International Journal of u- and e- Service, Science and Technology
Vol.8, No.4 (2015), pp.247-264
http://dx.doi.org/10.14257/ijunesst.2015.8.4.23

A Collaborative Filtering Recommender System Integrated with Interest Drift based on Forgetting Function

Wu Sen [1] , Zhang Xiaonan [2] and Du Yannan [3]

[1,2,3] Donlinks School of Economics and Management, University of Science and Technology Beijing, China,
wusen@manage.ustb.edu.cn [1], zhangxiaonan17@hotmail.com [2] ,
duyannan89757@126.com [3]

Abstract

Collaborative filtering (CF) is one of the most prevailing and promising approaches in recommender systems. The algorithms precision of collaborative filtering has attracted ever-increasing study of researchers. Traditional user-based approaches for collaborative filtering identify user similarity by analyzing the co-rating items between users and utilize user similarity as predicted weight in order to evaluate the importance of rating from a user on an item. However, other factors are not taken into account, including users' rating trend and changes of user interest, which will degrade the accuracy of the recommendation result obviously. Therefore, in this paper, user similarity index is introduced to improve user similarity calculation. To assign decreasing weights to dated data, exponential function is implemented to redefine the weight of each item rated at different times. Combining user similarity index with exponential function as the improved algorithm, this paper re-computes the predicted ratings based on traditional user-based CF using Pearson Correlation Coefficient. Experiments on Movielens dataset have shown that the improved algorithm is superior to the traditional one.

Keywords:

1. Introduction

Due to the development of the Internet and information technology, now people can access massive amount of information more conveniently than ever before, but simultaneously spend much more time searching for what they really want, which is called information overload. Generally, traditional searching engines can only retrieve the most relevant items, but can hardly satisfy people with their specific needs and wants. Therefore, based on the concept of personalization recommender systems emerge as time requires, aiming at serving users' individual demands and enhancing user experiences. Recommender systems have already developed into a kind of information service, and especially have been widely used in e-commerce to recommend precisely considering people's diverse tastes and preferences. It utilizes user information and user profiles to predict the utility or relevance of a certain item, then to provide personalized recommendation [1].

There is no doubt that collaborative filtering has become the most successful recommender system nowadays. By comparing with similar users or items, collaborative filtering system can filter worthless information and provide users with new items without the limitation of forms. Presently, user-based collaborative filtering is very popular due to its effectiveness and efficiency. It generates k-nearest neighbors to an active user using similarity based vector model. After k-nearest neighbors found, their corresponding user-item matrix is aggregated to identify the set of items to be recommended (Top-N list) [2]. Though traditional collaborative filtering has partly achieved success, there are still many problems, such as sparsity problem and cold start problem in original user data

ISSN: 2005-4246 IJUNESST

matrix. In practice, compared to the vast information of users and items, information of user opinion on items is usually sparse. Therefore it is difficult to analyze and estimate user interests, particularly interests of cold start users [3]. Another issue is that traditional approach is carried out on the premise that user interests never change and ratings at different times are weighted equally, which absolutely produce low precisions [4]. To solve the latter problem, based on the typical user-based algorithm, a new algorithm is put forward in this paper to enhance the precision of recommender systems.

The structure of this paper is organized as below: Firstly, section 2 briefly outlines traditional collaborative filtering algorithms and related work which has been done in algorithm improvement. In section 3, characteristic of practical recommender systems in China is analyzed to understand how recommendation methods are utilized. In section 4, the newly proposed algorithm is demonstrated and explained in detail. Afterwards, a series of experiments are illustrated in section 5 to evaluate the new algorithm. Finally, the conclusion of this study is presented in section 6.

2. Related Work

Over the years, various algorithms have been explored to optimize the application of collaborative filtering, including using psychological model and time model involving human forgetting features.

2.1 Collaborative Filtering Algorithm based on Psychological Model

Actually, in recommender systems trust mechanism can help users to accept others' suggestions, which means users believe that they'll benefit from the recommendation and can save more energy and time on the way of finding the best for themselves.

Based on this, scholars have already begun to improve traditional algorithms from the psychological perspective. Some scholars create concepts of global trust [5] and local trust [6] to enhance the overall effectiveness and carry out recommendation strategy for those without neighbor users, that is to say, it can solve the cold start problem partially [7]. Another paper [8] focuses on user interest change and the credibility of ratings data. Firstly, in the course of user similarity calculating, user ratings are weighted by a time attenuation function and a credit assessment function and then several users highly similar with the active user are selected as their neighbors.

Filtering algorithm combined with psychological model is proved to have a better understanding to some extent and thus a better recommendation results when it is contrasted with traditional approaches, which undoubtedly provides a new orientation into further implementation.

2.2 Collaborative Filtering Algorithm Considering Time Factors

With the fast growth of e-commerce, many collaborative filtering applications have been used for a long time and many websites have accumulated tens of millions of user ratings, some of which are very old [9]. So the validity and value of those out-of-date data need to be considered, which means that time model is another up-to the- moment issue.

To fix the problem above, user interest drift is considered. Therefore, paper [10] proposes a new forgetting function with normal distribution density to accommodate interest change. User interest is defined in a hybrid model that contains both long and short-term components. The long-time component is renewed by using the normal incremental forgetting distribution algorithm. The short-time component is renewed by removing the least recently used algorithm and adding recent interest through sliding window. A novel collaborative filtering system is presented in [11]. It is based upon the forgetting curve, traces user interests and classifies user interests into long-term and short-term interests, well reflecting the status quo.

