A New Approach for Improving Collaborative filtering Recommender Systems

Zhipeng Tang [a,*] and Zhengping Jin [b]

Beijing University of Posts and Telecommunications, P.R.China

[a] tangzhipeng1993@126.com, [b] zhpjin@bupt.edu.cn

*corresponding author

Keywords: collaborative filtering, user-based approach, time-weighted, dynamic multi-level, recommendation system.

Abstract. Nowadays, collaborative filtering (CF) is the most widely used method for proving recommendations in online environments, which is designed to filter large amounts of information in order to recommend to the user what they may be interested in. The user-based CF calculates the similarity between users by comparing the ratings of users to the common item, the rating generated at different times are weighted equally, but the effect of time on the user is not considered. In this paper, we propose a dynamic multi-level and time-weighted (DMLTW) collaborative filtering recommendation algorithm. Based on the original time-weighted, a new user-based time-weighted function is proposed, considering the influence of time factor on the recommendation result. At the same time, we present a positive and negative adjustment method to divide the user's similarity into different levels so as to achieve better recommendation quality. Experimental results show that our proposed method improves the accuracy of the user-based recommender systems and has a lower MAE compared to the reference method.

1. Introduction

With the rapid growth of the internet, the problem of information overload is more and more serious: the emergence of a large number of information makes it difficult for users to quickly locate the useful information and reduce the efficiency of using information. At present, the recommender systems have become the main way to help users filter large amounts of information, which aims to help them to choose item, thus alleviating the problem of information overload. There are many recommender systems have been put forward and practical applications, such as Amazon [1], YouTube [2], Netflix [3] and so on.

In the recommender systems, collaborative filtering (CF) is the most widely used technology. It can be divided into memory-based CF and model-based CF. Memory-based CF includes user-based CF and item-based CF [4]. User-based CF recommendation algorithm by calculating the similarity between user and other users, and then find the similar users, using the similar user's rating and similarity to predict the user's rating for item. There are many approaches to calculate similarity, such as Cosine Similarity (COS), Pearson Correlation Coefficient (PCC), and Euclidean.Distance(ED), etc. Item-based CF recommendation algorithm is similar to the user-based CF recommendation algorithm. Model-based CF recommendation algorithm is mainly to establish model, through training data to estimate some parameters of the model, and then use the model to predict the user's rating for item. The common models include latent factor model, clustering [5], Bias model [6], context awareness model [7], etc.

However, the accuracy of the prediction is still the main problem of collaborative filtering recommender systems. Y Ding [8] et al proposed time weight in the recommender systems, by different items to give different weights to enhance the effectiveness of the recommender systems. Y Koran [9] et al proposed a more sensitive method to better distinguish transient effects and long-term models, and demonstrated the time-varying behaviour how to be modelled throughout the life of the data. Besides, G Koutrika [10] et al proposed a similarity calculation method called Jacquard, which based on the number of common items between users. N Politicise [11] et al proposed a multi-level recommendation method in which similarity is divided into different levels and is enhanced if the similarity belongs to a certain level. M Gan [12] et al introduced a power-law adjustment of user similarity, with the aim is adjust the similarity between users, thus having high accuracy and increasing the diversity of the item. The accuracy of the recommender systems can be improved in many ways, but there is still a high room for improvement. In this paper, a dynamic multi-level and time-weighted (DMLTW) collaborative filtering recommendation algorithm is proposed, which combines the advantages of time-weighted recommendation algorithm and dynamic multi-level recommendation algorithm. The main work of the paper is as follows:

1. We implement a user-based time-weighted collaborative filtering recommender system, improve the recommendation algorithm based on positive and negative adjustments of the similarity values of users using constraints, and propose a dynamic multi-level and time-weighted collaborative filtering recommendation algorithm.

2. We have performed an extensive experimental evaluation of our proposed method using real datasets. In addition to that we have compared it against reference methods in order to show its effectiveness. Experimental results show that DMLTW has a smaller MAE compared with the reference method.

The rest of the paper is organized as follows. Section 2 briefly describes the problem definition of the recommender systems. In section 3, we propose a dynamic multi-level and time-weighted collaborative filtering recommendation algorithm. Section 4 describes our experimental work, providing detail information on our data sets, evaluation metrics, and results of different experiments. In the last section, we discuss and point out the direction of future work.

