BIG DATA MINING AND ANALYTICS
ISSN 2096-0695 03/06 pp128–136
Volume 1, Number 2, June 2018
DOI: 10.26599/BDMA.2018.9020012

An Improved Hybrid Collaborative Filtering Algorithm Based on Tags and Time Factor

Chunxia Zhang, Ming Yang*, Jing Lv, and Wanqi Yang

Abstract: The Collaborative Filtering (CF) recommendation algorithm, one of the most popular algorithms in Recommendation Systems (RS), mainly includes memory-based and model-based methods. When performing rating prediction using a memory-based method, the approach used to measure the similarity between users or items can significantly influence the recommendation performance. Traditional CFs suffer from data sparsity when making recommendations based on a rating matrix, and cannot effectively capture changes in user interest. In this paper, we propose an improved hybrid collaborative filtering algorithm based on tags and a time factor (TT-HybridCF), which fully utilizes tag information that characterizes users and items. This algorithm utilizes both tag and rating information to calculate the similarity between users or items. In addition, we introduce a time weighting factor to measure user interest, which changes over time. Our experimental results show that our method alleviates the sparsity problem and demonstrates promising prediction accuracy.

Key words: recommendation system; similarity; tag; time factor

1 Introduction

With the rapid development of information technology and networks, the volume of data is increasing and network information transmission is rapid, which brings convenience but also information overload to daily life. In this era of information explosion, the development of methods for extracting information from massive data in a timely manner is becoming the focus of increasing attention.

Recommendation Systems (RS)[1,2], a hot research topic, use a kind of information filtering technology that mines historical user behavior to excavate information and identify the individual needs of users in today's environment of inforation overload. Thus far, researchers have proposed a variety of recommendation techniques that have been successfully applied to various fields, e.g., recommendations regarding Amazon shopping[3], music[4], and the news[5].

Existing recommendation algorithms mainly include content-based recommendations[6,7], collaborative filtering[8,9], and knowledge-based recommendations[10,11]. Of these, collaborative filtering has attracted the attention of researchers in both academia and industry due to its high accuracy and wide range of recommendations. Collaborative filtering utilizes memory-based methods[12–14], model-based methods[15–17], and hybrid approaches. Memory-based methods first calculate the similarity between users/items, obtain the top k-neighbor users/items that are most similar to the target user/item, then generate prediction results based on these neighbors. In contrast, model-based methods learn a model from training data using machine learning or other techniques based on a user-item rating matrix, and then make predictions. The most commonly used model-based methods[18] utilize linear regression models, Bayesian models, and graph models.

• Chunxia Zhang, Ming Yang, Jing Lv, and Wanqi Yang are with the School of Computer Science and Technology, Nanjing Normal University, Nanjing 210023, China. E-mail: 1608195294@qq.com; m.Yang@njnu.edu.cn; jinglv_njnu@163.com; yangwq@njnu.edu.cn.

∗ To whom correspondence should be addressed.
Manuscript received: 2017-12-27; accepted: 2018-01-03

Rating prediction is an important issue in RS. When making predictions based on memory-based methods, the key step is measuring the similarity between users (or items), which directly affects the performance of the recommendation algorithm. The traditional collaborative filtering method makes recommendations based on a rating matrix. However, with the continuing expansion of network scales and the explosive growth in the numbers of items and users, traditional collaborative filtering algorithms inevitably suffer from problems with data sparsity and cold starts due to the extreme sparsity of the rating matrix. Researchers have made many efforts to solve this problem. The authors in Ref. [19] used clustering and association rules to solve the sparsity problem, whereas those in Ref. [20] introduced tag information preset by experts to the process of calculating similarity to compensate for the deficiencies of sparse rating matrixes. The authors in Ref. [21] obtained the optimal neighbor set by subspace clustering to alleviate data sparsity. To complicate the problem, user interest evolves over time and users do not maintain their inerest in items they previously liked. Traditional collaborative filtering methods cannot capture changing user interest. In this paper, we propose an improved collaborative filtering algorithm (TT-Hybrid CF) that utilizes tag and rating information to calculate the similarity between users (or items). In addition, it adds a hot-item penalty into the process of calculating users' similarity to offset the influence of the hot item in their co-rated items. In addition, the TT-Hybrid CF also takes the time factor into account when measuring user interest, since it changes over time. Our contributions in this paper are sumarized as follows:

- We use tag information in the calculation of items' (or users) similarity, and introduce a hot-item penalty into the user-similarity calculation process.
- We take into account the time factor when measuring user interest, which changes with time.
- We propose a new hybrid collaborative filtering model (TT-HybridCF) that combines improved user-based collaborative filtering (TT-UserCF) and item-based collaborative filtering (TT-ItemCF).
- Our extensive experimental results show that our proposed method is effective.

