01010
01010
01010 *information*

MDPI

Article

Tag-Driven Online Novel Recommendation with Collaborative Item Modeling

Fenghuan Li, Zhaosheng Lin and Zhenyu Wang *

School of Software Engineering, South China University of Technology, Guangzhou 510006, China; lifenghuan2007@163.com (F.L.); l.zhaosheng@mail.scut.edu.cn (Z.L.)

* Correspondence: wangzy@scut.edu.cn; Tel.: +86-20-3938-0332

Received: 10 February 2018; Accepted: 3 April 2018; Published: 5 April 2018

check for updates

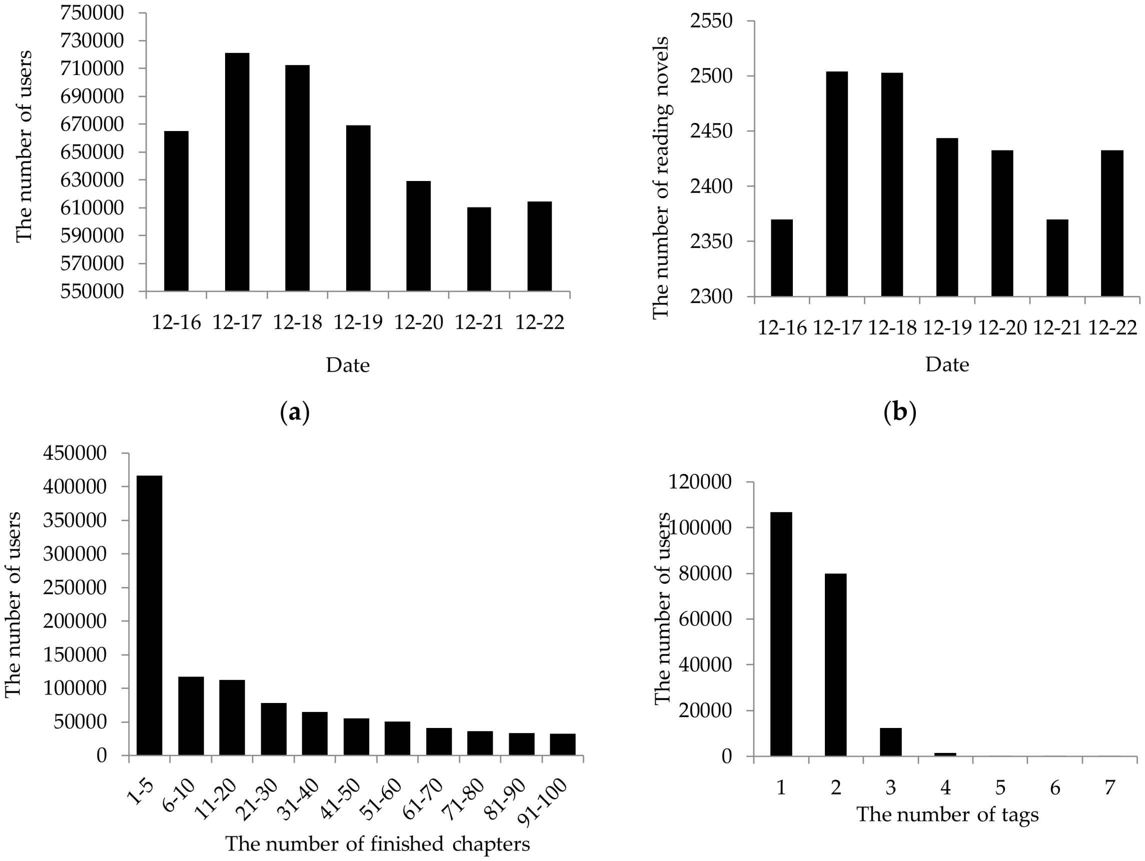
Abstract: Online novel recommendation recommends attractive novels according to the preferences and characteristics of users or novels and is increasingly touted as an indispensable service of many online stores and websites. The interests of the majority of users remain stable over a certain period. However, there are broad categories in the initial recommendation list achieved by collaborative filtering (CF). That is to say, it is very possible that there are many inappropriately recommended novels. Meanwhile, most algorithms assume that users can provide an explicit preference. However, this assumption does not always hold, especially in online novel reading. To solve these issues, a tag-driven algorithm with collaborative item modeling (TDCIM) is proposed for online novel recommendation. Online novel reading is different from traditional book marketing and lacks preference rating. In addition, collaborative filtering frequently suffers from the Matthew effect, leading to ignored personalized recommendations and serious long tail problems. Therefore, item-based CF is improved by latent preference rating with a punishment mechanism based on novel popularity. Consequently, a tag-driven algorithm is constructed by means of collaborative item modeling and tag extension. Experimental results show that online novel recommendation is improved greatly by a tag-driven algorithm with collaborative item modeling.

Keywords: online novel recommendation; tag-driven; latent preference; rating prediction; item-based collaborative filtering

1. Introduction

A recommendation system is a capable application for presenting a suggestion to a user, according to his previous preferences, as well as those of the community with similar likings and opinions to him. Recommendation Systems help us reduce the information overload that we suffer nowadays and provide customized information for a specific domain. Recommendation systems have provided significant assistance for numerous applications such as music [1], movie [2], TV program [3], electronic books [4] and so forth. In particular, book recommendation systems have been widely used in many online stores, such as Amazon, JD, Dangdang and so forth and there is extensive literature regarding book recommendation [4–6]. Crespo et al. [4] proposed a recommendation-based solution for the case of intelligent electronic books using gathered data from user interaction. Paula et al. [5] presented a hybrid recommendation system to help readers decide which book to read by combining two item-based collaborative filtering algorithms. Anand et al. [6] introduced a book recommendation system by using the ideas of content filtering, collaborative filtering and association rule mining.

The data used for book recommendation can be divided into two categories: user behavior and book content. User behaviors, which include user ID, book ID, book tag, browsing history and so forth, are commonly analyzed in collaborative filtering algorithms. The data distribution of user behaviors in the 3G literary community (http://www.3gsc.com.cn/) during 16–22 December 2016 is presented in Figure 1. It is shown that there are about 650 thousand active users (Figure 1a) and 2500 novels read (Figure 1b) every day. Online novels are read and updated in the light of chapters. This is different from traditional book marketing. Because the preference for novels can be expressed by the number of chapters precisely, chapter-related features are extremely important for online novel recommendation. Meanwhile, the tag of a novel describes the category of its content, such as "suspense", "metropolis" and "fantasy". There is a single tag on each novel. A user's interest in novels can be reflected by their tags. The data in the 3G literary community shows that the number of users decreases with increasing number of chapters (Figure 1c) and novel tags change slightly for the majority of users (Figure 1d), indicating that the interest of a certain reader remains stable over the short term.



(**a**) (**b**)

(**c**) (**d**)

Figure 1. The data distribution of user behaviors in the 3G literary community (http://www.3gsc.com.cn/) during 16–22 December 2016. (**a**) The number of active users; (**b**) The number of read novels; (**c**) The number of users with increasing number of finished chapters; (**d**) The number of users against the number of novel tags.

There are large quantities of users, therefore, a collaborative filtering approach by means of collective intelligence can be feasible for online novel recommendation. Popular algorithms of this approach are item-based and user-based recommendations. In the circumstances of a great number of users, user-based algorithms will be time-consuming and lead to data sparsity. In contrast, item-based algorithms are effective with relatively fewer items. Therefore, an item-based collaborative filtering algorithm incorporating tags is studied in this paper. It is feasible for online novel recommendation because of the following reasons:

1. There are many more users than novels according to user behavior analysis in the 3G literary community, therefore, item-based algorithm is more appropriate than user-based algorithm for practical application considering data sparsity.
2. There are few novels that have been read by one user in a certain period, leading to a sparse rating matrix and low accuracy of similarity calculation for a user-based algorithm. Therefore, it is difficult to find groups with similar interests. In contrast, the rating matrix of an item-based algorithm is denser than that of a user-based algorithm, leading to more accurate similarity calculation.
3. A user's interest in a short term remains stable according to user behavior analysis. In other words, every user has his special preference. In consideration of individualization, novel tag is regarded as an important determinant and an item-based collaborative filtering algorithm is more feasible.

The paper is organized as follows: Section 2 reviews some prior studies relevant to this study; Section 3 illustrates how to rate latent preference; Tag-driven recommendation with collaborative item modeling is described in Section 4; We perform extensive evaluation and discuss important results in Section 5 to demonstrate the effectiveness and scalability of the proposed algorithm; Finally, we conclude this study and present some future directions in Section 6.