Due to the existing time weight allocation and user interest drift problems, this paper focus on coming up with new user similarity measures to get a more proper k-nearest neighbors list and applying a psychological model and new fitting function based on H. Ebbinghaus forgetting curve into time weight function. Then the new collaborative filtering algorithm is proposed to generate the final recommendation list, which is confirmed to have a better understanding and a superior recommendation result compared with traditional approaches.

3. Characteristic Analysis of Practical Recommender Systems in China

It was Amazon that promoted the development of recommender systems at the early stage. In a manner of speaking, recommender system is the product of the certain stage of e-commerce development. Currently, recommender systems exist everywhere in people's life. As a consequence, it's necessary to analyze how Chinese websites utilize and integrate recommender systems. Samples in this paper are e-commerce, video and music websites, etc.

3.1 Recommender Systems in E-commerce Websites

3.1.1. Recommender System on JD.com Website: Back to year 2012, there were two ways for JD.com to recommend. One was Guess You Like module on its homepage, which still exists now. The other was Recommended for You module on navigation bar on the top of the page, which is changed into Interested Items of Similar User module on left sidebar.

Figure 3.1. Guess You Like Module on JD.com Website

For current Guess You Like module, recommendation results will not change with the logging status of users, but will totally change when cookies are cleaned. According to this, it's reasonable to infer that this module recommends based on users' browsing history and content-based recommendation.

Figure 3.2. Interested Items of Similar User Module on JD.com Website

Another module for now is Interested Items of Similar User module. It recommends based on user purchasing records or browsing histories. Each recommended item has a link with other personalized recommended items. Those items have been bought or browsed by users that have bought or browsed the current item. It could use item-based collaborative filtering to give recommended items. And there is nowhere for users to give feedback about the recommendation.

Other recommendation form on JD.com is Recommended for You on its members club module, but it's non-personalized. So it is not discussed here.

3.1.2. Recommender System on Dang Dang Website

Figure 3.3. Guess You Like Module on Dang Dang Website

Dang Dang only provides login users with Guess You Like module and it's really obvious. It recommends based on user purchasing records or browsing histories, thus it could adopt item-based collaborative filtering from recommended content. In aspect of user feedback, it offers no feedback.

Figure 3.4. Interested Items of Similar Users Module on Dang Dang Website

Dang Dang offers Interested Items of Similar Users module just below the items that user is looking for. Each recommended item has a link with items which have been bought by users that has bought current item and it is personalized. From the title and content, it can be inferred that item-based collaborative filtering is utilized here.

3.1.3. Recommender system on Vancl Website

Figure 3.5. Recommended for You Module on Vancl

Vancl only provides recommendation to login users. Unlike the other two, it divided recommendation results into several groups based on different recommended reasons. Users can get recommended items actively by choosing browsing history, purchase records, shopping cart or favorites. Among these, purchase records and favorites groups are probably item-based collaborative filtering. The other two groups could be content-based recommendation. There is no user feedback for Recommended for You module on Vancl.

3.2 Recommender Systems in Video and Music Websites

3.2.1. Recommender System on YOUKU Website:

Figure 3.6. Recommended for You Module on YOUKU

YOUKU has recommendation on several pages and are all on top of the site's page. The core modules are called Recommended for You, which have no difference in form of recommendation compared with other websites. But when users refresh the recommended list, it will change and strength the items that related to those that have been viewed. From this it can be surmised that it calculates partly online. And with further exploration and understanding, although it provides replacement function, repetition still exists which means it includes offline calculation as well. In consequence, it's very possible that it uses hybrid computing strategy.

3.2.2. Recommender System on Xiami Website

Figure 3.7. Full Site Recommendation on Xiami

Xiami is a famous music platform. As a matter of fact, Xiami is a full site music recommender system. Its recommendation method fuses many forms to help user find their interested music, such as editor's choice, algorithm recommendation, multi-user share and so on. This kind of combination of manual and algorithm can be the future direction. It can be seen from the recommended reason that recommended music is based on users' favorites and listening histories, for example, a piece of music, an album or an artist. Recommended item is only in form of music. Xiami provides users with feedback button, including positive and negative feedback. In addition, in a short period of time the change can be seen. Therefore, both online and offline calculation strategies are possibly adopted here.

3.2.3. Recommender System on Douban.fm Website

Figure 3.8. Full Site Recommendation on Douban.fm

Douban.fm is another typical music recommender system. Its slogan “music is an exploration, so forget about the playlist” is showed on its website, which can reflect it is different from traditional music recommender systems. Douban.fm only offers a simple interface and user can give positive and negative feedback. The system will recommend to users based on their feedback. Douban.fm can revise its recommendation list in time and can analyze users’ preference to overall music properly. According to these, it can be inferred that content-based recommendation and item-based collaborative filtering are used in Douban.fm.

3.3 Recommender Systems for Small Websites

3.3.1. Recommendation Service Provider – Ujian: Ujian is a recommendation service provider, using semantic analysis technology to match the most probably interested content to users. Ujian is a full site recommendation website; helping text dominated small websites, such as blog with recommendation service. It probably uses content-based method. Recommended content comes from websites with its plug-in. Ujian scans and stores this content for computing.

3.3.2. Recommendation Service Provider – Wumii: Wumii used to be a combination of recommendation service provider and online article recommendation. But now, it becomes to focus on providing service for small websites. When users visit websites with wumii plug-in and read articles, the most relevant articles will be recommended to users to save their time. It probably uses content-based method. Recommended content comes from websites with its plug-in.

Table 3.1 is the comparison and summary of characteristics of practical recommender systems in China mentioned above.

