strategy, *Novelty* increases to the maximum when the number of recommendations is set to 20, and it approaches that of the traditional algorithm continuously with the increase in the number of recommendations. When the number of recommendations is small, *Recall\_a* of time-aware algorithms decreases less than that of traditional algorithms, whereas *Recall\_b* does not change significantly, thus improving *Novelty*. An analysis on *Avg\_pop* and *Coverage* reveals that *Avg\_pop* of recommendation results is improved significantly when the number of recommendations is small, but *Coverage* declines to some extent. The average life cycle of a recommendation item set shows that when the number of recommended items is small, the recommendation results are apt to items that are

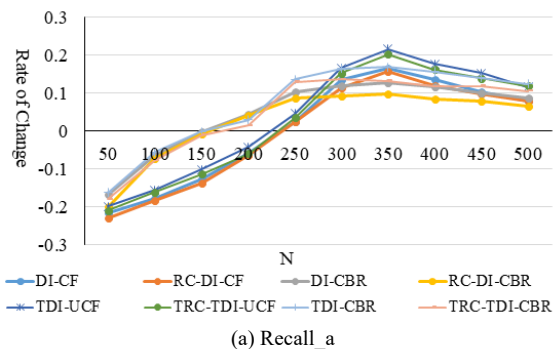
generated recently, thus improving *Avg\_pop* but decreasing *Coverage*. This result is obtained mainly because the weights of recent interaction items are increased when calculating the predicted score of the item, making the recommendation results prone to growing and mature items.

Time-aware algorithms can improve novel recommendation to some extent by using the result fusion strategy. In the following text, the TUCF and TCBR algorithms were applied to the DI and RC-DI algorithms, respectively. The experimental results are shown in Figs .4 and 5. N is the number of items in the candidate set and the number of items in the recommendation list is 20. Time-aware algorithms increase *Novelty* significantly in comparison with traditional algorithms when the candidate set has a few items. *Novelty* of time-aware algorithms reaches the peak when the number of items in the candidate set is between 300 and 400. However, not all time-aware algorithms achieve better *Novelty* than traditional algorithms. According to an analysis of *Recall\_a* and *Recall\_b*, although time-aware algorithms achieve some improvement in accuracy, the improvement of *Recall\_a* decreases gradually with the increase in the number of items in the candidate set, whereas the improvement of *Recall\_b* does not change greatly. As a result, the novelty of recommendation results of time-aware algorithms is not significantly higher than those of traditional algorithms. The improvement of *Avg\_pop* and *Coverage* is negatively related with the number of items in the candidate set.

In summary, adding the time variable into traditional algorithms fails to improve the novelty of recommendation results significantly, as proven by the same phenomena in follow-up experiments. Therefore, the experimental results of the TUCF and TCBR algorithms by combining time-aware unknown and difference are not introduced in the present study.

### 4.3 Time-aware unknown and difference

In the time-aware DI (TDI) algorithm, dendritcal clusters are formed first by a hierarchical clustering algorithm. Second, the PIA of each cluster is calculated by Eq. (7). In the TDI algorithm, the number of evaluated items in each type and the time of interaction behavior are considered. Hence, the TDI algorithm does not need to eliminate noise points but must maximize the quantity and time information provided by noise points. The evaluated items of users are clustered by the k-means algorithm, and the number of clusters is determined in accordance with the contour coefficient. The TDI algorithm is also called time-aware rated-item clustering (TRC) because it clusters the adopted items of the users.