The rest of this paper is arranged as follows. In Section 2, we introduce related work. In Section 3, we present our method. We report our experimental results and analysis in Section 4, and we draw our conclusion in Section 5.

2 Related Work

The recommendation algorithm is the core component of a recommendation system. With respect to the different recommendation approaches, recommendation algorithms can be divided into three main categories: content-based, collaborative filtering, and knowledge-based recommendation algorithms, of which collaborative filtering is the most popular. The collaborative filtering algorithm, first proposed by Goldberg et al.[22] in 1992, has made significant progress in the past two decades. This algorithm constructs a user's interest model by analyzing a user's historical behavior and then generates recommendations for target users. Collaborative-filtering recommendation algorithms comprise a large category containing many different types of algorithms, which can generally be divided into memory-based and model-based methods. Memory-based methods first calculate the similarity between users (or items), obtain the top k-neighbor users (items) that are most similar to the target users (items), then make predictions based on these neighbors. Memory-based methods include User-based Collaborative Filtering (UserCF)[23,24] and Item-based Collaborative Filtering (ItemCF)[25,26]. Model-based methods first construct a prediction model based on a user-item rating matrix and then make predictions. In this paper, we mainly focus on the memory-based collaborative filtering algorithm. Table 1 lists the main mathematical symbols we use in this article.

2.1 User-based collaborative filtering

Resnick et al.[27] first proposed UserCF and verified its validity with the MovieLens dataset in 1994. The UserCF algorithm is based on the assumption that users are likely to purchase the same items if they share similar interests, and it identifies users who share

Table 1 Definitions of mathematical symbols.

Symbol	Definition
U	The set of all users. $\forall u, v \in U$
I	The set of all items. $\forall i, j \in I$
I_{uv}	The set of co-rated items of user u and user v.
I_u, I_v	The set of items which are rated by user u, user v.
U_{ij}	The set of co-rated users on item i and item j.
R_{uj}, R_{vj}	The rating of user u on item j, the rating of user v on item, respectively.
$\bar{R}_u, \bar{R}_v$	The average rating of user u and user v respectively.
$N(u)$, $N(i)$	The neighbor set of user u and item i respectively.

similar interests with a target user as neighbor users. Therefore, to make a recommendation to a user, UserCF first identifies neighbor users by mining the user's historical behavior. Then, it makes recommendations to the target user based on the behavior of these neighbor users. The main steps in UserCF are as follows:

Step one: Calculate the similarity between users.

In the UserCF, the measurement of user similarity is a critical step that directly impacts the accuracy of recommendations. There are several commonly used similarity measures in RS, including the Pearson Correlation Coefficient (PCC), Euclidean distance, cosine similarity, and modified cosine similarity methods, the specific formulas of which are listed in Table 2. Of these, the PCC is the most frequently used and most accurate similarity measure strategy. PCC calculates the similarity between users based on an item set that is co-rated by users. The PCC formula is shown in Eq. (1) below.

$$sim(u,v) = \frac{\sum_{j\in I_{uv}} (R_{uj} - \bar{R}_u)(R_{vj} - \bar{R}_j)}{\sqrt{\sum_{j\in I_{uv}} (R_{uj} - \bar{R}_u)^2}\sqrt{\sum_{j\in I_{uv}} (R_{vj} - \bar{R}_v)^2}} \tag{1}$$

We use the average score for users u and v to eliminate any difference in different users' scoring scales and to ensure similarity accuracy. For example, some users tend to give high ratings for items, whereas other users are more demanding and give lower scores, despite the fact that these users may have the same interest.

Step two: Find the target user's neighborhood set.

After Step one, we obtain the user similarity matrix and sort this matrix according to the degree of similarity. Then, we can determine the top k-neighbor users who are most similar to the target user, which we label as $N(u)$.

Table 2 Traditional methods for measuring similarities.

Cosine similarity	$sim(u,v) = \frac{\sum_{j=1}^{n} R_{uj} R_{vj}}{\sqrt{\sum_{j=1}^{n} R_{uj}^2}\sqrt{\sum_{j=1}^{n} R_{vj}^2}}$
Modified cosine similarity	$sim(u,v) = \frac{\sum_{j\in I_{uv}} (R_{uj} - \bar{R}_u)(R_{vj} - \bar{R}_v)}{\sqrt{\sum_{j\in I_u} (R_{uj} - \bar{R}_u)^2}\sqrt{\sum_{j\in I_v} (R_{vj} - \bar{R}_v)^2}}$
Pearson correlation coefficient	$sim(u,v) = \frac{\sum_{j\in I_{uv}} (R_{uj} - \bar{R}_u)(R_{vj} - \bar{R}_v)}{\sqrt{\sum_{j\in I_{uv}} (R_{uj} - \bar{R}_u)^2}\sqrt{\sum_{j\in I_{uv}} (R_{vj} - \bar{R}_v)^2}}$

Step three: Predict the unknown rating of the target user u for item i.