Table 3.1. Comparison of Characteristics of Practical Recommender Systems in China

Site Category	Site Name	Module Name	Recommended Content	Data Source	Method	Feedback	Calculative Strategy	Give Reason
E-commerce	JD.com	Guess you like	Item	Browsing	CB	None	---	Yes
		Interested items of similar user	Item	Browsing/ Purchase	IBC	None	---	Yes
	DangDang	Guess you like	Item	Browsing	IBC	None	---	No
		Interested items of similar user	Item	Browsing/ Purchase	IBC	None	---	Yes
	Vancl	Recommended for you	Item	Browsing/ Purchase	CB/IBC	None	---	Yes
Video, Music	YOUKU	Recommended for you	Video	Browsing	CB/IBC	None	Hybrid	No
	Xiami	Full site	Music	Favorite	CB/IBC	+	Hybrid	Yes
	Douban.fm	Full site	Music	Favorite	CB/IBC	+/-	Hybrid	No
Recommendation Service Provider	Ujian	Plug-In	Article	---	CB	---	Offline	No
Recommendation Service Provider	Wumii	Plug-In	Article	---	CB	---	Offline	No

*CB represents Content Based recommendation, IBC stands for Item-Based Collaborative Filtering recommendation. All the methods are surmised based on the external feedback of each website, so there may be differences with internal mechanism.
**--- means there isn't enough information to make a judgment.

4. Proposed Algorithm

4.1 Improved Measure of User Similarity

As the core problem of user-based collaborative filtering is user similarity measurement [12], it's quite important to improve its accuracy. Traditional similarity measures are Pearson Correlation Coefficient, Cosine-based Similarity and Adjusted-Cosine Similarity. For Pearson Correlation Coefficient based collaborative filtering, it's probably that when two users have only several co-rating items and they happen to have the same rating tendency, then they'll be calculated as highly similar, which may not be the truth. To fix the defect of traditional similarity measures, based on Pearson Correlation Coefficient this paper innovatively define user similarity index $SI(u,v)$ as below:

$$SI(u,v) = \frac{N_{co-rating}}{N_u + N_v} \quad (4.1)$$

$N_{co-rating}$, N_u and N_v respectively are number of the co-rating items and numbers of rating items of user u and user v. Therefore, adjusted Pearson Correlation Coefficient $SIM(u,v)$ is calculated by formula below:

$$SIM(u,v) = \frac{(\sum_{i \in I_{uv}} (R_{u,i} - \overline{R_u})(R_{v,i} - \overline{R_v})) \times SI(u,v)}{\sqrt{\sum_{i \in I_{uv}} (R_{u,i} - \overline{R_u})^2} \sqrt{\sum_{i \in I_{uv}} (R_{v,i} - \overline{R_v})^2}} \quad (4.2)$$

- m is number of users and n is number of items.
- User set is $U = \{u_1, u_2, u_3, \ldots, u_m\}$ and item set is $I = \{i_1, i_2, i_3, \ldots, i_n\}$.
- User-item rating matrix is $R = m \times n$, $R_{u,i}$ is rating of item i from user u and $\overline{R_u}$ is user u's average score.
- I_u is rating set of user u, $I_{uv} = I_u \cap I_v$.

4.2 Improved Time Weight Allocation Algorithm Adapting to User Interest Drift

German psychologist H.Ebbinghaus [13] finds out that human's forgetting process is regular but unbalanced. At the beginning, people forget very fast but turn slow gradually, and hardly forget after a long time period. Based on this feature and experimental data, he draws the Ebbinghaus forgetting curve. Figure 4.1 shows the drawn curve. The horizontal axis is time and the vertical axis shows the percentage of existing memory. It certainly testifies the conclusion that H.Ebbinghaus has made. This characteristic is analyzed and fitted with a new function as a parameter of the new time weight function proposed in this paper.

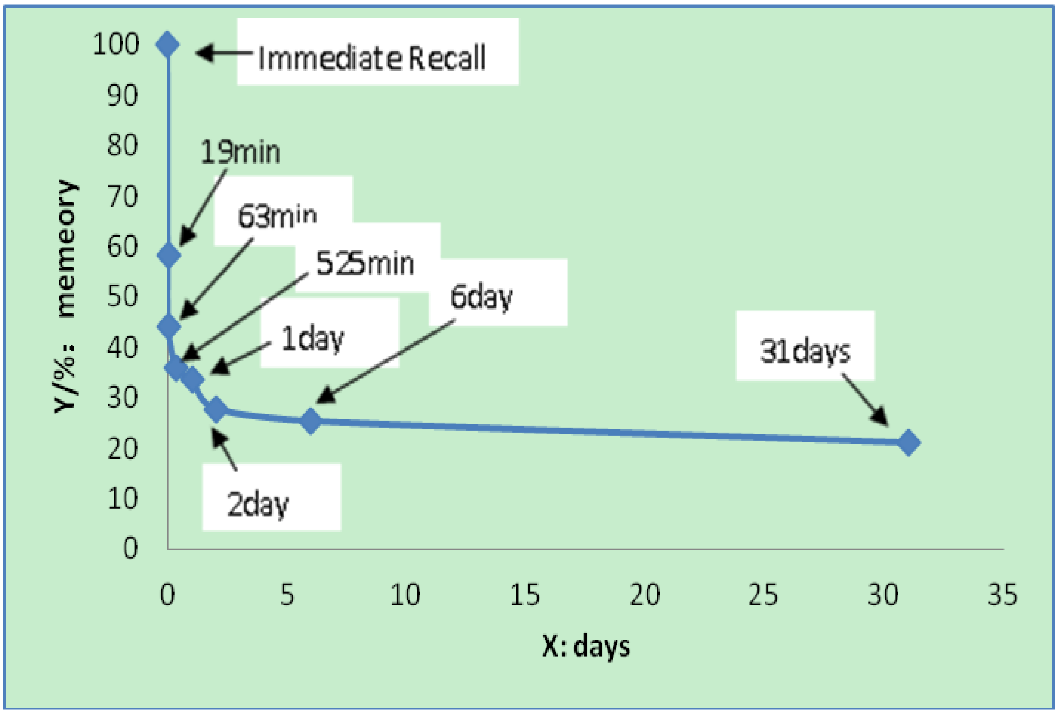

Figure 4.1. H.Ebbinghaus Forgetting Curve

In recommender systems, ratings for specific item can reflect users' subjective recognition of it, which will decline with time, resembling the forgetting tendency. According to this, the validity of rating items in recommender system is based on H.Ebbinghaus forgetting curve.