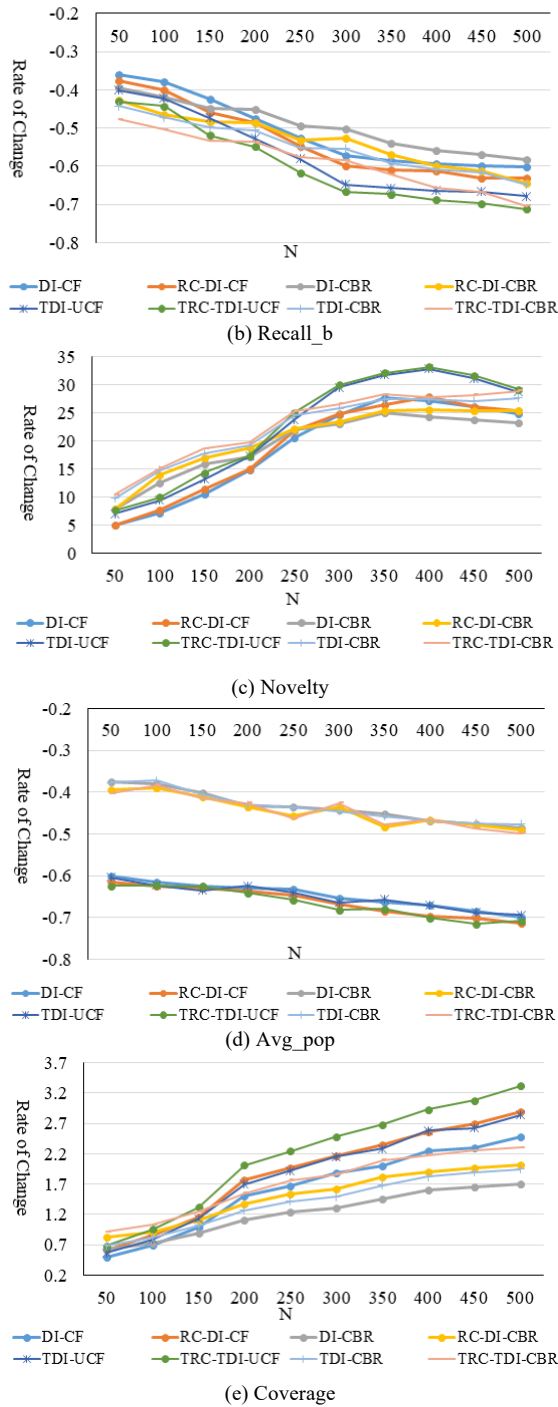


Fig. 6. Experimental results of the TUCF and TCBR algorithms using the result fusion strategy based on MovieLens

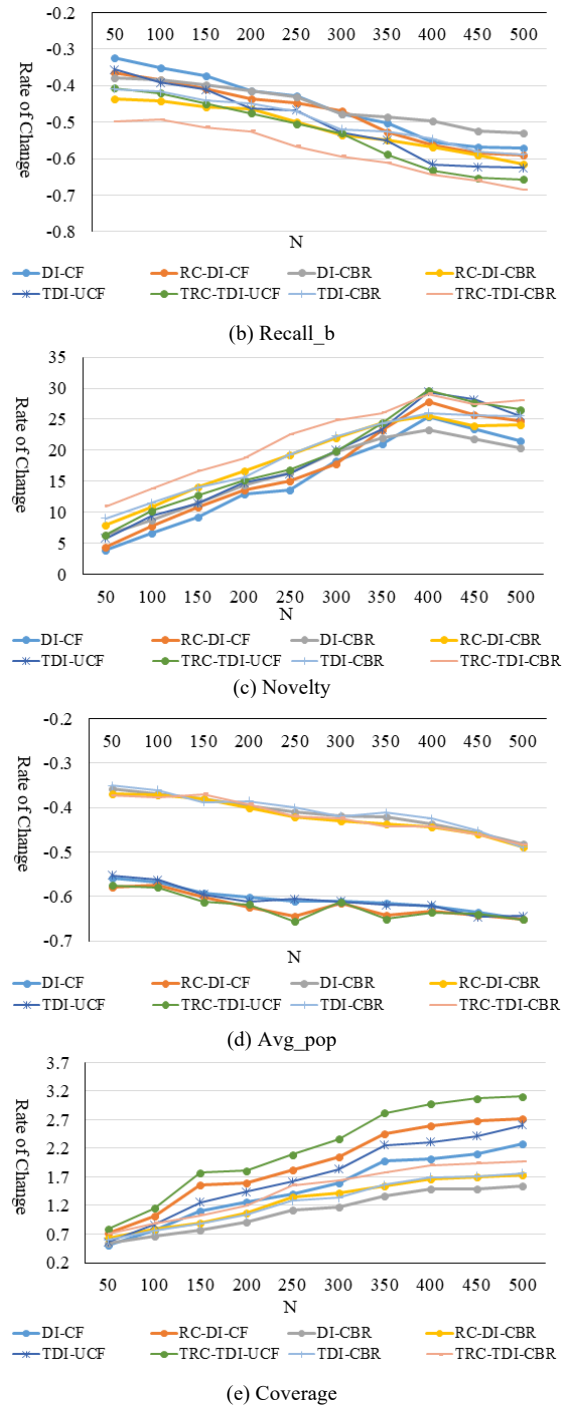
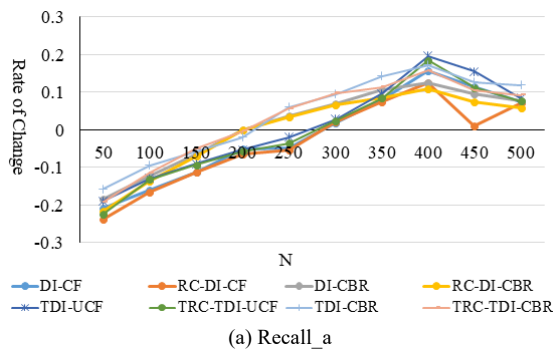