2. Related Works

Personalized recommendation the recommendation of attractive items according to the preferences and characteristics of users or items, which are key information for the recommendation. Both preferences and characteristics can be divided into two categories: explicit and implicit. Item characteristics (film's director, actor, name, etc.) and user characteristics (gender, age, etc.) are explicit. The content of books and television programs, audio features and user behaviors are implicit. Implicit preferences and characteristics should be converted to explicit information. Núñez-Valdéz et al. [7] introduced the technology that converted certain specific behaviors to explicit user feedback. Choi et al. [8] predicated a user's ratings on items according to a given user's purchase behaviors. Moshe et al. [9] suggested an approach that was centered on representing environmental features as low dimensional unsupervised latent contexts. The latent contexts were automatically learned for each user utilizing unsupervised deep learning techniques and principal component analysis.

Recommendation algorithms are generally classified into collaborative filtering (CF) [10,11] and content-based filtering (CB) [12]. In general, CF uses an information filtering technique based on a user's previous evaluation of items or purchase history. However, this technique has been known to reveal major issues, such as sparsity, scalability, cold-start, the Matthew effect and so forth. Typical algorithms of CF include item-based and user-based CF. Their characteristics are shown in Table 1. In contrast, CB analyzes a set of documents rated by an individual user and uses the content of documents as well as provided ratings to infer a user profile that can be used to recommend additional items of interest. However, the syntactic nature of CB, which detects similarities between items with same attributes or characteristics, causes overspecialized recommendations, only including items very similar to those of which the user is already aware [13]. Over the last decade, great efforts have been made to solve these problems, such as knowledge-based recommendation [14], association rules [15], hybrid approach [16,17] and so forth. To address the data sparsity issue of classic CF, Fang [18] introduced an improved algorithm based on a sigmoid function. Different items were modeled with a sigmoid function in order to capture their popularity, while different users were modeled to map ratings into preferences. Predictions were made according to the phenomenon that preferences should keep consistent with popularity. Content-based filtering and collaborative filtering may be combined in a hybrid approach that helps to overcome the limitations of each method. Albanese et al. [16] proposed a multimedia recommender system based on modeling recommendations as a social choice problem, which incorporated some content-based characteristics into a collaborative strategy. This hybrid strategy was able to address both the cold start problem and the anonymous user issue.

Meanwhile, some extended algorithms were implemented in real world situations [19–22]. Yu [23] proposed to use entropy to describe distribution characteristics of user's ratings and measured the contribution of neighbor recommendation by the difference between rating distributions. Then double criteria were used to calculate neighbor recommendation weights, achieving performance improvement. In order to find a better way to depict user preferences and make the algorithm more suitable for personalized recommendation, Zhang et al. [24] introduced a framework that utilized item domain features to construct user preference models and combined these models with collaborative filtering. The framework not only integrated domain characteristics into a personalized recommendation but also aided implicit relationships detection among users, which were missed by the conventional CF method. San et al. [25] proposed a co-authorship, network-based, task-focused literature recommendation technique to meet users' information needs specific to a task under investigation and developed different schemes for estimating the closeness between scholars based on their co-authoring relationships. Francesco et al. [26] proposed a novel collaborative user-centered recommendation approach, in which several aspects related to users and available in online social networks were considered and integrated together with items' features and context information within a general framework that can support different applications using proper customizations.

Table 1. Summary of collaborative filtering algorithms.

Task	User-Based CF	Item-Based CF
preprocessing	similarity matrix construction between users	similarity matrix construction between items
recommendation	top k users selection that have rated the item and all rating predictions	top k items selection that have been rated by an active user and all rating predictions
personalization	dependent on the group with similar interests and inclined to socialization	dependent on user's historical preference and inclined to individualization
adaptability	time-consuming with a great number of users and effective with relatively few users	time-consuming with a great number of items and effective with relatively few items

Although collaborative filtering is one of the most successful techniques, it is becoming increasingly difficult for users to find the most attractive content and users often struggle with a great challenge in terms of information overload. Jadhav [27] proposed a decision-tree-based framework, which used an efficient classification algorithm combined with collaborative recommendation approach for book recommendation. A decision tree classifier was applied on all transactions and collaborative filtering was performed on user profile records. Finally, two results were combined into one with maximum confidence values. However, this recommendation system worked in an offline mode and stored recommendations in the buyer's web profile, which is significantly different from a social media service. Proper recommendations require abundant information that includes characteristics, preferences, needs and so forth. This information is represented by the ratings of users on a set of items. Besides ratings, item characteristics are allowed, such as tags, which are rich and compact enough to characterize the same main concepts of items. Due to these challenges, a number of studies have tried to combine recommendation systems with tags [28,29]. By leveraging user-generated tags as preference indicators, Kim et al. [28] proposed a new collaborative approach to user modeling that can be exploited to recommender systems. This model provided a better representation in user interests and achieved better recommendation results.

3. Latent Preference Rating

Predicting a user's rating of a new item depends on their ratings of previous items. Generally, there are binary rating strategies and fine-grained rating strategies. In traditional book marketing, a user's rating on a book is binary rating generally, corresponding to "have bought" or "haven't bought". This is not very applicable in our scenario. Suppose that one user bought one book, he may like it and finish reading it. Otherwise, he may not like it and give up after reading several chapters.

The ratings in these two situations should be different. However, only purchase time can be acquired in the website, without reading time. What's more, a user's interest in a book will change over time. After one user bought one book, there are two situations. If one finishes reading the book (or just gives it up) in a short term, the interest will decrease over time. The other situation is that one is reading this book continuously and slowly, illustrating that there is still a strong interest. This difference cannot be distinguished in traditional book recommendation. Online novel is updated in chapters, providing more detailed behaviors. However, it does not require readers to rate novels online generally. Therefore, there is no explicit preference degree of a particular user on a novel. A user's rating can only be modeled by latent preference in behaviors.

The preference degree of one user on a novel cannot be expressed by finished chapters directly. One popular method is to use the proportion of finished chapters in all chapters. Because an online novel is updated continually, this method is not applicable to online novel. Update speed is determined by authors; therefore, the number of finished chapters is not accurate to evaluate a user's preference degree. Considering this reason, preference degree of one user on a novel is proposed in Formula (1).

$$R(u,i) = \frac{C(u,i)}{C_{\max}(i)} \tag{1}$$

where, $C(u,i)$ is the number of finished chapters of novel i that user u reads, $C_{\max}(i)$ is the maximal number of finished chapters in all users that read the same novel i. If there are few updated chapters (for example 5 chapters), rating of one user who just has finished reading 2 or 3 chapters is also high. Therefore, new novels should be considered in Formula (1), whose confidence is judged by the number of updated chapters, as Formula (2). Where, x_i is the number of updated chapters of novel i, α and β are shape parameters. The smaller x_i is, the smaller $S(i)$ is, leading to lower confidence of Formula (1).

$$S(i) = \frac{1}{1 + e^{-(x_i - \alpha)/\beta}} \tag{2}$$

In addition to the finished chapters, spent time is also an important factor for a user's rating. A user's preference may change over time; therefore, new interests should be captured for accurate recommendations. For the same novel, if last reading time is close to current time, the interest is strong, otherwise, the interest is weak and the user's rating is low. Interest decay is expressed by the Weibull decay function in Formula (3).

$$W(u,i) = e^{-(\frac{t_{u,i}}{T_i})^k \times \log 2} \tag{3}$$

where, $t_{u,i}$ is the number of days from the latest date when user u read novel i to the present. T_i is the total time interval from the issuance of novel i. k is a shape parameter. When k = 1, it is exponential decay. Consequently, the rating of one user on one novel is determined by Formula (4), according to above analysis. The situation that recommendation performance is affected by the lack of preference degree is solved.

$$P(u,i) = \frac{C(u,i) e^{(x_i - \alpha)/\beta}}{C_{\max}(i)(1 + e^{(x_i - \alpha)/\beta}) e^{(\frac{t_{u,i}}{T_i})^k \times \log 2}} \tag{4}$$

If one item is popular, it will occur in a great quantity of users' recommendation lists—namely, the Matthew effect. It means that the more popular the item is, the more likely that it is recommended, otherwise, the more unheeded it is. This situation frequently occurs in collaborative filtering algorithms, as well as online novel recommendation. Personalized recommendation will be neglected, leading to more and more unheeded novels and worse long tail effects. To address this problem, a punishment mechanism based on novel popularity is proposed shown in Formula (5). Where U_i is the number of users that have read novel i and $\log(1 + U_i)$ is a penalty function. The more popular the novel is, the higher the penalty degree is. The weight of unheeded novel is enhanced; therefore, recommendation probability is increased.

$$P_{pun}(u,i) = \frac{P(u,i)}{\log(1+U_i)} \tag{5}$$

Popular novels are punished by Formula (5). Recommendation coverage and novelty are strengthened. However, recommendation accuracy is improved insufficiently for some users. There are three reasons in online novel recommendation: (1) The themes appeal to majority users; (2) The qualities of novels are satisfactory; (3) There are enough updated chapters (there are few users when one novel is new). For lots of users, popular novels are preferable choices. Weights of all popular novels are reduced by the punishment in Formula (5). That is to say, recommendation probabilities of popular novels are reduced for all users. However, unread popular novels may be attractive to users who prefer popular novels. Therefore, weights cannot be reduced for all users. Popular novels should be recommended to users who prefer popular novels, while recommendation probabilities of popular novels should be reduced for users who show little interest in popular novels.