According to the known score of a neighbor set, we can predict the unknown rating of target user u for item i. Using UserCF, we can predict the rating of user u for item i, which can be expressed as shown in Eq. (2).

$$P_{UserCF}(R_{ui}) = \bar{R}_u + \frac{\sum_{v\in N(u)} sim(u,v)(R_{vi} - \bar{R}_v)}{\sum_{v\in N(u)} sim(u,v)} \tag{2}$$

The above equation indicates that we can determine the unknown rating of the target user u for item i based on the weighted sum of the known rating of the user's neighbor set for item i. We calculate $sim(u, v)$ using Eq. (1) .

2.2 Item-based collaborative filtering

Linden et al.[3] first proposed ItemCF, which assumes that a user may like items that are similar to items they have previously liked. The main steps of ItemCF are as follows:

Step one: Calculate the similarity between items.

The calculation of the similarity between items i and j is based on a user set in which users rate items i and j together. We again use PCC to calculate the similarity of items, as shown in Eq. (3).

$$sim(i,j) = \frac{\sum_{u\in U_{ij}} (R_{ui} - \bar{R}_i)(R_{vj} - \bar{R}_j)}{\sqrt{\sum_{u\in U_{ij}} (R_{ui} - \bar{R}_i)^2}\sqrt{\sum_{u\in U_{ij}} (R_{vj} - \bar{R}_j)^2}} \tag{3}$$

Step two: Find the target item's neighborhood set.

We then sort the obtained similarity matrix based on the degree of similarity. Then, we can obtain the top k-neighbor items that are most similar to the target item, which we label as $N(i)$.

Step three: Predict the unknown rating of the target item.

Equation (4) shows the use of ItemCF to predict the rating of user u for item i.

$$P_{ItemCF}(R_{ui}) = \bar{R}_i + \frac{\sum_{j\in N(i)} sim(i,j)(R_{uj} - \bar{R}_j)}{\sum_{j\in N(i)} sim(i,j)} \tag{4}$$

This equation indicates that the unknown rating of user u for target item i can be represented by the known rating of the target item i's neighbor set. $sim(i, j)$ represents the similarity of items i and j, which is calculated using Eq. (3).

From the above, we can see that UserCF recommends hotspots in a group in which members have the same interest as those of a target user, i.e., UserCF emphasizes socialization. In contrast, ItemCF emphasizes individualization, in that it recommends similar items based on a user's historical behavior. However, both methods suffer from problems of sparsity and cold start.

2.3 Hybrid CF

When using just one recommended algorithm, the resulting recommendation accuracy is not very high, since UserCF and ItemCF both have advantages and disadvantages. Therefore, to make up for the shortcomings of individual algorithms, some scholars have proposed the integration of different recommendation algorithms when making recommendations. The authors of Ref. [24] integrated UserCF and ItemCF in making recommendations, as shown in Eq. (5) below.

$$R_{ui} = \gamma \times \left(\bar{R}_u + \frac{\sum\limits_{v \in N(u)} sim(u,v)(R_{vi} - \bar{R}_v)}{\sum\limits_{v \in N(u)} sim(u,v)} \right) + (1-\gamma) \times \left(\bar{R}_i + \frac{\sum\limits_{j \in N(i)} sim(i,j)(R_{uj} - \bar{R}_j)}{\sum\limits_{j \in N(i)} sim(i,j)} \right) \quad (5)$$

The first part of Eq. (5) comprises the prediction based on UserCF and the second part on ItemCF. The parameter γ is an adjustment parameter, which controls the degree to which this method relies on UserCF and ItemCF.

3 Our Proposed Model

In this paper, we propose an improved collaborative filtering method based on tags and a time factor (TT-Hybrid CF) for RS. In calculating similarity, the TT-Hybrid CF algorithm utilizes tag and rating information to calculate the similarity of users (or items). In addition, it employs a hot-item penalty when calculating users' similarity to penalize the influence of a hot item in their co-rated items. In the prediction phase, TT-Hybrid CF takes the time factor into account to measure user interest, which changes over time.

3.1 Calculating similarity with tags

Existing collaborative filtering algorithms only use a rating matrix to calculate similarity, but this rating matrix is very sparse, so they experience sparsity problems. The authors in Ref. [19] proposed an improved collaborative filtering algorithm based on the combination of tags and ratings, known as UTR-CF. In this paper, we use this UTR-CF method to calculate the tag similarity between users or items. In the MovieLens dataset, the movie tag information is the genre, e.g., action, adventure, animation, and the user tag information user comprises demographic characteristics, e.g., {Man, 28, 'educator'}. Before calculating the similarity of a tag set, we first transform the tag set and other text information into digital form to facilitate the modeling process. Assume that two users (or two movies) are converted to digital information and are represented as two vectors in m-dimensional space: $\boldsymbol{t}$= (t_1, t_2, ..., t_m), $\boldsymbol{s}$ = (s_1, s_2, ..., s_m). Then, we use the cosine similarity to calculate the similarity of the tag vectors, as follows:

$$sim(\boldsymbol{t},\boldsymbol{s}) = \frac{\sum_{k=1}^{m} t_k s_k}{(\sqrt{\sum_{k=1}^{m} t_k^2 \sum_{k=1}^{m} s_k^2})} \quad (6)$$