Fig. 7. Experimental results of the TUCF and TCBR algorithms using the result fusion strategy based on LastFM

The traditional UCF and CBR algorithms were combined with TDI and TRC algorithms to form the TDI-UCF, TRC-TDI-UCF, TDI-CBR, and TRC-TDI-CBR algorithms. The number of items in the recommendation list was 20, and N was the number of candidate sets. The experimental methods were the same. The experimental results of the novel recommendation of the TDI-UCF, TRC-TDI-UCF, TDI-CBR, and TRC-TDI-CBR algorithms are shown in Figs. 6 and 7.

The TDI and TRC algorithms further improve the novel recommendation. Compared with traditional algorithms, the TDI and TRC algorithms increase *Recall\_a* by approximately 25% but decrease *Recall\_b* by nearly 60%. Consequently, *Novelty* is increased by approximately 28

times, *Coverage* is improved by approximately 200%, and *Avg\_pop* is decreased by nearly 50%.

## 5. Conclusions

For users, the novelty of items is closely related with time. Hence, time information must be fully considered when studying novel recommendation. The proposed novel recommendation algorithms based on time-aware integrate time information into the calculation of user preference, unknown, and difference. First, a time attenuation parameter was introduced into the traditional algorithms to increase the weights of recent adopted items. Next, PIA was adjusted in accordance with time information. The difference of recommended items was calculated by combining the time of adopted items and the number of categories. Lastly, the novelty of recommended items was calculated by using the result fusion strategy, and Top-N items are recommended. In accordance with the experimental results, some major conclusions could be drawn:

(1) If only time-aware preference is considered, Novelty of the TUCF and TCBR algorithms reaches the peak when the number of recommended items is 20 in the DI algorithm. Novelty of the TUCF and TCBR algorithms approach that of traditional algorithms with the increase in the number of recommended items.

(2) With consideration to time-aware unknown and difference, novel recommendation is further improved by the TDI and TRC algorithms. Compared with traditional algorithms, *Recall\_a* and *Coverage* of the TDI and TRC algorithms increase by approximately 25% and nearly 200%,

respectively; *Recall\_b* and *Avg\_pop* decrease by nearly 60% and nearly 50%, respectively.

In this study, time information is added into the calculation of user preference, unknown, and difference of recommended items, thus effectively improving the novel recommendation of RSs. However, other situational information, including weather, geological position, and user emotions, can influence user satisfaction to recommendation results. Thus, how to collect and use such situational information to improve the performance of RSs must be studied further. The result fusion strategy only integrates the extensively used CF and CBR algorithms, thus restricting the applications of relevant algorithms. In future studies, time and other situational information can be combined, and additional algorithms must be used in fusion to improve the novelty and application situations of RSs.

## Acknowledgements

This work was supported by the Doctoral Research Project of Guizhou Normal University (Program Number: GZNUD[2017]36), the Innovation and Entrepreneurship Project of Guizhou Normal University, the 2018 Major Education Reform Project of Guizhou Normal University (Project Name: Innovation and Practice of E-commerce Talent Cultivation Mode with Deep Integration of Industry and Education in Guizhou Province, China), and the 2019 Provincial First-class Major in Marketing in Guizhou Province, China.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License