Paper [14] takes users' specific visiting time into account to capture the change of their interests. The two time weight functions are linear function [14] $y = 0.486 - 0.0109x$ and exponential function [4] $y = 0.50 + 0.50e^{-x}$.

In this paper, exponential function is used to compute time weight of ratings from user u on item i. It is:

$$Tweight(u,i) = e^{-a \times \frac{T_{ui}}{T_u}} \quad (4.3)$$

- $a \in (0,1)$ can be adjusted dynamically. To be specific, a higher value of a indicates a faster speed that user forgets.
- T_{ui} =time of user u gives a mark on item i minus the first time user u gives a mark in this recommender system.
- T_u is the time span that user u utilizes recommender system.
- Trend curves are as below as different values of parameter a.

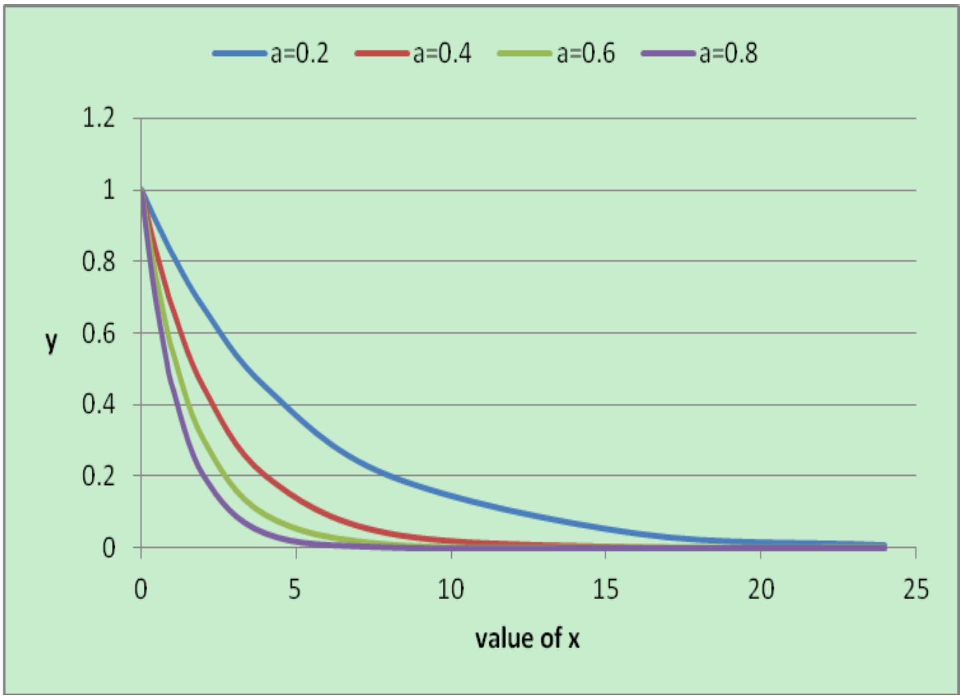

Figure 4.2. Time Weight Exponential Fitting Function

4.3 Improved Algorithm for Calculating Predicted Ratings

To get a better balance between user ratings and predicted weight, this paper harmonize user ratings with predicted weight to propose a new method to predict ratings so as to get the final score for each item and generate the recommendation list. It is computed as:

$$P(u,i) = \frac{\sum_{v \in knn} Tweight(v,i) \times SIM(u,v) \times (R_{v,i} / R_{v,\max})}{\sum_{v \in knn} SIM(u,v)} \quad (4.4)$$

- u is target user
- v is a neighbor of user u
- $R_{v,\max}$ is the highest score of user v
- $R_{v,i}$ is rating for item i from user v

4.4 Procedure of the Whole Algorithm

There're two strategies in selecting nearest neighbors in traditional collaborative filtering. One is to choose Top-N similar users, and the other is to choose those whose similarities are above a certain threshold. The first strategy, Top-N model, is adopted here.

Table 4.1 Process to Get Nearest Neighbor Model

Get Nearest Neighbor Model
Input: item- rating matrix $\left[\mathbf{I}(\mathbf{u_p},\mathbf{i_q})\right]_{\mathbf{m \times n}}$.
Step 1: according to similarity formula (4.2), get "item-ratings" similarity matrix $\left[\mathbf{sim}(\mathbf{u_i},\mathbf{u_j})\right]$;
Step 2: size $\left[\mathbf{sim}(\mathbf{u_i},\mathbf{u_j})\right]$ down for each row;

Step 3: select the first k users in the ranked matrix as user $u's$ nearest neighbors;

Output: Nearest Neighbor Model M .

After constructing the nearest neighbor model, recommendation list can be generated. The outline of the new algorithm can be summed up as below.

Table 4.2 Process to Generate Recommendation List

Generate Recommendation List

Input: target user u and his/her visited item set I_u, nearest-neighbor model and neighbors' visited item set I_M .