In light of the above analysis, users are classified into two categories according to the preference for popular novels. For users who prefer popular novels, more popular novels are recommended to them, according to the rating prediction determined by Formula (4). For those who show little interest in popular novels, rating prediction is determined by Formula (5). Correspondingly, less popular novels are recommended. Recommendation accuracy of the true preference is improved, meanwhile, coverage and novelty are enhanced.

For small-scale data, the similarity between novel i and j can be calculated by the Jaccard similarity coefficient directly by Formula (6). Where U_i or U_j is the number of users that have read novel i or j respectively. But, when dealing mass data, time and space complexity will be increased sharply. The goal of MinHash is to estimate $J(U_i,U_j)$ quickly, without explicitly computing the intersection and union. Let h be a hash function that maps the members of U_i to distinct integers and $h_{\min}(U_i)$ be the minimal member of U_i with respect to h. For $h_{\min}(U_i)$ and $h_{\min}(U_j)$, the same value is got when the elements of the union $U_i \cup U_j$ with minimum hash value lie in the intersection $U_i \cap U_j$. The probability of this situation is the ratio in Formula (6), as shown in Formula (7). That is to say, the probability that $h_{\min}(U_i) = h_{\min}(U_j)$ is true is equal to the similarity $J(U_i,U_j)$. The idea of the MinHash scheme is applied to be a useful estimator for the Jaccard similarity between novels and reduces calculation time.

$$Nsim_{ij} = J(U_i, U_j) = \frac{|U_i \cap U_j|}{|U_i \cup U_j|} \tag{6}$$

$$P(h_{\min}(U_i) = h_{\min}(U_j)) = J(U_i, U_j) \tag{7}$$

4. Tag-Driven Recommendation with Collaborative Item Modeling

There are broad categories of items in an initial recommendation list by means of item-based collaborative filtering. Nevertheless, the interests of a majority of users remain stable over the short term. It is very likely that there are many inappropriate novels in the initial recommendation list. This issue should be tackled according to novel tags, which can reflect a user's preference. Novels with the same tag can be recommended to users in order to extend reading scope and to mine latent interests.

Tag-driven recommendation with collaborative item modeling (TDCIM) is proposed. The detailed architecture is shown in Figure 2. TDCIM combines the tag-driven recommendation with improved item-based collaborative filtering to build a hybrid recommendation system. Both algorithms to be combined belong to item-based collaborative filtering. Improved item-based collaborative filtering is based on the novel itself, while tag-driven recommendation is based on the novel tag. Basically, the proposed architecture consists of a three-step process. The first step is an item-based collaborative filtering to predict a user's novel preference and initial recommendation list, which is improved by latent preference rating and the MinHash scheme proposed in Section 3. The second step is a tag-driven process, which is to find the target novel tag's neighbor community set and select top K

neighbors of each tag, then predict a user's tag preference. The third step is tag extension by means of a correspondence relationship between novels and tags, as well as each user's initial recommendation list.

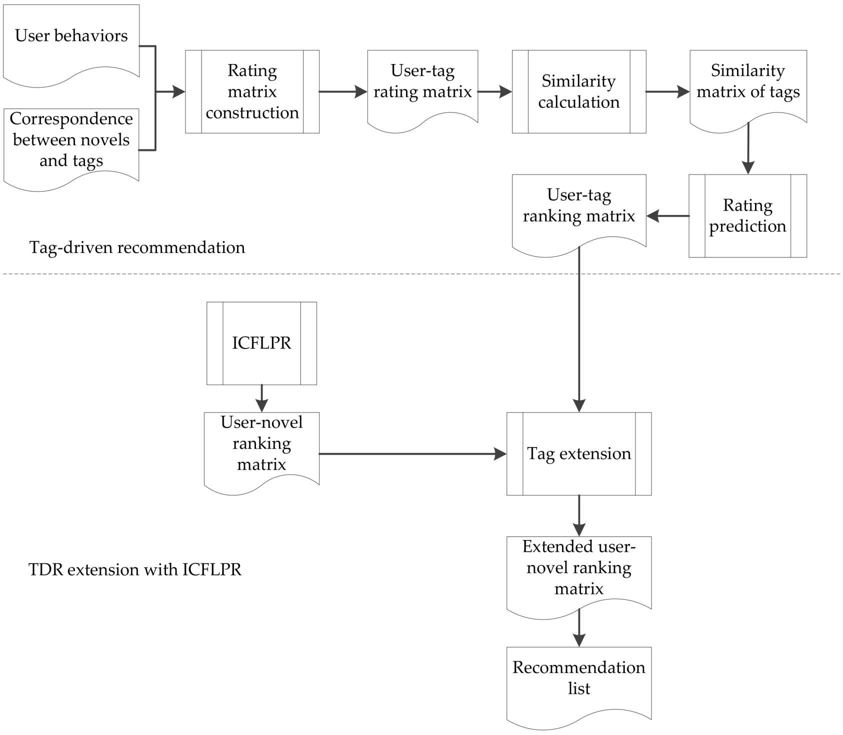

User behaviors
Correspondence between novels and tags
Rating matrix construction
User-tag rating matrix
Similarity calculation
Similarity matrix of tags
Rating prediction
User-tag ranking matrix
Tag-driven recommendation
ICFLPR
User-novel ranking matrix
Tag extension
Extended user-novel ranking matrix
TDR extension with ICFLPR
Recommendation list

Figure 2. Tag-driven recommendation with collaborative item modeling.

Tag-driven recommendation (TDR) can lower the rankings of uninterested categories and promote those of interested categories. The tag-driven process consists of user-tag rating matrix construction, similarity calculation and rating prediction. The user-tag rating matrix is constructed by a user's historical behaviors as well as by the correspondence between novels and tags, as shown in Figure 3. tag_j is a novel tag and $user_u$ is a user ID. Every r_{uj} denotes the preference degree of $user_u$ on tag_j, which is determined by the number of novels with tag_j that $user_u$ has read. Then, similarities between tags shown in Figure 4 are calculated by Formula (8), where $R = \{u | r_{uk} \neq 0, r_{uj} \neq 0\}$, $\overline{r_j}$ is average preference degree of tag_j. $Tsim_{kj}$ is the similarity between tag_k and tag_j. Top K neighbors of each tag are selected according to similarities between tags. Because the tag is not final consumable item but an abstraction, the tag itself will be reserved. This is different from traditional neighbor selection and the rankings of novels with same tag are promoted in initial recommendation lists.

$$Tsim_{kj} = \frac{\sum_{u \in R} (r_{uk} - \overline{r_k})(r_{uj} - \overline{r_j})}{\sqrt{\sum_{u \in R} (r_{uk} - \overline{r_k})^2} \sqrt{\sum_{u \in R} (r_{uj} - \overline{r_j})^2}} \tag{8}$$

$$PT_{uk} = \frac{\sum\limits_{tag_j \in (NT_j \cap RT_u)} Tsim_{kj} r_{uj}}{\sum\limits_{tag_j \in (NT_j \cap RT_u)} Tsim_{kj}} \tag{9}$$

The prediction of $user_u$'s rating on tag_k is determined by Formula (9) after tag neighbor selection, where NT_j is top K neighbors of tag_j. $RT_u = \{tag_j | r_{uj}! = 0\}$ is the set of tags read by $user_u$. The ratings are ordered and converted to rankings, achieving the user-tag ranking matrix and user's tag preference. There are different rating metrics in different algorithms. This situation can be solved for tag extension by rankings instead of ratings.

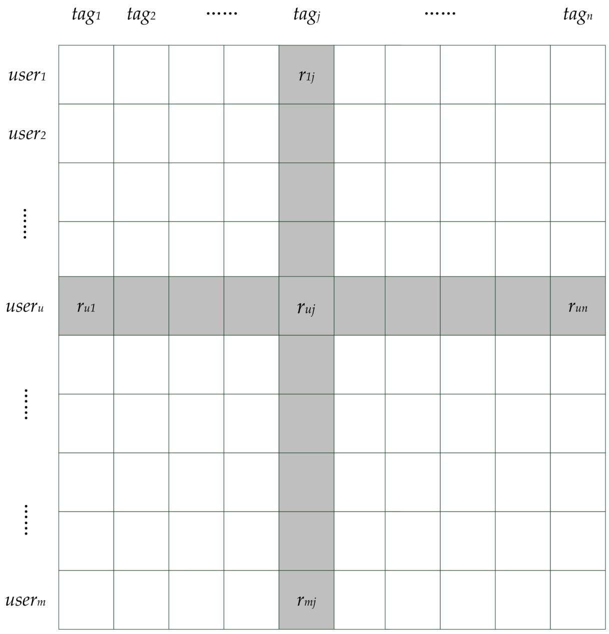


Figure 3. User-tag rating matrix.