3.2 Hot item punishment

In UserCF, the traditional method for calculating user similarity is to consider item ratings co-rated by two users without considering the influence of the hot items. For example, if two users buy a Xinhua Dictionary, this does not mean that they have the same interest because most Chinese people have bought this book. However, if both users buy a book with the title Machine Learning, we can consider that they have the same interest because only those who study this field of research would buy this book. In summary, if two users buy hot items, this does not indicate that they have the same interests. As such, we introduce a weight w_i to reduce the influence of hot items on user similarity, as shown in Eq. (7).

$$w_i = \frac{1}{\lg(1+N_i)} \quad (7)$$

In this equation, $i \in I_{uv}$, N_i represents the number of users who have rated the item i.

3.3 Temporal weight

Actually, user interest in items fluctuates, but traditional collaborate filtering algorithms do not take this into consideration. The recent behavior of users is more influential than their earlier behavior. If a user liked an item last month, this does not mean that he (or she) still likes that item this month. Recent behavior is more likely to indicate a user's current interest. So, we introduce the $f(t_{ui})$ to represent a time weight, which penalizes earlier behavior and highlights recent behavior, as shown in Eq. (8).

$$f(t_{ui}) = 1 - \exp(-t_{ui}) \tag{8}$$

where t_{ui} is the rating time of user u on item i. We use the exponential function to indicate user interest, which decays over time. The value of $f(t_{ui})$ increases with time. A larger t_{ui} value indicates that the time of the rating behavior is more recent, so the time weight is greater.

Considering the above, we propose three algorithms: the TT-UserCF, TT-ItemCF, and TT-Hybrid CF algorithms.

(1) TT-UserCF

In contrast to the traditional UserCF algorithm, TT-UserCF adds the hot-item penalty weight w_i introduced above to the calculation of the rating similarity between users, which reduces the influence of hot items on user similarity. This improved method for calculating the rating similarity of users is shown in Eq. (9) below.

$$sim^*_{rating}(u,v) = \frac{\sum_{j \in I_{uv}} (R_{uj} - \bar{R}_u)(R_{vj} - \bar{R}_j) w_i}{\sqrt{\sum_{j \in I_{uv}} (R_{uj} - \bar{R}_u)^2 w_i} \sqrt{\sum_{j \in I_{uv}} (R_{vj} - \bar{R}_v)^2 w_i}} \tag{9}$$

We indicate the similarity calculated using Eq. (6) as $sim_{tag}(u, v)$ and, using our proposed method, we utilize tag and rating information to compute the similarity between users or items. This integrated similarity is calculated as shown as Eq. (10).

$$sim^*_{unify}(u,v) = \beta \times sim^*_{rating}(u,v) + (1-\beta) \times sim_{tag}(u,v) \tag{10}$$

$sim^*_{unify}(u, v)$ is the integrated similarity between user u and user v. We calculate $sim^*_{tag}(u, v)$ using Eq. (9) to obtain the rating similarity between users u and v. The parameter β is an adjustment parameter, which controls the proportions of rating and tag information when calculating similarity.

The calculation process of the TT-UserCF algorithm in predicting the rating of user u for item i is as follows:

$$R_{ui} = \bar{R}_u + \frac{\sum_{v \in N(u)} sim^*_{unify}(u,v)(R_{vi} - \bar{R}_v) f(t_{vi})}{\sum_{v \in N(u)} sim^*_{unify}(u,v) f(t_{vi})} \tag{11}$$

$f(t_{vi})$ penalizes the influence of the past interest of neighbor user v and can thus make more accurate recommendations.

(2) TT-ItemCF

In contrast to traditional ItemCF, TT-ItemCF utilizes tag and rating information to calculate the similarity between items. We refer to the rating similarity calculated using Eq. (3) as $sim_{rating}(i, j)$ and the tag similarity calculated using Eq. (6) as $sim_{tag}(i,j)$. Thus, the integrated similarity is as follows:

$$sim^*_{unify}(i,j) = \alpha \times sim^*_{rating}(i,j) + (1-\alpha) \times sim_{tag}(i,j) \tag{12}$$

In Eq. (12) above, parameter α is an adjustment parameter, which controls the proportion of rating and tag information in the process of calculating similarity.