Step 1: read the nearest neighbor model M ; select all the rated items from k-nearest neighbors as initial candidate set $Candidate$;

Step 2: delete items that target user have already rated in $Candidate$, then get the ultimate candidate set $C = Candidate - I_u$;

Step 3: according to formula (4.3), calculate time weight for each item in C ;

Step4: use formula (4.4) to predict ratings for each item in C ;

Step5: select Top-N scoring items as the Top-N recommendation list for user u .

Output: Top-N recommendation list for user u .

5. Experiment and Evaluation

Experiment and evaluation is one of the most important parts in proposing new algorithm since it is the way to objectively assess the effect of the improved algorithms.

5.1 Datasets

To evaluate the new algorithm in this paper, Movielens 100k dataset is used here. Movielens [15] was published by Grouplens project team of Minnesota University in America. 1682 users rank 943 movies that they've ever seen and scoring range is 1-5 points. It also contains the timestamp for each score, starting from 1970-1-1. Each user has at least 20 records. Movielens has three datasets of different sizes to adapt to different-sized algorithms. The smallest one has 100,000 pieces of data, while the biggest one has 1 million pieces. Small-scale dataset is used in the experiment. The dataset is analyzed and found that ratio of the numbers of users to numbers of items is at 56% and the sparsity is 99.36%.

5.2 Evaluation Criteria

With the development of recommender systems, it becomes an important subject to evaluate how efficient a recommender system can be. Evaluation methodology has also become an independent research area. Currently, common evaluation criteria for recommender system are prediction precision, coverage rate, diversity and credibility.

Prediction precision and coverage rate will be introduced and used as the evaluation criteria in this paper.

5.2.1. Predict Precision: Paper [16] and paper [17] innovatively introduce precision into evaluation of recommender systems. Precision represents the possibility that a user interested in recommended items and measures the ability that a recommender system predicts user behaviors, which is one of the most important off-line evaluation indicators. The calculation can be done off-line which provides convenience for research. Precision is calculated as:

$$precision = \frac{Hits}{N_s} \quad (5.1)$$

- N_s is total number of recommended items.
- $Hits$ is number of user's interested items in recommendation list.

5.2.2. Coverage Rate: Coverage rate is an effective index that can measure the ability of recommender systems, including item mining and long-tail effect eliminating. Suppliers usually pay close attention to coverage rate. It is because a higher coverage rate indicates consumers can have access to items as widely as possible and recommender systems can have a better user interest mining ability. As a result of this, a great recommender system should be not only excellent in predict precision, but also in coverage rate. Coverage rate is defined as:

$$\mathrm{cov}erage = \frac{I_{Hits}}{I_s} \quad (5.2)$$

- I_s is the total kinds of items in recommender systems.
- I_{Hits} is the kinds of user's interested items in recommendation list.

5.3 Results and Discussions

Movielens dataset is utilized as the experimental dataset and three trials have been made. One experiments used to test and set reasonable time weight function parameter. One is used to analyze the impact of improved similarity calculation measurement. The third one is used to analyze influence of time weights on recommendation results. Pearson Correlation Coefficient collaborative filtering referred to in section 4.1 is used as comparison experiment. To illustrate, CF is short for traditional Pearson Correlation Coefficient collaborative filtering. SCF represents collaborative filtering with improved similarity calculation measurement in this paper. TSCF is collaborative filtering combing improved similarity calculation with time weight function.

When generating Top-N recommendation list, contrast SCF with CF and TSCF with SCF. In the ultimate recommendation list, N represents number of recommended movies and k represents number of neighbors, both of which lie between 5 and 25 and increase by 5. Experimental results are as below.

5.3.1. Time Weight Function Parameter Setting: On the basis of traditional user-based collaborative filtering, add time attenuation function into predicted ratings to represent user interest change. With different values of parameter a, recommendation results can be observed at different rate of forgetting. In this experiment, set k equal to 20 and N equal to 5. Parameter a ranges from 0.2 to 0.7 increasing by 0.1. From Fig5.1, the algorithm considering time factor is more effective than traditional one, except when a is 0.2 and 0.3.When a is 0.6, precision is most excellent reaching 9.17% indicating most close to law of people's forgetfulness.

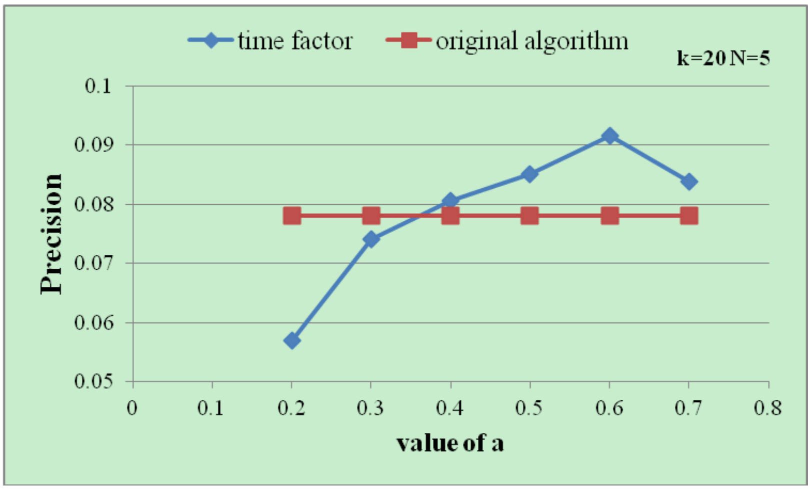

Figure 5.1. Contrast Influence of Time Weight Function Parameter on Recommendation Results

5.3.2. Influence of Improved Similarity Calculation Measurement on Recommendation Results: This part contrasts SCF with CF to see the improvement of improved similarity calculation. According to Figure 5.2., precision increases with value of N. Accuracy of both SCF and CF experience a downward trend. They get to their summit separately when N is 5 and get to their bottom when N is 25.