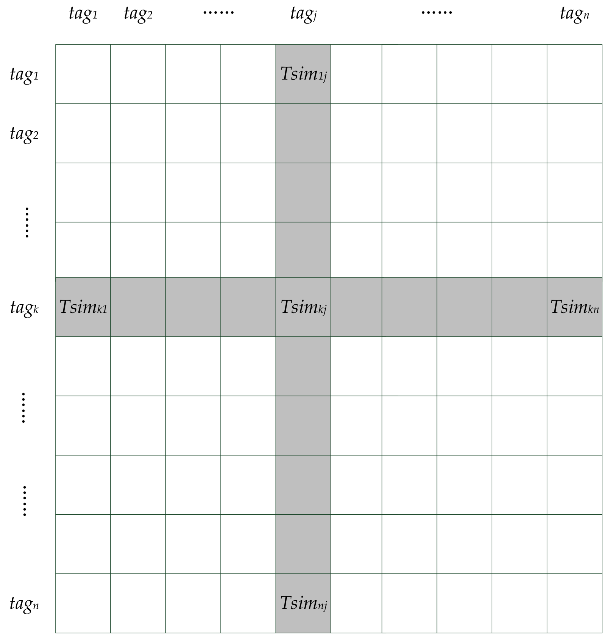


Figure 4. Similarity matrix of tags.

One user-novel ranking matrix and an initial recommendation list are obtained by improved item-based collaborative filtering algorithm with latent preference rating (ICFLPR) in Section 3. The ranking matrix calculated by TDR is extended with the one calculated by ICFLPR employing the schedule in Figure 2, resulting in tag-driven recommendation with collaborative item modeling. User-tag ranking matrix is extended to the user-novel ranking matrix when predicting a user's rating of novels with the same tag, by means of a correspondence relationship between novels and tags, as well as each user's initial recommendation list. For example, the tag of novels nov_1 and nov_2 is tag_1 and the tag of novel nov_5 is tag_3. Given $user_u$'s initial recommendation list $\{nov_1, nov_5, nov_2\}$, the rankings of nov_1, nov_5 and nov_2 are 1, 2 and 3. If $user_u$'s tag ranking vector is $\{tag_3, tag_1\}$, the rankings of nov_1 and nov_2 are 2 and the ranking of nov_5 is 1 in extended user-novel ranking matrix.

5. Experiments and Discussions

5.1. Experimental Setup

The proposed algorithm is evaluated with the dataset consisting user's reading behaviors over five days, which is crawled from the 3G literary community (http://www.3gsc.com.cn/). The dataset is divided into a training set and a test set. Reading behaviors are merged. That is to say, there is only one behavior when one user reads one novel. The lack of reading behaviors may lead to inaccurate evaluation. To avoid this situation, users with less than 4 behaviors are ignored. Behaviors of each user are sorted in chronological order. The last 2 behaviors are regarded as the test data. Remaining behaviors are training data. The length of the recommendation list is assigned to be 5. After preprocessing, the numbers of users, novels and tags are 15,236, 1804 and 13, respectively.

Precision, recall, F1, hit ratio, coverage and running time are used to evaluate the performance. Hit rate is the proportion of users whose reading novels are recommended successfully in all users. When hit rate is larger, retention rate and the dependence of users are enhanced. It is also a crucial metric for recommendation evaluation. Coverage is the proportion of recommended novels in all novels. The user may only read one novel in a short term. Reading behaviors should be considered as many as possible, so that some novels in the recommendation list can be selected by the user. This user will stay on in the website and user retention rate can be improved. What's more, the number of reading behaviors is smaller than the length of recommendation list in test set. In this case, precision maximum is dependent on the number of reading behaviors. For example, one user only has two behaviors but the length of recommendation list is 5. The upper limit of precision will be 40%. Therefore, recall, hit ratio, coverage and running time are dominant, while precision and F1 are auxiliary for performance evaluation.

5.2. Impact of Tag Neighbors

The influence of tag neighbors on tag-driven recommendation with collaborative item modeling is investigated. Different numbers of tag neighbors are employed to evaluate precision, recall, F1, hit ratio and coverage. Those performances with different tag neighbors and novel neighbors are listed in Figures 5–9. It is shown that the variation tendency of every evaluation criterion is similar when the number of novel neighbors is different. The variation tendencies of precision, recall and F1 with respect to increasing number of tag neighbors are similar in this experiment. The performance is significantly different when the number of tag neighbors is different. Therefore, the novel tag is an important determinant for online novel recommendation.

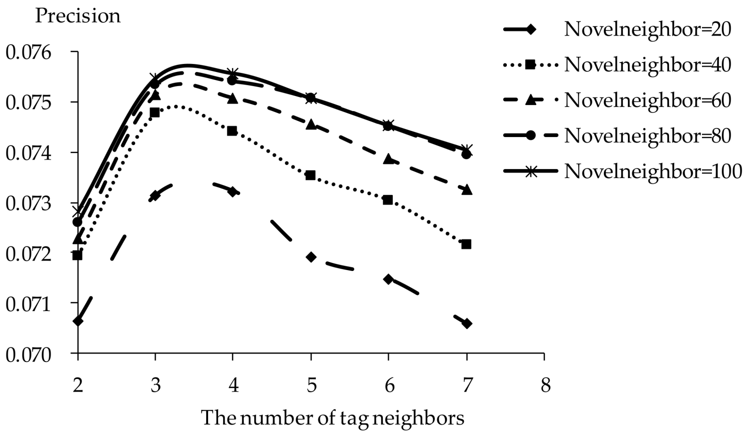


Figure 5. Precision of tag-driven algorithm with collaborative item modeling (TDCIM) with respect to increasing number of tag neighbors.

If there are few tag neighbors, collective intelligence cannot contribute. Therefore, when there are fewer than 3 tag neighbors, recall and hit ratio increase with more tag neighbors in Figures 6 and 8. Otherwise, when there are more than 4 tag neighbors, the curve shows a negative correlation between recall and the number of tag neighbors. Similarities between neighbors in significantly different tags become smaller with increasing number of tag neighbors. The decrease of neighbor representativeness results in worse recall. There are similar situations with precision, F1 and hit ratio. More tag neighbors mean that more different kinds of novels are considered. Therefore, coverage becomes better and more stable with increasing number of tag neighbors. When the number of tag neighbors is 4, coverage performs excellently compared to best results. Therefore, the number of neighbors is defined as 4 in our experiments. The results show that one reader's interest over the short term remains stable.

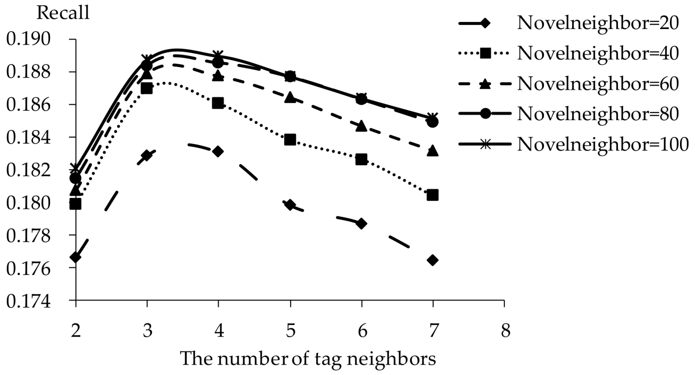


Figure 6. Recall of TDCIM with respect to increasing number of tag neighbors.

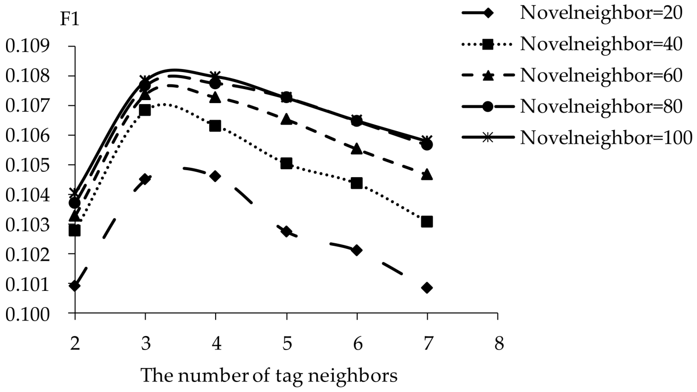


Figure 7. F1 of TDCIM with respect to increasing number of tag neighbors.