The TT-ItemCF formula for predicting the rating of user u for item i is as follows:

$$R_{ui} = \bar{R}_i + \frac{\sum_{j \in N(i)} sim^*_{unify}(i,j)(R_{uj} - \bar{R}_j) f(t_{uj})}{\sum_{j \in N(i)} sim^*_{unify}(i,j) f(t_{uj})} \tag{13}$$

(3) TT-Hybrid CF

The TT-Hybrid CF algorithm employs the following formula to predict the rating of user u for item i.

$$R_{ui} = \lambda \times \left(\bar{R}_u + \frac{\sum_{v \in N(u)} sim^*_{unify}(u,v)(R_{vi} - \bar{R}_v) f(t_{vi})}{\sum_{v \in N(u)} sim^*_{unify}(u,v) f(t_{vi})} \right) + (1-\lambda) \times \left(\bar{R}_i + \frac{\sum_{j \in N(i)} sim^*_{unify}(i,j)(R_{uj} - \bar{R}_j) f(t_{uj})}{\sum_{j \in N(i)} sim^*_{unify}(i,j) f(t_{uj})} \right) \tag{14}$$

Equation (14) is used to predict the behavior of user u with respect to item i. The first part of this formula is the improved TT-UserCF and the second part is the improved TT-ItemCF. Parameter λ is an adjustment parameter, which controls the degree to which the method relies on TT-UserCF and TT-ItemCF.

4 Experimental

4.1 Dataset

In this study, we used the MovieLens1 dataset collected by the GroupLens research team at the Uniersity of Minnesota. This dataset includes 100 000 rating data from 943 users with respect to 1682 films. Each user was asked to rate at least 20 films, with a rating range from 1 to 5, whereby the higher the score, the more interested the user was in the movie. We chose this dataset because it conains the label information used in our method and because most researchers have conducted experiments based on this dataset.

4.2 Evaluation metric

The Mean Absolute Error (MAE), a commonly used evaluation metric in collaborative filtering RS, is used to evaluate the accuracy of a recommendation system. The MAE is the average absolute error between predicted and real ratings. The lower the MAE value, the higher is the accuracy of system's recommendation. Assume that the predicted ratings are $(P_{u1}, P_{u2}, P_{u3}, ..., P_{uN})$ and the real ratings are $(r_{u1}, r_{u2}, r_{u3}, ..., r_{uN})$, then the MAE is as follows:

$$\mathrm{MAE} = \frac{\sum\limits_{i \in N} |P_{ui} - r_{ui}|}{N} \tag{15}$$

4.3 Experimental results

To confirm the superiority of our approach, we compared the results of our methods with those of UserCF[27], ItemCF[3], HybridCF[24], and UTR-CF[20], and we present our detailed analysis in this section.

Experiment 1: Impact of Parameter β

In the TT-UserCF algorithm, parameter β in Eq. (10) plays an important role in calculating the similarity between users. It controls the proportions of tag and rating information in the similarity calculation. In this experiment, we set the size of the user's neighbor set to 20 and changed the value of β from 0 to 1 with steps of 0.1. Figure 1 shows our experimental results for the testing set, which shows the MAE values of TT-UserCF for different β values.

Figure 1 shows the impact on the MAE in the TT-UserCF algorithm, from which we can observe that the MAE value decreases and reaches a minimum, then increases as the value of β increases. Obviously, we can conclude that the value of β affects the accuracy of the recommendation. On this basis, we found the optimal value of parameter β to be 0.7, which means that the proportion of rating similarity is 0.3 and the proportion of tag similarity is 0.7.

Experiment 2: Impact of Parameter α

In the TT-ItemCF algorithm, we use Eq. (12) to calculate the similarity between items, in which factor α is very important. This parameter balances the influence of the tag and rating information. In this experiment, we set the size of the neighbor set to 20 and changed the value of α from 0 to 1 with steps of 0.1. Figure 2 shows our experimental testing results of the MAE values for the TT-ItemCF with different α values.

Figure 2 shows the impact of parameter α on the MAE results, in which we can see that as the value of α increases, the value of MAE decreases until it reaches a minimum, after which it increases. We determined the optimal value of parameter α to be 0.4, which means that the optimal proportion of the rating similarity is 0.4 and that of tag similarity is 0.6.

Experiment 3: Impact of Parameter λ

In the TT-HybridCF algorithm, λ is an important factor that balances the influence of the TT-UserCF and TT-ItemCF algorithms. In this experiment, we set the size of the neighbor set to 10 and changed the value of λ from 0 to 1 with steps of 0.1. Figure 3 shows our experimental testing results of the MAE values for the TT-HybridCF for different λ values.