2. Preliminary

Collaborative filtering (CF) recommendation algorithm is defined as follows: Given the user set U and item set I, the user's interest for the item can be represented as a matrix R of | U | * | I |. In this matrix, each row vector represents a user’s rating set for all items, and each column vector represents all users rating sets for an item, each element $r_{u,i} \in$ R represents the rating of the user u for item i (the higher the score, the stronger the user's interest for the item). In general, the number of known ratings in R is far less than the number of unknown ratings. Given the rating set T ⊂ R as the training set, construct a recommender system according to the known ratings in T, and the system needs to be able to predict the unknown score in R with minimal cumulative error. The cumulative error of the recommender system will be validated on the test set V. To avoid over fitting, the data in test set V cannot be used to on the training process of the system, i.e. V ⊂ R and V ∩ T = Ø.

A common model for CF recommender systems is the nearest neighbour model. In the nearest neighbour model, the precondition of predicting the unknown rating is modelling the relationship between user and user. The k-nearest neighbour is a widely adopted model [13]. In the k-nearest neighbour model, for the user u, the system sorts the similarity between the user u and other users v ∈ (R - u) in the system in descending order, and record the k users that has the highest rating similarity with user u, the k users is the k-nearest set of the user u, denoted by in (u). When the user needs to make a prediction rating for item i, the prediction of the unknown rating $r_{u,i}$ is based on the rating of the neighbourhood users for item i, defined as:

$$r_{u,i} = \frac{\sum_{v\in kNN(u)\cap R(i)} r_{v,i}\times Sim(u,v)}{\sum_{v\in kNN(u)\cap R(i)} Sim(u,v)} \tag{1}$$

Where $r_{u,i}$ represents the prediction rating of user u for item i. R (i) represents the set of users who have rated on item i. Sim (u, v) represents the similarity between user u and user v. When calculating Sim(u,v) using Pearson Correlation Coefficient, the calculation method is as follows:

$$Sim(u,v) = \frac{\sum_{i\in I_u\cap I_v}(r_{u,i}- \bar{r_u})(r_{v,i}- \bar{r_v})}{\sqrt{\sum_{i\in I_u\cap I_v}(r_{u,i}- \bar{r_u})^2}\cdot\sqrt{\sum_{i\in I_u\cap I_v}(r_{v,i}- \bar{r_v})^2}} \tag{2}$$

Where $\bar{r_u}$ and $\bar{r_v}$ represent the average score of the user u and the user v, respectively. I_u Represents the rating item of user u, I_v represents the rating item of user v, $I_u \cap I_v$ represents the common score item of user u and user v.

3. Proposed DMLTW Method

In this section, we first propose a user-based time weight algorithm. Then, we propose a dynamic multi-level and time-weighted (DMLTW) collaborative filtering recommendation algorithm.

3.1 User-based time-weighted

Unlike the user-item time-weighted approach, the main idea of our user-based time-weighted algorithm is to allocate different similarity weights for different users when calculating similarity between users. Because the user is sensitive to time, we assign a time function f (t) to the user in the phase of similarity calculation. In other words, in our proposed algorithm, Eq.2 is modified to:

$$Sim(u,v) = \frac{\sum_{i\in I_u \cap I_v}(r_{u,i} - \bar{r_u})(r_{v,i} - \bar{r_v})\cdot f(t_{u,i},t_{v,i})}{\sqrt{\sum_{i\in I_u \cap I_v}(r_{u,i} - \bar{r_u})^2}\cdot\sqrt{\sum_{i\in I_u \cap I_v}(r_{v,i} - \bar{r_v})^2}} \tag{3}$$

Where $\bar{r_u}$ and $\bar{r_v}$ represent the average score of the user u and the user v, respectively. $I_u \cap I_v$ Represents the common rating item of user u and user v, $f(t_{u,i}, t_{v,i})$ represents the time weight.

In addition, we assume that the time function $f(t_{u,i}, t_{v,i})$ is a monotonically decreasing function that decreases as time t increases, and $f(t_{u,i}, t_{v,i}) \in (0, 1)$. That is, all the data contribute to the user's similarity calculation. Therefore, we choose to use exponential form to achieve our purpose. Exponential time functions are widely used in many algorithms to eliminate historical behavior over time. The time function is defined as follows:

$$f(t_{u,i}, t_{v,i}) = e^{-\lambda\cdot(t_{u,i} - t_{v,i})} \tag{4}$$

Where $t_{u,i}$ represents the rating time of user u for item i, $t_{v,i}$ represents the rating time of user v for item i, λ is used to adjust the weight size, we have conducted many experiments to determine the value of λ when the similarity is optimal. From Eq. 4 we can see that the range of time function is between (0, 1), the closer the two users' rating time, the greater the value of the time function.