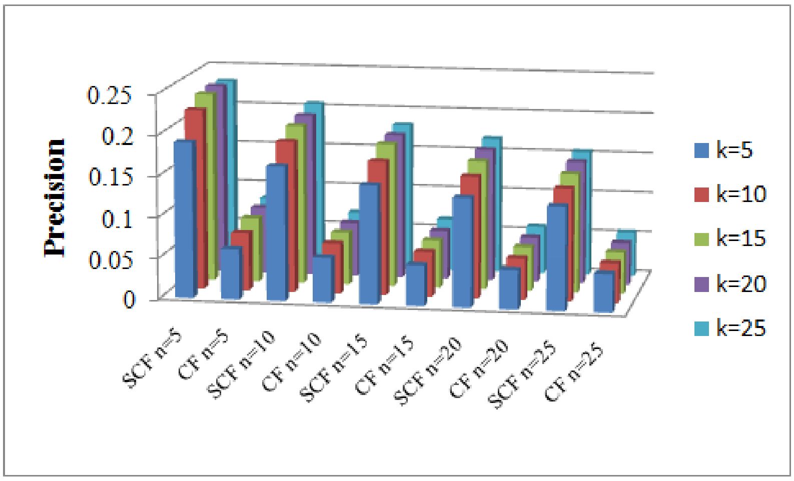

Figure 5.2 Contrast Influence of SCF with that of CF on Recommendation Results

Figure 5.3 demonstrates the result when the value of N is set to 5 in detail. Precision increases with value of k. By comparing both of the two algorithms, it's obvious that value of precision of CF is always lower than that of SCF, which is to say SCF is the more precise algorithm. Accuracy of both CF and SCF experience an upward trend. They get to their bottom when k is 5.

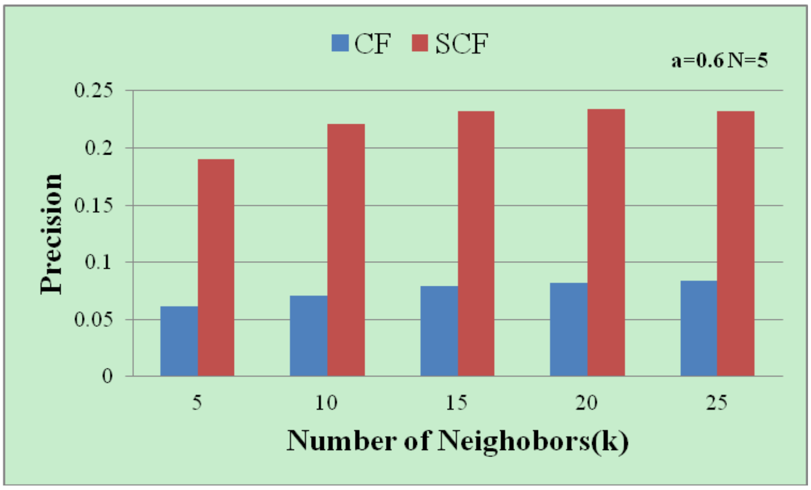

Figure 5.3 Precision Contrast of SCF with that of CF on Recommendation Results when N is 5

Based on the analysis in paper [18], when the number of nearest neighbors is 20, recall ratios increases maximally. As a result of this, 20 neighbors (k=20) are chosen here to demonstrate the precision.Table5.1 below describes the improved percentage within the two algorithms.

Table 5.1 Improved Precision of SCF and CF on Recommendation Results

N	Algorithm(k=20)	
	SCF	CF
5	23.39%	8.21%
10	19.97%	6.63%
15	17.84%	5.98%
20	16.31%	5.51%
25	15.16%	5.20%

Compared with the original similarity algorithm, it can be seen that there is a considerable improvement in the modified one. Accuracy of both algorithms reduces with the increase of number of recommended movies. When N is 5, the strength is the most obvious. SCF improves by 15.18% in contrast with CF.

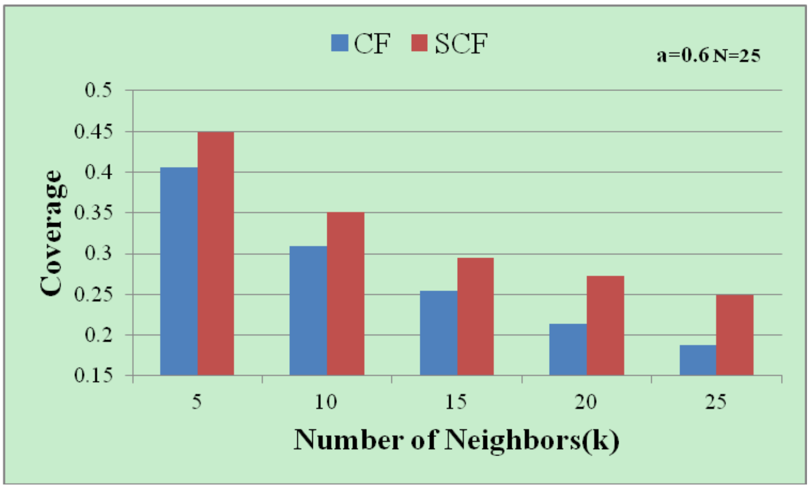

Figure 5.4 Coverage Contrast of SCF with that of CF on Recommendation Results when N is 25

From the point of coverage rate, SCF outstands through the comparison with CF as well. Coverage of both CF and SCF experience a downward trend with the number of neighbors increasing. Coverage rate is increased averagely by 4.87% when N is 25.