Figure 3 shows the impact of λ on the MAE results, in which we can see that as the λ value increases, the MAE value decreases until it reaches a minimum, after which it increases. We determined the optimal value of parameter λ to be 0.4, which means that the optimal

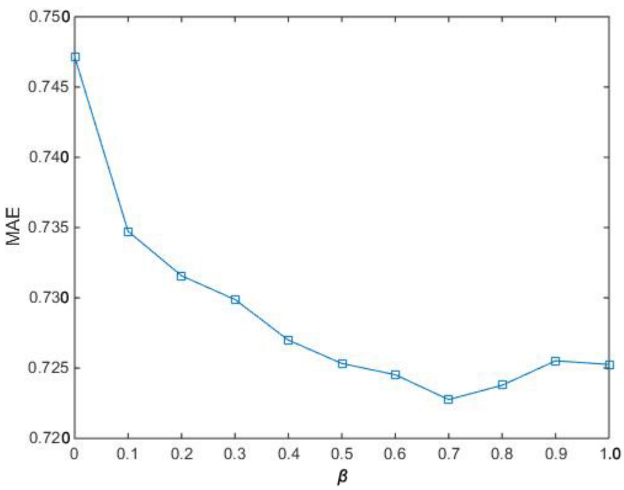

Fig. 1 Effect of parameter β on MAE of TT-UserCF algorithm.

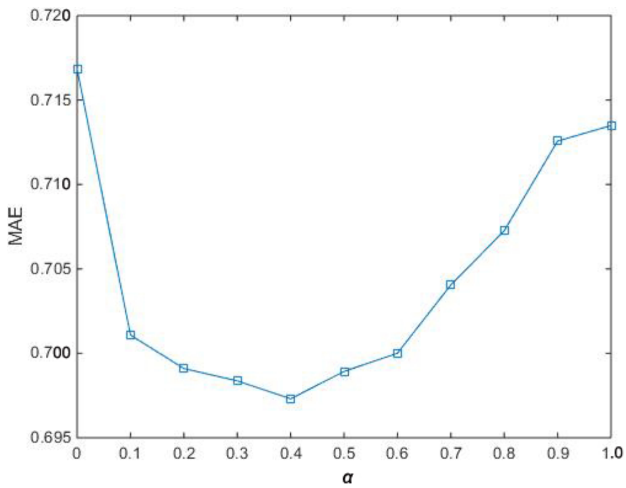

Fig. 2 Effect of parameter α on MAE values of TT-ItemCF algorithm.

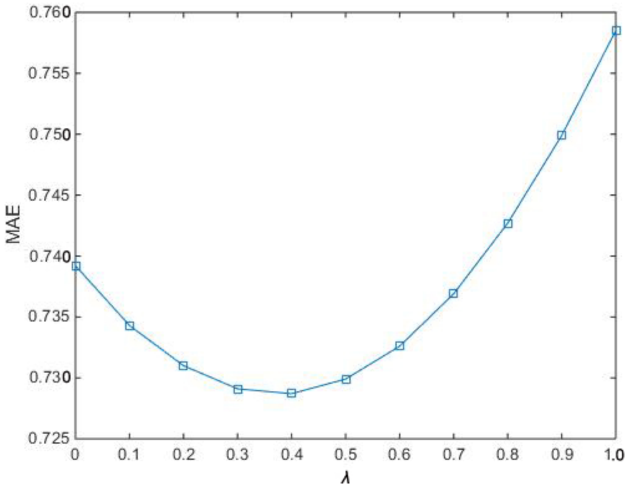

Fig. 3 Effect of parameter λ on MAE by the TT-HybridCF algorithm.

proportion of TT-UserCF is 0.4 and that of TT-ItemCF is 0.6.

Experiment 4: Comparison of TT-UserCF algorithm with two other contrastive algorithms

In this experiment, we compared the TT-UserCF algorithm with the UserCF and UTR-UserCF algorithms. We set the number of neighbor K to range from 10 to 50 with steps of 5. Table 3 shows the MAE values of all the algorithms.

Table 3 shows that as the value of K increases, the MAE value has a downward trend. Obviously, the number of neighbor users is very important. By taking more neighbors into considerations, the recommendation becomes more accurate. In addition, the performance of our method is better than those of the traditional UserCF and UTR-UserCF algorithms, which means that it is effective to take into consideration the tag information and changing users' interest.

Experiment 5: Comparison of TT-ItemCF with two other contrastive algorithms

In this experiment, we compared the TT-ItemCF algoithm with ItemCF and UTR-ItemCF algorithms. We set the number of neighbor K to range from 10 to 50 with steps of 5. Table 4 lists the MAE values of all the algorithms. The MAE value varies with the number of K. We can clearly see that our proposed TT-ItemCF approach outperforms the other approaches (ItemCF and UTR-ItemCF).

Experiment 6: Comparison of TT-Hybrid CF with two other contrastive algorithms

In this experiment, we compared the TT-Hybrid CF with the UserCF, ItemCF, HybridCF, and UTR-ItemCF algorithms. We set the numer of neighbor Ks to range from 10 to 50 with steps of 5. Table 5 shows the MAE values of all the algorithms. A downward trend in the MAE values as the size of the neighbor set increases, so we can clearly see that the ac uracy of our method TT-Hybrid CF outperforms the other three methods. That is, taking into consideration both tag information and the time factor is very important.