3.2 Proposed DMLTW algorithm

In order to make user-based time-weighted algorithm has better recommendation effect, we consider introducing dynamic multi-level algorithm to adjust the similarity between users. Dynamic multi-level algorithm is based on the number of common rating items among users to adjust the similarity. We create multiple levels based on the total number of users and items available in the dataset, and according to the dataset characteristics and the actual situation, the dynamic multi-level collaborative filtering is improved, so as to achieve better recommendation effect.

As shown below, Eq.5 gives a formula for making positive and negative adjust of similarity.

$$Sim(u,v) = \begin{cases} Sim(u,v) + Sim(u,v), & if\ \mathrm{SR} \geq \mathrm{DvI} \\ Sim(u,v) + \frac{Sim(u,v)}{n}, & if\ SR < DvI - (n-2)*DvIU\ and \\ & SR \geq DvI - (n-1)*DvIU\ and \\ & n = 2,3,\dots,DvU-1 \\ Sim(u,v) * \left(\frac{\frac{1}{1+(Sim(u,v)*Sim(u,v))}}{DvU-1} - 1\right), & otherwise \end{cases} \tag{5}$$

Where SR represents the number of items of the same rating between the user u and user v, Eq.6 indicates the number of levels, Eq.7 and Eq.8 give the level to which the same rating item should be assigned. In addition, if the number is not an integer, it will be rounded to the nearest integer. When the number of the same rating items between user u and user v is greater than Devi, it is assigned to the first level, and the similarity is 2*Sims (ump). Next, according to the value of Duve, DvU-2 levels are assigned to dynamically adjust the similarity value, and the value of n is from 2 to DvU-1. Finally, if the number of the same rating items is not in previous DvU-1 level, it will be assigned to the last level. By Eq.5, the similarity value between user u and user v is adjusted either positively or negatively.

$$DvU = log_{10}(number\ of\ users\ in\ dataset) \tag{6}$$

$$DvI = log_{2}(number\ of\ items\ in\ dataset) \tag{7}$$

$$DvIU = \frac{DvI}{DvU} \tag{8}$$

Through the above mentioned, we give a new user-based time weight function, and on this basis present the improved dynamic multi-level algorithm, the new dynamic multi-level and time-weighted recommendation algorithm can calculate the similarity between users more accurately, thus having better recommendation quality.

4. Experimental evaluation

4.1 Experimental dataset

In order to verify the effect of the proposed algorithm on the actual dataset, we conduct experiments on the Movie Lens dataset. This dataset is collected by the Group Lens research team at the University of Minnesota on the Movie Lens site, which contains 671 users, 10004 rating and 163,949 items. After data processing, the number of items that are not graded by all users is removed, and get 671 users, 10004 rating and 9066 items, with the data sparseness is 99.83%. All rating values are from 0 to 5, and the higher the score, the stronger the user's interest.

4.2 Experimental evaluation metrics

Experiment using the mean absolute error (MAE) as the measure of the recommender system. MAE [14] is a widely used measure for judging the prediction accuracy of the recommender system, which indicates the deviation between the prediction rating and the actual rating. The deviation is smaller, the prediction accuracy is higher.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|p_i - r_i| \tag{9}$$

4.3 Experimental settings

In each experimental dataset, the training set and the test set is constructed according to the proportion of 80%-20% and 60%-40%, respectively. Through several experiments, we find the optimal value of the time weightλ, and all of the experiments are conducted under the optimum conditions ofλ. In the experiment, the traditional collaborative filtering (CF) recommendation algorithm and time-weighted CF recommendation algorithm are used as the reference methods to verify the effectiveness of the proposed algorithm.

4.4 Experimental results

Fig.1 and Fig.2 show the comparison of the MAE of the recommender algorithm in the dataset. Fig.1 use the 80% training set and 20% test set. Fig.3 use 60% training set and 40% test set. According to the experimental results, it is shown that our proposed algorithm can significantly improve the performance of the recommender systems. Compared with the reference algorithm, the proposed algorithm can achieve lower MAE values.