Table 5.2 Improved Coverage of SCF and CF on Recommendation Results

k	Algorithm(N=25)	
	SCF	CF
5	44.83%	40.56%
10	35.08%	30.98%
15	29.43%	25.39%
20	27.23%	21.40%
25	24.91%	18.79%

To conclude, according to the experiment, SCF has shown to be positively effective in Movielens dataset in both predict precision and coverage rate.

5.3.3. Influence of Time Weight on Recommendation Results: This part contrasts TSCF with SCF to see the influence of time weight on recommendation result.

It can be seen from Figure 5.5 that the accuracy of both proposed algorithms decrease with the increase of number of recommended movies, starting from nearly 23% and ending at nearly 12%.

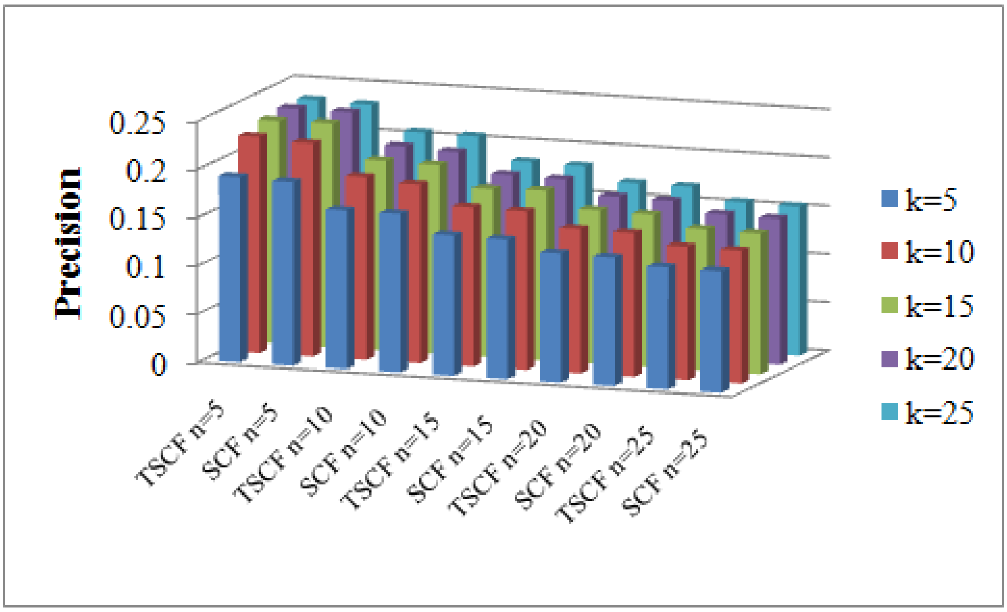

Figure 5.5 Contrast Influence of TSCF with that of SCF on Recommendation Results

N is still set to 5 and it can be seen from Figure 5.6 that the accuracy of both proposed algorithms experience upward trends with the increase of number neighbors from nearly19% to23%.

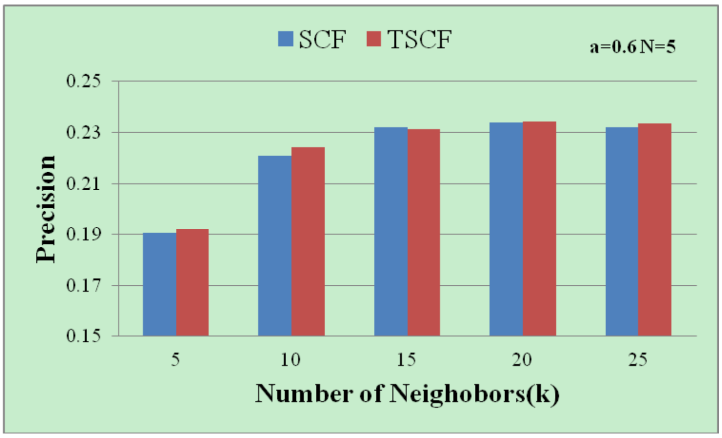

Figure 5.6. Contrast Influence of TSCF with that of SCF on Recommendation Results when N is 5

Below is the improvement percent within the two algorithms and still 20 neighbors are chosen to demonstrate the results.

Table 5.3. Improved Precision of TSCF and SCF on Recommendation Results

N	Algorithm(k=20)	
	TSCF	SCF
5	23.41%	23.39%
10	20.19%	19.97%
15	17.96%	17.84%
20	16.38%	16.31%
25	15.25%	15.16%

From Table5.3, it can be known that algorithm concerning user interest drift is more accurate than the one without thinking about time factors.

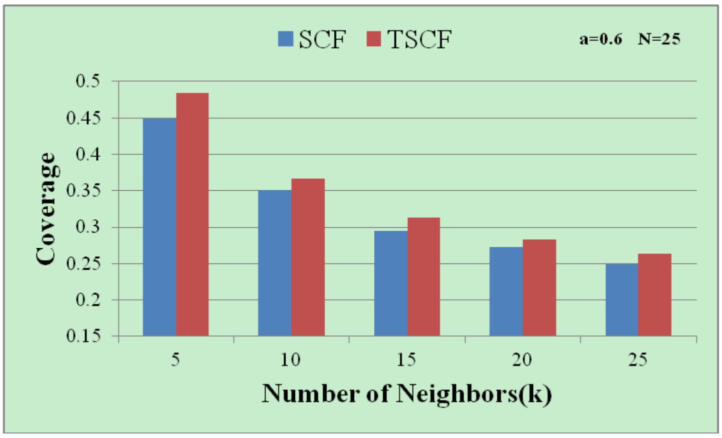

Figure 5.7. Coverage Contrast of TSCF with that of SCF on Recommendation Results when N is 25

From the coverage rate perspective, TSCF outstands through the comparison with SCF, too. Coverage of both SCF and TSCF experience a downward trend with the number of neighbors increasing. Coverage rate is increased averagely by 1.90% when N is equal to 25. When k is 5, it improves the most, reaching 3.57%.