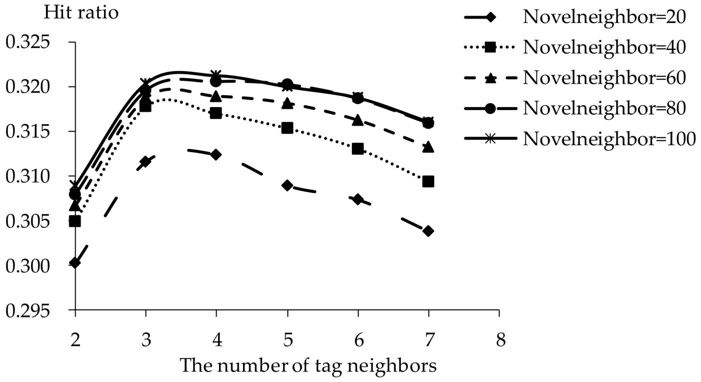


Figure 8. Hit ratio of TDCIM with respect to increasing number of tag neighbors.

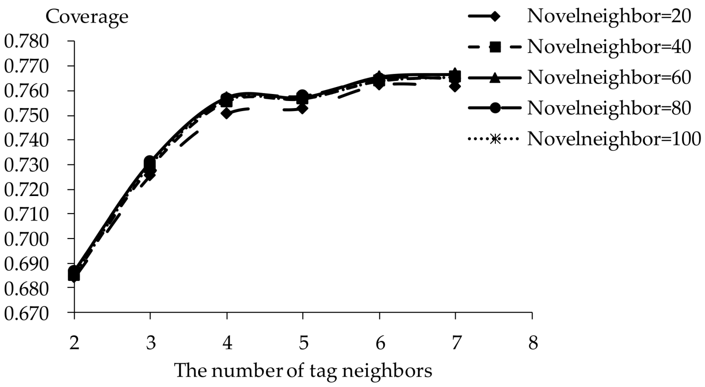


Figure 9. Coverage of TDCIM with respect to increasing number of tag neighbors.

5.3. Impact of Novel Neighbors

The influence of novel neighbors on tag-driven recommendation with collaborative item modeling is investigated. Different numbers of novel neighbors are employed to evaluate precision, recall, F1, hit ratio and coverage. Those performances with different novel neighbors are shown in Figures 10–14 respectively. Meanwhile, the performance is best when the number of tag neighbors is 3 or 4, therefore, only the performances when the number of tag neighbors is 3 and 4 are listed in Figures 10–14. It is shown that the performance is significantly different when the number of novel neighbors is different and variation tendency of every evaluation criterion is similar when the number of tag neighbors is different. The variation tendencies of precision, recall and F1 with respect to increasing number of novel neighbors are similar in this experiment.

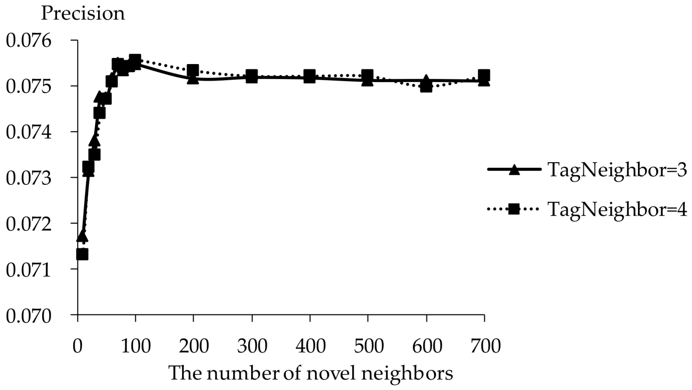


Figure 10. Precision of TDCIM with respect to increasing number of novel neighbors.

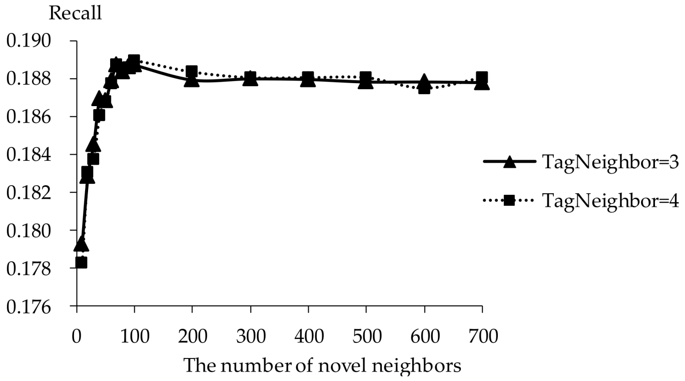


Figure 11. Recall of TDCIM with respect to increasing number of novel neighbors.

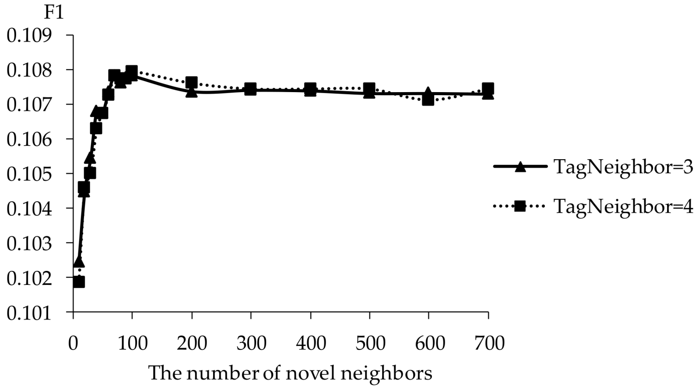


Figure 12. F1 of TDCIM with respect to increasing number of novel neighbors.

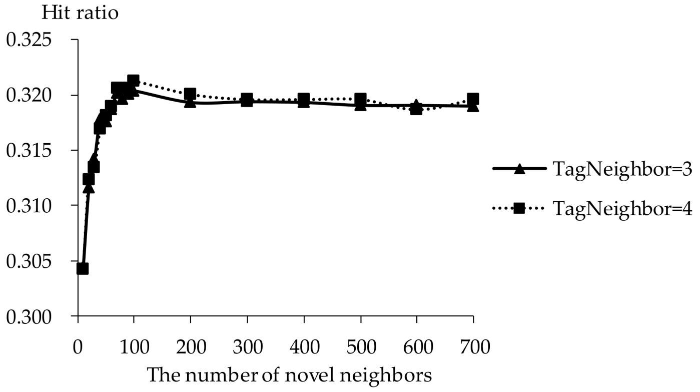


Figure 13. Hit ratio of TDCIM with respect to increasing number of novel neighbors.

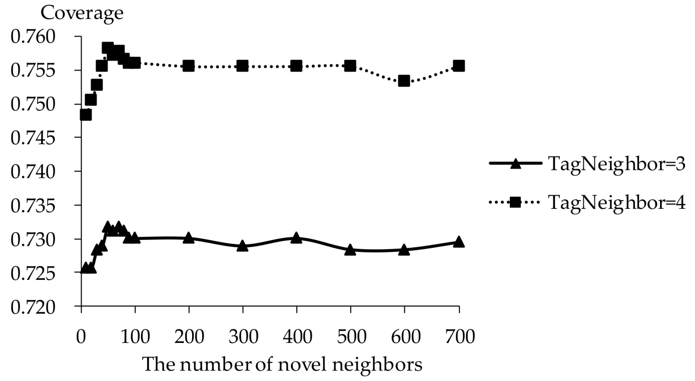


Figure 14. Coverage of TDCIM with respect to increasing number of novel neighbors.

It is shown that the performance is best in terms of every evaluation criterion when the number of novel neighbors is about 100. If there are few novel neighbors, collective intelligence cannot contribute. Therefore, when there are fewer than 100 novel neighbors, the recall becomes better with increasing number of novel neighbors. Otherwise, when there are more than 100 novel neighbors, the curve becomes stable. More novel neighbors mean that more novels in long tail and more different novels are considered. Due to latent preference rating with the punishment mechanism, popular novels are punished and unheeded novels are enhanced, therefore, Matthew effect and long tail effect do not become worse but lead to stable performance. There are similar situations on precision, F1, hit ratio and coverage. Meanwhile, it will be time-consuming to analyze much more novel neighbors. Therefore, the number of novel neighbors is defined as 100.