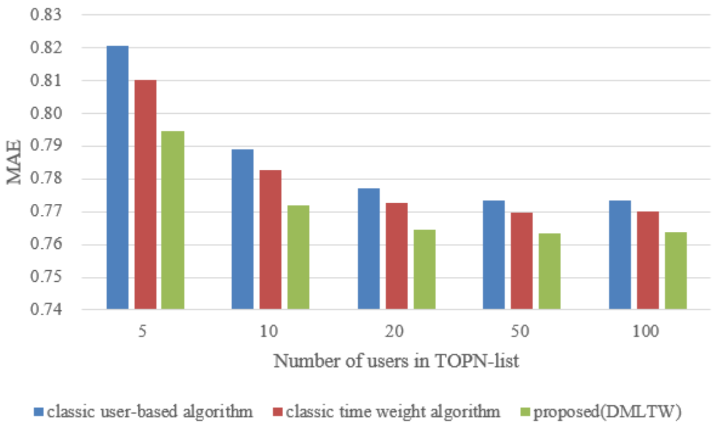

Figure 1.MAE for MovieLens with 80% training set

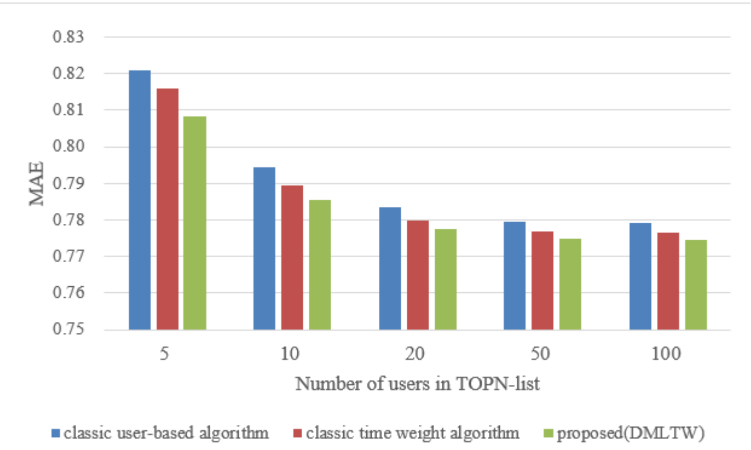

Figure 2.MAE for MovieLens with 60% training set

From Fig.1 and Fig.2, we can see that in the real dataset, changing the proportion of the training set and test set, our proposed DMLTW algorithm reduces the value of MAE. That is, compared with the reference algorithm, our proposed outperforms the alternatives.

5. Discussion and future work

We have proposed a DMLTW algorithm, and the experimental results show that our proposed algorithm has higher accuracy and better quality. In addition, compared with alternative recommendation methods, there are better recommendation results. Furthermore, online companies can use it to reduce server load and help users make decisions by providing more accurate recommendations in different domains.

In future, we will study the impact of DMLTW algorithm on the novelty of user recommendation, and will add more factors to provide personalized recommendation.

Acknowledgements

This work is supported by NSFC (Grant No.61502044), the Fundamental Research Funds for the Central Universities (Grant No.2015RC23).

References

[1]. G Linden, B Smith, and J York. (2003) Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7, 76-80.

[2]. J Davidson, B Liebald, et al. (2010) the youtube video recommendation system. Acm Conference on Recommender System, 293-296.

[3]. Y Koren. (2009) the bellkor solution to the netflix grand prize. *Netflix Prize Documentation*.

[4]. A Salah, N Rogovschi, and M Nadif. (2016) a dynamic collaborative filtering system via a weighted clustering approach. *Neurocomputing*, 175, 206-215.

[5]. H Sobhanam, and AK Mariappan. (2013) Addressing cold start problem in recommender systems using association rules and clustering technique. International Conference on Computer Communication & Informatics, 74, 1-5.

[6]. P Samatthiyadikun, and a Takasu, S Maneeroj. (2013) Bayesian model for a multicriteria recommender system with support vector regression. IEEE International Conference on Information Reuse & Integration, 38-45.

[7]. Y Fang, and Y Guo.(2013) A context-aware matrix factorization recommender algorithm. IEEE International Conference on Software Engineering & Service Science, 914-918.

[8]. Y Ding, and X Li. (2005) Time weight collaborative filtering. Acm Cikm International Conference on Information & Knowledge Management, 1, 485-492.

[9]. Y Koren. (2010) Collaborative filtering with temporal dynamics. ACM, 53, 89-97.

[10]. G Koutrika, B Bercovitz, and H Garcia-Molina.(2009) FlexRecs: expressing and combining flexible recommendations. Acm Sigmod International Conference on Management of Data, 745-758.

[11]. N Polatidis, and CK Georgiadis.(2016) A multi-level collaborative filtering method that improves recommendations. *Expert Systems with Applications*, 48, 100-110.

[12]. M Gan, and R Jiang.(2013) Improving accuracy and diversity of personalized recommendation through power law adjustments of user similarities. *Decision Support Systems*,55, 811-821.

[13]. G Adomavicius, and a Tuzhilin. (2005) Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge & Data Engineering*, 17, 734-749.

[14]. G Shani, and a Gunawardana.(2011) Evaluating Recommendation Systems. *Recommender Systems Handbook*, 257-297.