## References

- Jannach, D., Resnick, P., Tuzhilin, A., "Recommender Systems: Beyond Matrix Completion". *Communications of the ACM*, 59(11), 2016, pp.94-102.
- Zhang, L., Luo T., Zhang, F., and Wu, Y., "A recommendation model based on deep neural network", *IEEE Access*, 6, 2018, pp.9454-9463.
- Barberá, P., Jost, J.T., Nagler, J., Tucker, J.A., Bonneau, R., "Tweeting from left to right: is online political communication more than an echo chamber? ", *Psychological Science*, 26 (10), 2015, pp. 1531-1542.
- Del Vicario, M., Bessi, A., Zollo, et al., "The spreading of misinformation online", *Proceedings of the National Academy of Sciences*, 113(3), 2016, pp.554-559.
- Gravino, P., Caminiti, S., Sirbu, A., Tria, F., Servedio, V.D.P., Loreto, V., "Unveiling political opinion structures with a web-experiment", In: *1st International Conference on Complex Information Systems*, Rome, Italy, 22-24 April, 2016.
- Zhou, T., Kuscsik, Z., "Solving the apparent diversity/accuracy dilemma of recommender systems", *Proceedings of the National Academy of Sciences*, 107(10), 2010, pp. 4511-4515.
- Flodin, J., Lampka, K., Yi, W., "Experimental evaluation of context-dependent collaborative filtering using item splitting", *User Modeling and User-Adapted Interaction*, 24(1-2), 2014, pp.7-34.
- Zhao, Y., Wang, S., Wang, Y., "Stratified and time-aware sampling based adaptive ensemble learning for streaming recommendations", *Applied Intelligence*, 2020, <https://doi.org/10.1007/s10489-020-01851-9>.
- Noulapeu N. A., ElAyeb, W., Choukair, Z., "A time-aware service recommendation based on implicit trust relationships and enhanced user similarities", *Journal of Ambient Intelligence and Human Computing*, 2020, <https://doi.org/10.1007/s12652-020-02462-5>
- Chou, S.Y., Yang, Y. H., Jang, J. S. R., "Addressing cold start for next-song recommendation", In: *Proceeding of the 10th ACM Conference on Recommender Systems*, New York, ACM Press, 2016, pp.115-118.
- Chen L.J., Gao, J., "A trust-based recommendation method using network diffusion processes", *Physica A: Statistic Mechanic and Its Application*, 2018, <https://doi.org/10.1016/j.physa.2018.04.089>.
- Kapoor, K., Kumar, V., Terveen, L., et al., "I like to explore sometimes": Adapting to dynamic user novelty preferences", In: *Proceeding of ACM Conference on Recommender Systems*, New York: ACM press, 2015, pp.19-26.
- Ma, W. P., Feng, X., Wang, Sh.F., "Personalized recommendation based on heat bidirectional transfer", *Physica A: Statistical Mechanics & Its Applications*, 444, 2016, pp. 713-721.
- Hu, J.M., Lin, X., "Design of fusion recommendation algorithm for socialized users' minority based on the energy diffusion theory of heat spreading", *Information Studies: Theory & Application*, 39 (4), 2016, pp.119-123 .
- Wu, H., Cui, X.H., He, J., "On improving aggregate recommendation diversity and novelty in folksonomy-based social systems", *Personal & Ubiquitous Computing*, 18(8), 2014, pp.1855-1869.
- Wang, B., Cao, H., "Research on tourism recommendation model based on novelty and diversity", *Journal of Computer Engineering and applications*, 52(6), 2016, pp. 219-222.
- Yu, Q., Peng, Z.Y., Hong, L., Wang, Y.L., "A novel Web community recommendation approach based on user neighborhoods and topics", *Journal of Software*, 2016, 27(5):1266-1284.
- Pietro, G., Bernardo, M., Vittorio, L., "Towards novelty-driven recommender systems" . *Comptes Rendus Physique*, 20, 2019, pp.371-379
- Han, H.M., Dong, X.H., Zuo, C.H., "A weighted recommendation algorithm based on multiview clustering of user", *Journal of Intelligent & Fuzzy Systems*, 38(1), 2020, pp.441-451.
- Jiang, M. Y., Zhang, Z.F., Jiang, J.Q., "A collaborative filtering recommendation algorithm based on information theory and bi-clustering", *Neural Computing & Applications*, 31(12), 2019, pp.8279-8287.