5 Conclusion

In this paper, we proposed an improved hybrid

Table 3 MAE values of UserCF, UTR-UserCF, and TT-UserCF algorithms

K	UserCF	UTR-UserCF	TT-UserCF
10	0.7736	0.7666	0.7496
15	0.7617	0.7565	0.7305
20	0.7555	0.7514	0.7227
25	0.7523	0.7488	0.7197
30	0.7505	0.7470	0.7168
35	0.7489	0.7459	0.7145
40	0.7482	0.7453	0.7131
45	0.7475	0.7449	0.7113
50	0.7471	0.7447	0.7119

Table 4 MAE values of the ItemCF, UTR-ItemCF, and TT-ItemCF algorithms

K	ItemCF	UTR-ItemCF	TT-ItemCF
10	0.7725	0.7469	0.7091
15	0.7595	0.7404	0.7013
20	0.7528	0.7372	0.6973
25	0.7489	0.7359	0.6972
30	0.7466	0.7354	0.6964
35	0.7450	0.7352	0.6962
40	0.7440	0.7351	0.6968
45	0.7436	0.7352	0.6968
50	0.7433	0.7354	0.6972

Table 5 MAE values of the UserCF, ItemCF, HybridCF, UTR-Hybrid, and TT-HybridCF algorithms

K	UserCF	ItemCF	HybridCF	UTR-Hybrid	TT-HybridCF
10	0.7740	0.7725	0.7517	0.7308	0.7151
15	0.7617	0.7595	0.7445	0.7281	0.7066
20	0.7555	0.7528	0.7404	0.7269	0.7027
25	0.7523	0.7489	0.7382	0.7267	0.7009
30	0.7505	0.7466	0.7370	0.7267	0.6991
35	0.7489	0.7450	0.7361	0.7269	0.6990
40	0.7482	0.7440	0.7355	0.7270	0.6982
45	0.7475	0.7436	0.7352	0.7272	0.6980
50	0.7471	0.7433	0.7350	0.7274	0.6980

collaborative filtering method based on tags and the time factor (TT-Hybrid CF). In the process of calculating similarity, we used both tag and rating information. In addition, the TT-Hybrid CF introduces a hot-item penlty to the calculation of users'similarity to penalize the influence of a hot item among co-rated items. In the process of rating prediction, TT-Hybrid CF takes into consideration the users'interest by introducing a temporal weight to measure the changing user interest over time. Compared with four other collaborative filtering algorithms (UserCF, HybridCF, ItemCF, and UTR-CF), our proposed TT-Hybrid CF realizes a great improvement in recommendation performance. In future work, we will continue to research the problems of sparsity and cold start in traditional collaboraive filtering.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 61432008 and 61272222).

References

[1] G. Adomavicius and A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, 2005.

[2] L. Y. Lü, M. Medo, C. H. Yeung, Y. C. Zhang, Z. K. Zhang, and T. Zhou, Recommender systems, *Phys. Rep.*, vol. 519, no. 1, pp. 1–49, 2012.

[3] G. Linden, B. Smith, and J. York, Amazon.Com recommendations: Item-to-Item collaborative filtering, *IEEE Int. Comput.*, vol. 7, no. 1, pp. 76–80, 2003.

[4] O. Celma and X. Serra, FOAFing the music: Bridging the semantic gap in music recommendation, *Web Semant.*, vol. 6, no. 4, pp. 250–256, 2008.

[5] F. Hopfgartner, T. Brodt, J. Seiler, B. Kille, A. Lommatzsch, M. Larson, R. Turrin, and A. Serény, Benchmarking news recommendations: The CLEF NewsREEL use case, *ACM SIGIR Forum*, vol. 49, no. 2, pp. 129–136, 2015.

[6] M. Balabanović and Y. Shoham, Combining content-based and collaborative recommendation, *Commun. ACM*, vol. 40, no. 3, pp. 66–72, 1997.

[7] J. B. Shu, X. X. Shen, H. Liu, B. L. Yi, and Z. L. Zhang, A content-based recommendation algorithm for learning resources, *Multimed. Syst.*, doi: 10.1007/s00530-017-0539-8.

[8] Y. Koren, Factorization meets the neighborhood: A multifaceted collaborative filtering model, in *Proc. 14th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Las Vegas, NV, USA, 2008, pp. 426–434.

[9] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, Item-based collaborative filtering recommendation algorithms, in *Proc. 10th Int. Conf. World Wide Web*, Hong Kong, China, 2001, pp. 285–295.

[10] A. Felfernig, Koba4MS: Selling complex products and services using knowledge-based recommender technologies, in *Proc. 7th IEEE Int. Conf. E-Commerce Technology*, Munich, Germany, 2005, pp. 92–100.