5.4. Effectiveness

Tag-driven recommendation with collaborative item modeling (TDCIM) and improved item-based CF with latent preference rating (ICFLPR) are compared against item-based CF(ICF) [30], user-based CF (UCF) [31] and tag-item (TI) [32] algorithms. Item-based CF and user-based CF are commonly used in recommendation systems in general. The tag-item algorithm is related to a specific recommendation system in which the tags of items (such as movie, book, music, etc.) have been labeled. An improved item-based CF with latent preference rating is used in specific novel recommendation systems. Precision, recall, F1, hit ratio and coverage are measured for each algorithm. Detailed experimental results are shown in Figure 15.

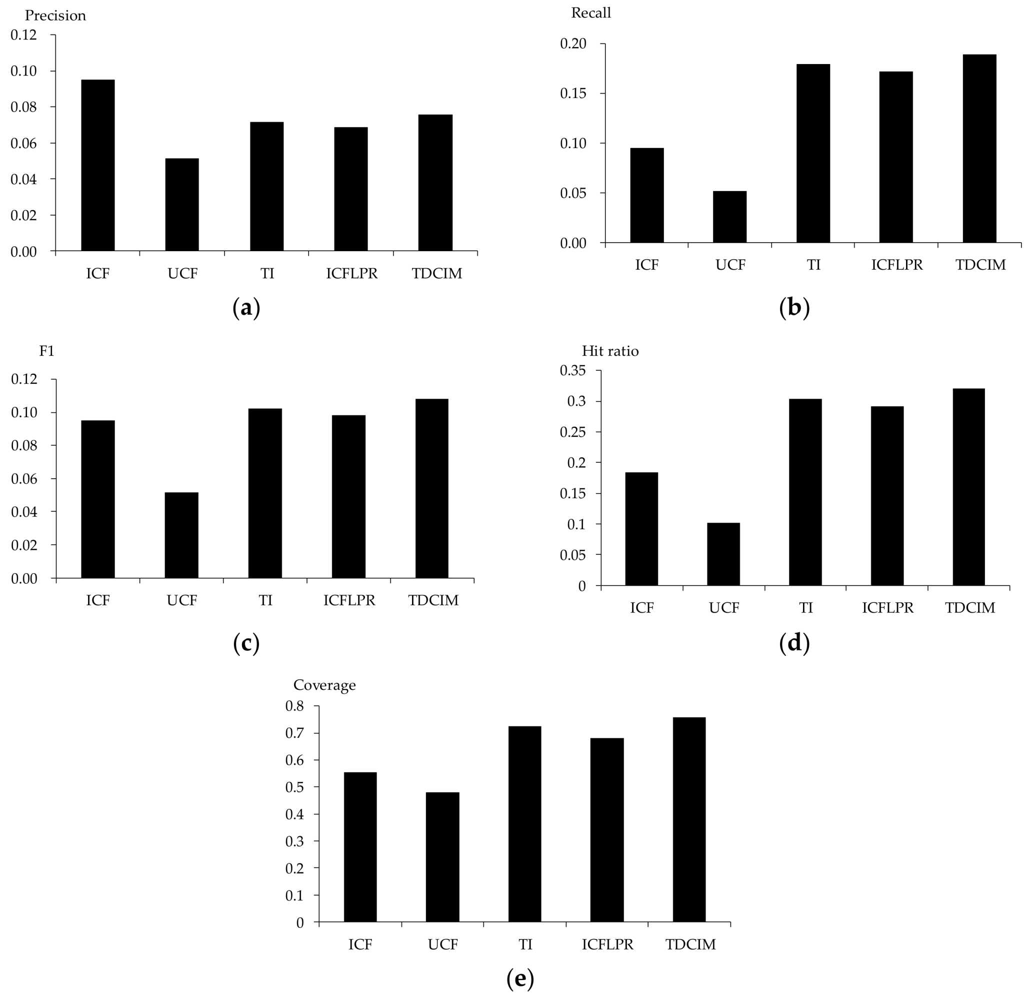


Figure 15. Performance comparison of several algorithms: (**a**) Precision; (**b**) Recall; (**c**) F1; (**d**) Hit ratio; (**e**) Coverage.

It is shown that the item-based collaborative filtering algorithm performs much better than the user-based collaborative filter algorithm in terms of all evaluation criteria, due to the fact that there are many more users than there are novels. That is to say, the item-based algorithm is more applicative than the user-based algorithm in online novel recommendation. Therefore, the item-based algorithm has been considered to extend the tag-driven algorithm. Novels that have been rated by an active user will be selected according to similarities between novels. Therefore, the item-based algorithm performs better than other algorithms with regards to precision. It also can be shown that precision performs worst in all algorithms compared to other evaluation criteria. This situation is common in recommendation systems and will be studied further.

The number of finished chapters is an important factor for preference degree rating. Therefore, the performance is improved significantly with a latent preference rating and a punishment mechanism from a performance comparison of ICFLPR with ICF. Meanwhile, novel tags are representative of one user's interest, which remains stable in the short term; therefore, the performance is improved significantly by tag-item algorithm. The recommendation list, achieved by ICFLPR, is re-ranked. Rankings of uninterested categories are downgraded and those of interested categories are upgraded for users. Consequently, the tag-driven algorithm with collaborative item modeling performs best in recall, F1, hit rate and coverage compared to other algorithms.

5.5. *Time Complexity*

For time complexity analysis, tag-driven recommendation with collaborative item modeling (TDCIM) and improved item-based CF with latent preference rating (ICFLPR) are compared against item-based CF(ICF), user-based CF (UCF) and tag-item(TI) algorithms. The experiments are conducted in a Spark distributed system on 3 servers, each of which works with 2 Intel® Xeon® processors E5-2620 and 128 G main memory.

Given n users, m novels and r novel tags, where $r << m << n$, the time complexity of TI, ICFLPR, TDCIM, ICF and UCF is $O(rm^2)$, $O(nmlogm)$, $O(nmlogm)$, $O(nm^2)$ and $O(mn^2)$ respectively. Elapsed time of every algorithm for different numbers of users is shown in Figure 16. There are many more users than there are novels, therefore, the runtime of item-based CF outperforms that of user-based CF. TI performs excellently because of fewer novel tags and novels. The idea of the MinHash scheme is applied as a useful estimator for the Jaccard similarity between novels and reduces calculation time in ICFLPR and TDCIM. When there are more than 180,000 users, the runtime of TI, ICFLPR and TDCIM is slightly different. However, UCF and ICF show a sharp rise in the runtime when there are more than 240,000 users. It means that our algorithm is workable and very effective for recommending online novels.