21. Raja, D. R. K, Pushpa, S., "Diversifying personalized mobile multimedia application recommendations through the Latent Dirichlet Allocation and clustering optimization", *Multimedia Tools and Applications*, 78(17), 2019, pp.24047-24066.
22. Das, J., Majumder, S., Gupta, P., "Collaborative Recommendations using Hierarchical Clustering based on K-d Trees and Quadrees", *International Journal of Uncertainty Fuzziness and Knowledge-based Syatems*, 27(4), 2019, pp.637-668.
23. Mezni, H., Ait A., Benslimane, D., "An evolutionary clustering approach based on temporal aspects for context-aware service recommendation", *Journal of Ambient Intelligence and Humanized Computing*, 11(1), 2020, pp.119-138.
24. Selvi, C., Sivasankar, E., "A novel Adaptive Genetic Neural Network (AGNN) model for recommender systems using modified k-means clustering approach", *Multimedia Tools and Applications*, 78(11), 2019, pp.14303-14330.
25. Devarajan, M., Fatima, N. S., "Swarm intelligence clustering ensemble based point of interest recommendation for social cyber-physical systems", *Journal of Inteeligent & Fuzzy Systems*, 36(5), 2019, pp.4349-4360.
26. Liu. Z.B., Xiao. S.B., "Collaborative filtering algorithm based on user interest fuzzy clustering", *Journal of Modern book information technology*, 2015, 31(11):12-17.
27. Mohanty, S. N., Parvin, J. R., Kumar, K. V., "Optimal rough fuzzy clustering for user profile ontology based web page recommendation analysis", *Journal of Inteeligent & Fuzzy Systems*, 37(1), 2019, pp.205-216.
28. Zeng, Q., Cheng, X., "Research on network Collaborative filtering recommendation Algorithm based on similarity propagation and situation clustering", *Journal of Modern intelligence*, 2016, 36(11): 50-54.
29. Aliannejadi, M., Rafailidis, D. and Crestani, F. "A Joint Two-Phase Time-Sensitive Regularized Collaborative Ranking Model for Point of Interest Recommendation," *IEEE Transactions on Knowledge and Data Engineering*, 32(6), 2020, pp. 1050-1063.
30. Li, G.H., Chen Q., Zheng, B.L., "Time-aspect-sentiment Recommendation Models Based on Novel Similarity Measure Methods", *Acm Transactions on the Web*, 14(2), 2020, No.5, <https://doi.org/10.1145/3375548>
31. Chen, YC., Hui, L. & Thaipisutikul, T., "A collaborative filtering recommendation system with dynamic time decay", *The Journal of Supercomputing*, (2020). <https://doi.org/10.1007/s11227-020-03266-2>.
32. Zhang, J., Wang, Y.F., Yuan, Z.Y. and Jin Q., "Personalized Real-Time Movie Recommendation System: Practical Prototype and Evaluation", *Tsinghua Science and Technology*, 25(2), 2020, pp.180–191.
33. Zhang, J.Y. AND Lu, X.L., "A Multi-Trans Matrix Factorization Model With Improved Time Weight in Temporal Recommender Systems", *IEEE Access*, 8, 2020, pp. 2408-2416.
34. Zhao, H.X., Wu, H.CH., "Dtree2vec: A High-Accuracy and Dynamic Scheme for Real-Time Book Recommendation by Serialized Chapters and Local Fine-Grained Partitioning", *IEEE Access*, 8, 2020, pp.23197-23208
35. Baeza-Yates, RA., Ribeiro-Neto, B., "Modern Information Retrieval", Addison Wesley, 2011, 43(1):26–28.
36. Berkovsky, S., Freyne, J., Coombe, M., "Aggregation trade offs in family based recommendations", In: *Australasian Joint Conference on Advances in Artificial Intelligence*, Melbourne, Australia, Springer, 2009, pp.646–655.
37. Zhang, L., Qian, X.SH., Lv, P., Zhou, X., "A novel recommendation algorithm based on diffusion of innovation theory", *Journal of Engineering Science and Technology Review*, 12(6), 2019, pp.87-95.
38. Zhang, L., Liu, X.J. and Zhou, X., "A Novel Recommendation Algorithm Based on Clustering Dissimilarity Measures", *Journal of Engineering Science and Technology Review*, 13 (3), 2020, pp. 88 – 95
39. Zhang, L., Peng, L.F., Phelan, C.A., "Novel recommendation of user-based collaborative filtering", *Journal of Digital Information Management*, 12(3), 2014, pp.165-175.