[11] A. Felfernig and K. Shchekotykhin, Debugging user interface descriptions of knowledge-based recommender applications in *Proc. 11st Int. Conf. Intelligent User Interfaces*, Sydney, Australia, 2006, pp. 234–241.

[12] J. Wang, A. P. De Vries, and M. J. T. Reinders, Unifying user-based and item-based collaborative filtering approaches by similarity fusion, in *Proc. 29th Annual Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Seattle, WA, USA, 2006, pp. 501–508.

[13] B. L. Wang, J. H. Huang, L. B. Ou, and R. Wang, A collaborative filtering algorithm fusing user-based, item-based and social networks, in *Proc. 2015 IEEE Int. Conf. Big Data*, Santa Clara, CA, USA, 2015, pp. 2337–2343.

[14] H. Koohi and K. Kiani, User based collaborative filtering using fuzzy c-means, *Measurement*, vol. 91, pp. 134–139, 2016.

[15] X. Y. Liu, C. Aggarwal, Y. F. Li, X. N. Kong, X. Y. Sun, and S. Sathe, Kernelized matrix factorization for collaborative filtering, in *Proc. 2016 Siam Int. Conf. Data Mining*, Miami, FL, USA, 2016, pp. 378–386.

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

[17] R. Salakhutdinov, A. Mnih, and G. Hinton, Restricted Boltzmann machines for collaborative filtering, in *Proc. 24th Int. Conf. Machine Learning*, Corvalis, OR, USA, 2007, pp. 791–798.

[18] H. L. Xu, X. Wu, X. D. Li, and B. P. Yan, Comparison Study of internet recommendation system, (in Chinese), *J. Softw.*, vol. 20, no. 2, pp. 350–362, 2009.

[19] M. K. Najafabadi, M. N. Mahrin, S. Chuprat, and H. M. Sarkan, Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data, *Comput. Hum. Behav.*, vol. 67, pp. 113–128, 2017.

[20] N. Gao and M. Yang, An improved unifying tags and ratings collaborative filtering for recommendation system, (in Chinese), *J. Nanjing Nor. Univ. (Nat. Sci. Ed.)*, vol. 38, no. 1, pp. 98–103, 2015.

[21] H. Koohi and K. Kiani, A new method to find neighbor users that improves the performance of collaborative filtering, *Expert Syst. Appl.*, vol. 83, no. C, pp. 30–39, 2017.

[22] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, Using collaborative filtering to weave an information tapestry, *Commun. ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[23] Z. D. Zhao and M. S. Shang, User-based collaborative-filtering recommendation algorithms on hadoop, in *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining*, Phuket, Thailand, 2010, 478–481.

[24] H. Ji, J. F. Li, C. R. Ren, and M. He, Hybrid collaborative filtering model for improved recommendation, in *Proc. 2013 IEEE Int. Conf. Service Operations and Logistics, and Informatics*, Dongguan, China, 2013.

[25] S. Y. Wei, N. Ye, S. Zhang, X. Huang, and J. Zhu, Item-based collaborative filtering recommendation algorithm combining item category with interestingness measure, in *Proc. 2012 Int. Conf. Computer Science and Service System*, Nanjing, China, 2012, pp. 2038–2041.
[26] G. Karypis, Evaluation of item-based top-N recommendation algorithms, in *Proc.* 10^{th} *Int. Conf.* Information and Knowledge Management, Atlanta, GA, USA, 2001, pp. 247–254.
[27] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, GroupLens: An open architecture for collaborative filtering of Netnews, in *Proc. 1994 ACM Conf. Computer Supported Cooperative Work*, Chapel Hill, NC, USA, 1994, pp. 175–186.

Chunxia Zhang is currently a master student in the School of Computer Science and Technology, Nanjing Normal University, and received the BS degree from Jiangsu University of Science and Technology in 2015. Her research interests include machine learning and recommend system.

Jing Lv is currently a PhD candidate in the School of Mathematical Sciences, Nanjing Normal University. Her research interests include machine learning and pattern recognition.

Ming Yang received the PhD degree from Southeast University at Nanjing in 2004. He received the MS degree from University of Science & Technology of China, and BS degree from Anhui Normal University, in 1990 and 1987, respectively. He is currently a professor in the school of computer science and technology at Nanjing Normal University. His research interests include data mining and knowledge discovery, machine learning, pattern recognition, and their applications.

Wanqi Yang received the PhD degree from Nanjing University in 2015. Her research interests include multiview learning, feature selection, multimodal fusion, abnormal event detection and activity recognition. She has published several papers in top conferences and journals, e.g., IEEE TNNLS, CVIU. Her work currently focuses on multiview dimension reduction, crossview correlation analysis and their applications to real-world problems in image/video analysis.