Table 5.4. Improved Coverage of TSCF and SCF on Recommendation Results

k	Algorithm(N=25)	
	TSCF	SCF
5	48.40%	44.83%
10	36.68%	35.08%
15	31.27%	29.43%
20	28.30%	27.23%
25	26.34%	24.91%

In this segment, the contrast shows that time factor does have influence on recommendation, especially for coverage improvement, and improves precision by 3.57% mostly in Movielens dataset.

6. Conclusion

In this paper, three issues are presented. One is theoretical analysis of characteristics of practical recommender systems in China. The others are experimental analysis of two common problems in traditional collaborative filtering system, which are optimized user similarity measurement and reasonable time weight allocation.

Characteristics analysis of practical recommender systems in China involves recommendation module analysis, recommended content analysis, recommendation methods analysis and so on in practical recommender systems. Based on user similarity index, a new user similarity function is proposed. Experiments indicate that recommendation combined with new user similarity function (SCF) can partly eliminate the situation when two users have only several co-rating items. When considering time factors, the improved collaborative filtering algorithm (TSCF) is put forward using exponential function so as to allocate different time weights to adapt to user interest drift. Contrast experiments in Movielens dataset are done to test and verify its validity. Both SCF and TSCF have outperformed original algorithms.

Acknowledgments

The authors would like to thank the research support by National Natural Science Foundation of China under Grant Number 71271027, by the Fundamental Research Funds for the Central Universities of China under Grant No. FRF-TP-10-006B and by the Research Fund for the Doctoral Program of Higher Education under Grant No. 20120006110037.

References

[1] F. Cacheda, V. Carneiro, D. Fernández and V. Formoso, "Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems", *ACM Transactions on the Web (TWEB)*, vol. 5, no. 1, (**2011**), p. 2.

[2] S. Puntheeranurak, "A Multi-clustering Hybrid Recommender System", *Computer and Information Technology*, 7th IEEE, 2007-10, (**2007**), pp. 223-228.

[3] Y. Zukun, S. Wei, Z. Xiaolin and C. Deren, "A Recommender System Model Combining Trust with Topic Maps", In *Web Technologies and Applications*, Springer Berlin Heidelberg, (**2013**), pp. 208-219.

[4] Z. Yuchuan and L. Yuzhao, "A Collaborative filtering algorithm based on time period partition", In *Intelligent Information Technology and Security Informatics (IITSI), 2010 Third International Symposium on* IEEE, (**2010**), pp. 777-780.

[5] L. Zhubing, "Study on Trust Relationship Based Collaborative Filtering Recommender Strategy", *WestSouth University*, Chong qing, (**2008**).

[6] J. Yaya and M. Yuanchao, "Collaborative Filtering Recommendation Algorithm Based on Improved Trust worthiness", *New Technology of Library and Information Service*, (**2010**), p. 10.

[7] S. Lei, "Improving Collaborative Filtering with Psychology Model", *Beihang University*, (**2010**).

[8] C. Zhimin, J. Yi and Z. Yao, "A Collaborative Filtering Recommendation Algorithm Based on User Interest Change and Trust Evaluation", *JDCTA*, vol. 4, no. 9, (**2010**), pp. 106-113.
[9] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts", *Machine learning*, vol. 23, no. 1, (**1996**), pp. 69-101.
[10] G. Xinming and Y. Gaizhen, "A hybrid algorithm to track drift of user's interests", *CAAI Transactions on Intelligent Systems,* vol. 5, no. 2, (**2010**), pp. 181-184.
[11] Y. Hong and L. Zhuanyun, "A collaborative filtering recommendation algorithm based on forgetting curve", *Journal of Nanjing University (Natural Sciences)*, vol. 5, (**2010**), pp.006.
[12] C. Chunsheng, Z. Zhenchun, L. Feng and Q. Ying, "Application of vague set in recommender systems", *Journal of System and Management Sciences,* vol. 3, no. 1, (**2013**), pp. 74-86.
[13] E. Hermann, "Memory: a contribution to experimental psychology [EB/OL]", [2011-12-09]. http://psy.Ed.asu.edu/~classics/Ebbinghaus/index.htm, (**2011**).
[14] Z. Xianrong, T. Zeying and C. Xianbin, "Non-lineal gradual forgetting collaborative filtering algorithm capable of adapting to users' drifting interest", *Computer aided engineering*, vol. 16, no. 2, (**2007**), pp. 69-73.
[15] MovieLens Dataset [OL]. [2011-10-5]. http://MovieLens.umn.edu/
[16] D. Billsus and M J. Pazzani, "Learning Collaborative Information Filters", In *ICML*, vol. 98, (**1998**), pp. 46-54.
[17] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, „Application of dimensionality reduction in recommender system-a case study (No. TR-00-043)", *Minnesota Univ Minneapolis Dept of Computer Science,* (**2000**).
[18] D. Yi and L. Xue, "Time weight collaborative filtering", *In Proceedings of the 14th ACM international conference on Information and knowledge management*, ACM, (**2005**), pp. 485-492.