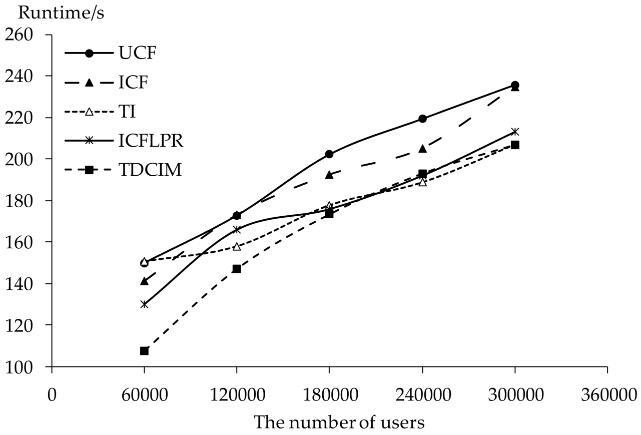


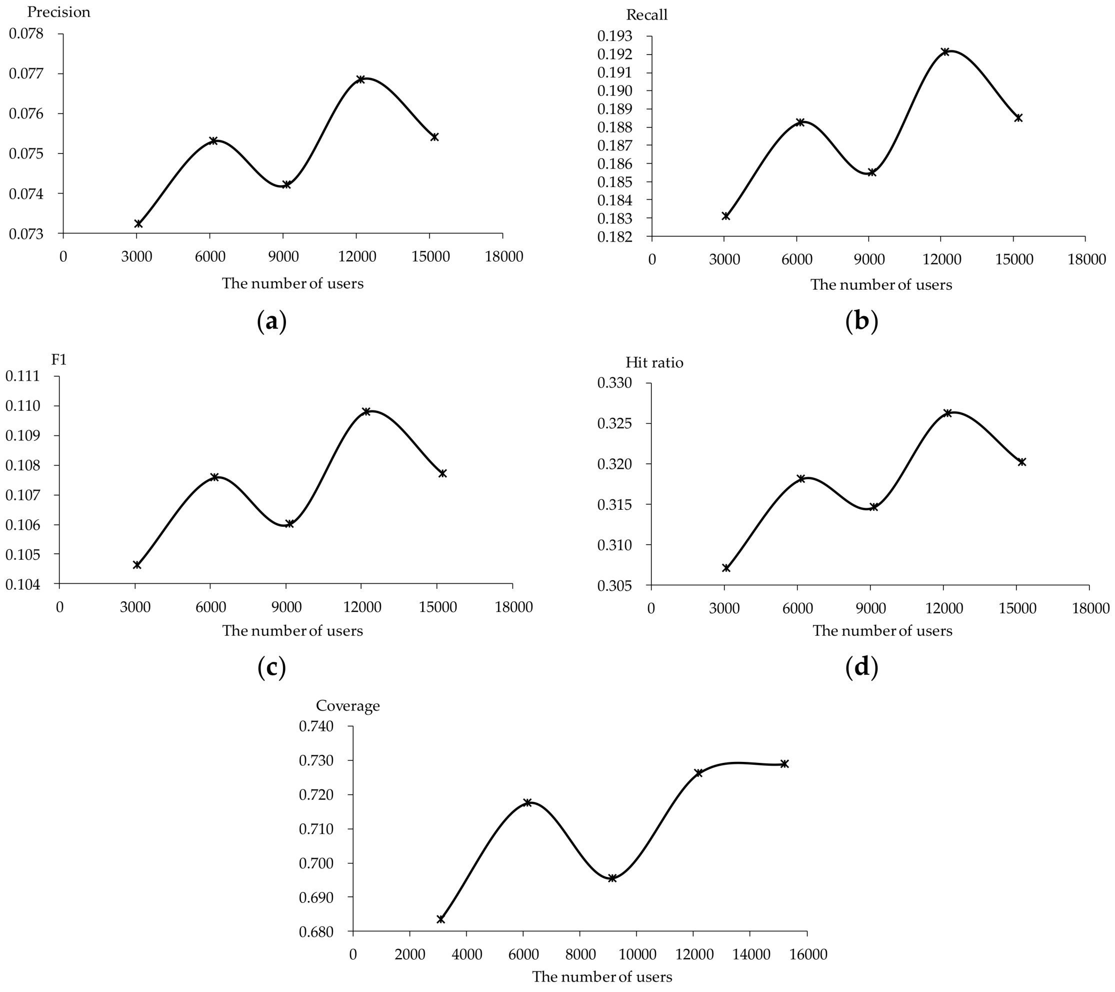
Figure 16. Runtime trendline of every algorithm against the number of users.

5.6. *Scalability*

The similarity between novels is calculated by the MinHash scheme in Section 3, which is a technique for using hash functions to map large datasets down to smaller hash values. When two data have a small distance between each other, their hash values are likely to be the same. The hashing process of different data can be conducted in a distributed environment to help compute similarities. After hashing process, the data for similarity computation is reduced greatly. Therefore, our experiments are conducted in a Spark distributed system on 3 servers in Section 5.5, ameliorating the scalability issue of collaborative filtering recommendation. The elapsed time of different algorithms as the number of users increases is shown in Figure 16. It can be seen that the proposed algorithm

TDCIM takes only about 3 min even for 320,000 users and speeds up more smoothly than other algorithms, implying that our algorithm is scalable.

The scalability issue is ameliorated but the key point lies in whether quality is guaranteed in the meantime. Therefore, the performance of TDCIM for different numbers of users (3094, 6162, 9148, 12,190, 15,236) and with different numbers of servers (1, 2, 3) are analyzed using the data in Section 5.1. About 130 threads are allocated for each server automatically. The results are shown in Figures 17 and 18 respectively. It can be seen that the performance is slightly sensitive to the number of users in Figure 17 and the fluctuation is about 10^{-3}. What's more, the performance does not become worse with more users and the difference is caused by the data randomly. The performance remains virtually unchanged with different numbers of servers for different numbers of users in terms of all evaluation criteria shown in Figure 18. The fluctuation is about 10^{-5} and the exclusive maximal different is about 10^{-3} for 6162 users in terms of coverage (Figure 18e). This means that recommendation quality is guaranteed with the MinHash scheme and the distributed system. In other words, TDCIM contributes to the performance of online novel recommendation not only with regard to the recommendation quality but also to scalability.



(**a**)

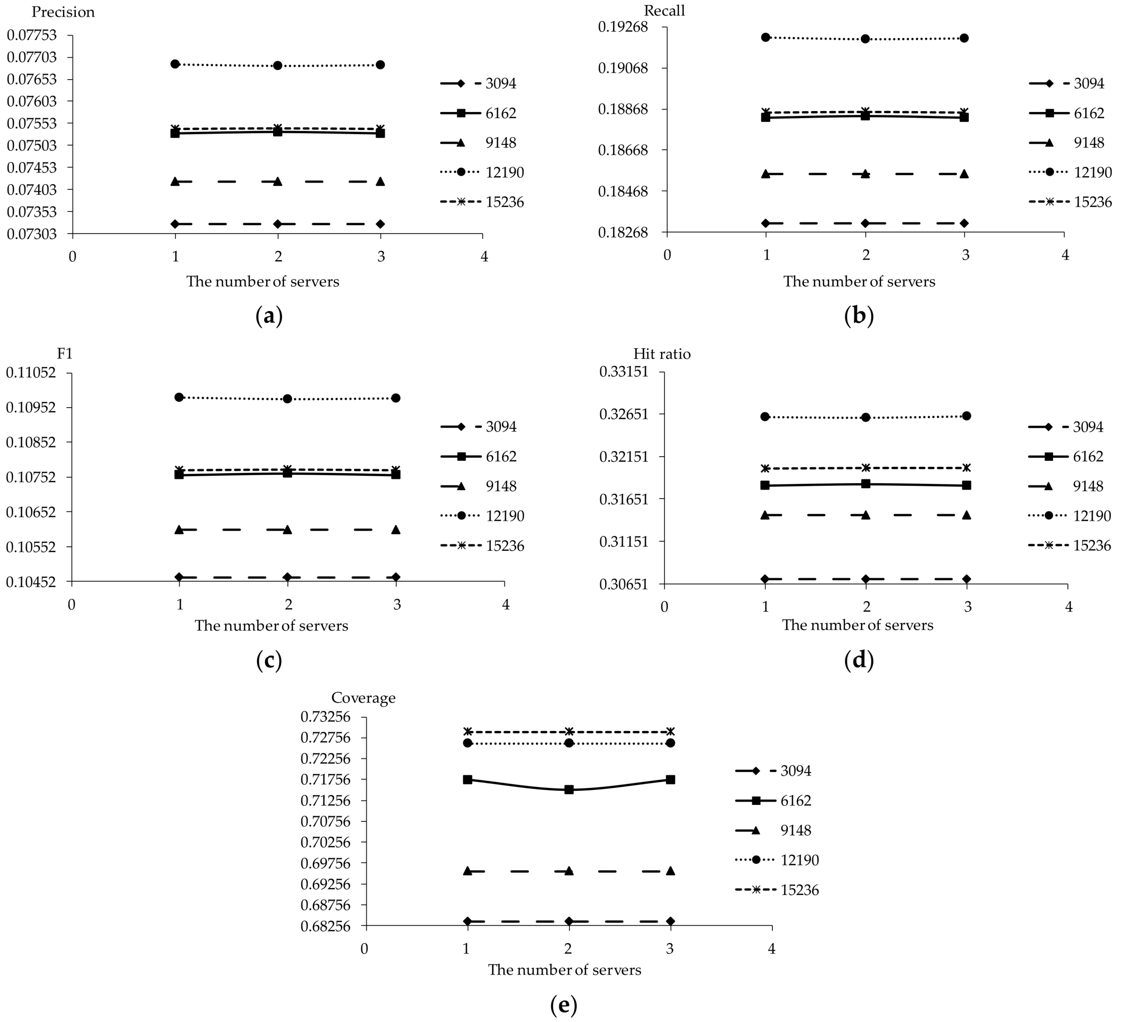
(**b**)

(**c**)

(**d**)

(**e**)

Figure 17. Performance trendline as the number of users increases: (**a**) Precision; (**b**) Recall; (**c**) F1; (**d**) Hit ratio; (**e**) Coverage.



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 18. Performance trendline as the number of servers increases: (**a**) Precision; (**b**) Recall; (**c**) F1; (**d**) Hit ratio; (**e**) Coverage.

6. Conclusions and Future Works

The interests of the majority of users who read online novels remain stable over a certain period. However, there are broad categories in the initial recommendation list obtained by collaborative filtering. That is to say, there are many inappropriately recommended novels. Meanwhile, most algorithms make an assumption that users can provide an explicit preference, however, the online novel reading preference of one reader is implicit in most cases. To solve these issues, a tag-driven algorithm with collaborative item modeling is proposed for online novel recommendation. The algorithm has the following contributions:

1. Data sparsity and recommendation personalization are considered in an item-based approach, which is more effective than a user-based approach for online novel recommendation.
2. Inaccurate preference rating and the lack of preference degree are solved by latent preference rating according to the difference between online novel reading and traditional book marketing.
3. A punishment mechanism based on novel popularity is considered, ameliorating the Matthew effect and the long tail effect for online novel recommendation.
4. A user's interest remains stable over the short term and can be reflected adequately by novel tags. Consequently, performance is greatly improved by the tag-driven algorithm with collaborative item modeling.

5. The proposed algorithm contributes to the performance of online novel recommendation not only with regards to recommendation quality but also to scalability.

The proposed algorithm is based on collective intelligence, which still causes the cold-start issue. The reading behaviors of a new novel are few, or even none. In this case, similarities between this novel and other novels are extremely low. Accordingly, the probability of this novel being recommended is small. Therefore, it is difficult to recommend new novels. In addition, recommendation precision is extremely poor. These problems need to be studied further.

Acknowledgments: This work is being supported by the Science and Technology Program of Guangzhou (No. 201806020075), the Fundamental Research Funds for the Central Universities (No. 2017BQ024) and the Natural Science Foundation of Guangdong Province (No. 2017A030310428).

Author Contributions: Zhaosheng Lin and Zhenyu Wang conceived problem definition and theoretical analysis; Fenghuan Li and Zhaosheng Lin designed and performed the experiments; Fenghuan Li analyzed the data and wrote the paper. All authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tan, S.L.; Bu, J.J.; Chen, C.; Xu, B.; Wang, C.; He, X.F. Using rich social media information for music recommendation via hypergraph model. *ACM Trans. Multimed. Comput. Commun. Appl.* **2011**. [CrossRef]
2. Carrer, N.W.; Hernández, A.M.L.; Valencia, G.R.; Francisco, G.S. Social knowledge-based recommender system. Application to the movies domain. *Expert Syst. Appl.* **2012**, *39*, 10990–11000. [CrossRef]
3. Barragáns, M.A.B.; Costa, M.E.; Burguillo, J.C.; Marta, R.L.; Fernando, A.M.F.; Ana, P. A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. *Inf. Sci.* **2010**, *180*, 4290–4311. [CrossRef]
4. Crespo, R.G.; Martínez, O.S.; Lovelle, J.M.C.; Bustelo, B.C.P.G.; Gayo, J.E.L.; Patricia, O.D.P. Recommendation system based on user interaction data applied to intelligent electronic books. *Comput. Hum. Behav.* **2011**, *27*, 1445–1449. [CrossRef]
5. Vaz, P.C.; David, M.D.M.; Martins, B.; Calado, P. Improving a hybrid literary book recommendation system through author ranking. In Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries, Washington, DC, USA, 10–14 June 2012; pp. 387–388.
6. Tewari, A.S.; Kumar, A.; Barman, A.G. Book recommendation system based on combine features of content based filtering, collaborative filtering and association rule mining. In Proceedings of the IEEE International Advance Computing Conference, Gurgaon, India, 21–22 February 2014; pp. 500–503.
7. Núñez-Valdéz, E.R.; Lovelle, J.M.C.; Martínez, O.S.; Vicente, G.D.; Patricia, O.D.P.; Carlos, E.M.M. Implicit feedback techniques on recommender systems applied to electronic books. *Comput. Hum. Behav.* **2012**, *28*, 1186–1193. [CrossRef]
8. Choi, K.; Yoo, D.; Kim, G.; Suh, Y. A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis. *Electron. Commer. Res. Appl.* **2012**, *11*, 309–317. [CrossRef]
9. Unger, M.; Bar, A.; Shapira, B.; Rokach, L. Towards latent context-aware recommendation systems. *Knowl. Based Syst.* **2016**, *104*, 165–178. [CrossRef]
10. Birtolo, C.; Ronca, D. Advances in clustering collaborative filtering by means of fuzzy c-means and trust. *Expert Syst. Appl.* **2013**, *40*, 6997–7009. [CrossRef]
11. Park, Y.K.; Park, S.C.; Lee, S.G.; Jung, W. Fast collaborative filtering with a k-nearest neighbor graph. In Proceedings of the International Conference on Big Data and Smart Computing, Bangkok, Thailand, 15–17 January 2014; pp. 92–95.
12. Ricci, F.; Rokach, L.; Shapira, B.; Kantor, P.B. *Recommender Systems Handbook*; Springer: Boston, MA, USA, 2011; pp. 73–105.
13. Park, D.H.; Kim, H.K.; Choi, I.Y.; Kim, J.K. A literature review and classification of recommender systems research. *Expert Syst. Appl.* **2012**, *39*, 10059–10072. [CrossRef]
14. Mandl, M.; Felfernig, A.; Teppan, E.; Schubert, M. Consumer decision making in knowledge-based recommendation. *J. Intell. Inf. Syst.* **2011**, *37*, 1–22. [CrossRef]

15. Cakir, O.; Aras, M.E. A recommendation engine by using association rules. *Procedia Soc. Behav. Sci.* **2012**, *62*, 452–456. [CrossRef]
16. Albanese, M.; Acierno, A.D.; Moscato, V.; Persia, F.; Picariello, A. A multimedia recommender system. *ACM Trans. Internet Technol.* **2013**, *13*. [CrossRef]
17. Reena, P.; Shalmali, A.P. Study of collaborative filtering recommendation algorithm-Scalability issue. *Int. J. Comput. Appl.* **2013**, *67*, 10–15.
18. Fang, Y.N.; Guo, Y.F.; Hu, H.C.; Lan, J.L. Improved collaborative filtering recommender algorithm based on sigmoid function. *Appl. Res. Comput.* **2013**, *30*, 1688–1691.
19. Salehi, M.; Kmalabadi, I.N. A hybrid attribute-based recommender system for e-learning material recommendation. *IERI Procedia* **2012**, *2*, 565–570. [CrossRef]
20. Ullah, F.; Sarwar, G.; Lee, S.C.; Yun, K.P.; Moon, K.D.; Kim, J.T. Hybrid recommender system with temporal information. In Proceedings of the International Conference on Information Networking, Bali, Indonesia, 1–3 February 2012; pp. 421–425.
21. Ghazanfar, M.A.; Prugel, B.A. Building switching hybrid recommender system using machine learning classifiers and collaborative filtering. *IAENG Int. J. Comput. Sci.* **2010**, *37*, 272–287.
22. Amato, F.; Moscato, V.; Picariello, A.; Piccialli, F. SOS: A multimedia recommender system for online social networks. *Futur. Gener. Comput. Syst.* **2017**. [CrossRef]
23. Yu, Y.; Yu, H.T.; Huang, R.Y. Collaborative filtering recommendation algorithm based on entropy optimization nearest-neighbor selection. *Appl. Res. Comput.* **2017**, *34*, 2618–2623.
24. Zhang, J.; Peng, Q.K.; Sun, S.Q.; Liu, C. Collaborative filtering recommendation algorithm based on user preference derived from item domain features. *Physica A* **2014**, *396*, 66–76. [CrossRef]
25. San, Y.H.; Chin, P.W.; Yi, F.L. Coauthorship networks and academic literature recommendation. *Electron. Commer. Res. Appl.* **2010**, *9*, 323–334.
26. Colace, F.; Santo, M.D.; Greco, L.; Moscato, V.; Picariello, A. A collaborative user-centered framework for recommending items in online social networks. *Comput. Hum. Behav.* **2015**, *51*, 694–704. [CrossRef]
27. Jadhav, S.D.; Channe, H.P. Efficient recommendation system using decision tree classifier and collaborative filtering. *Int. Res. J. Eng. Technol.* **2016**, *3*, 2113–2118.
28. Kim, H.N.; Alkhaldi, A.; Saddik, A.E.; Jo, G.S. Collaborative user modeling with user-generated tags for social recommender systems. *Expert Syst. Appl.* **2011**, *38*, 8488–8496. [CrossRef]
29. Ma, H.F.; Jia, M.H.Z.; Zhang, D.; Lin, X.H. Combining tag correlation and user social relation for microblog recommendation. *Inf. Sci.* **2017**, *385–386*, 325–337. [CrossRef]
30. Ghazarian, S.; Nematbakhsh, M.A. Enhancing memory-based collaborative filtering for group recommender systems. *Expert Syst. Appl.* **2015**, *42*, 3801–3812. [CrossRef]
31. Park, Y.K.; Park, S.C.; Jung, W.S.; Lee, S.G. Reversed CF: A fast collaborative filtering algorithm using k-nearest neighbor graph. *Expert Syst. Appl.* **2015**, *42*, 4022–4028. [CrossRef]
32. Mou, B.H.; Zhang, Z.H.; Zhang, L.; Min, F. Comparison study of collaborative filtering algorithms based on quadripartite graph. *J. Front. Comput. Sci. Technol.* **2017**, *11*, 875–886.