2012 Fifth International Conference on Business Intelligence and Financial Engineering

A Collaborative Filtering Algorithm Based on User Activity Level

Yongli Cui[1], Shubin Song[2], Liang He[3*], Guorong Li[4]
[1,2,3]Department of Computer Science and Technology, East China Normal University, Shanghai, China
[4]Shanghai KINGSWAY CO., LTD, Shanghai, China
{lhe@cs.ecnu.edu.cn}

***Abstract*—Collaborative Filtering Algorithm is one of the most successful recommender technologies, and has been widely used in E-commerce. However, traditional Collaborative Filtering often focus on user-item ratings, but ignore the information implicated in user activity which means how and how often a user makes operations in a system, so it misses some important information to improve the prediction quality. To solve this problem, we bring user activity factor into collaborative filtering and propose a new collaborative filtering algorithm based on user activity level (UACF) . Finally, experiments have shown that our new algorithm UACF improves the precision of traditional collaborative filtering.**

Keywords-user activity; collaborative filtering; recommender system

I. Introduction

With the rapid development of internet and e-commerce, information overload makes us have to spend much more time searching valuable information. Faced with vast amounts of goods, customers can't find right goods quickly. Therefore, many e-commerce systems such as Amazon, Ebay, etc. provide recommender systems to help customers choose the goods they might be interested in.

Collaborative Filtering (CF) is one of the most widely used algorithms in information filtering and information systems. Systems define the items or objects through the relevant properties and learn user interests based on the characteristics of users' rating, make recommendations depending on the relation between user data and items to be predicted, and try to recommend items that are similar to the items user has liked in the past. Unlike traditional content-based filtering which makes recommendations direct by analyzing the content, Collaborative Filtering analyzes user interest, finds users that are similar to the specified user in user group and combines the evaluation made by these users on a certain item. Finally the system predicts the rating of the specified user on the item.

Over the years, various approaches for improving collaborative filtering have been developed, such as Rating Distribution based, Time Period Partition based, etc. [2, 3, 4, 5, 7]. Traditional CF filters useless information by calculating item-item similarities or user-user similarities, and the direct factor in similarity calculating is user-item rating. Most existing algorithms also improve prediction quality based on the user-item rating [2, 3, 5], but few algorithms take user activity into account. User activity means users' participation in systems, it contains many factors, such as login times, operation times, percentage of a user's rating history on different types, operation habits in different periods, etc. The more active user participation in a system, the higher activity level is. User activity may reflect user interests to a certain extent and play an important role in finding similar user and rating prediction. Thus, we propose a collaborative filtering algorithm based on user activity level.

The organization of the rest of the paper is as follows. In section II, we review traditional CF algorithm briefly. Section III describes the UACF we proposed. Section IV presents our experiments. Section V makes a conclusion of our algorithm in the final.

II. Related Work

Our approach is based on traditional CF algorithm, so we review traditional CF algorithm briefly in this section. In general, the data set of recommender system consists of user set $U=\{u_1,u_2,...,u_m\}$, item set $T=\{t_1,t_2,...,t_n\}$ and the rating matrix $U\times T$, $R=(R_{u,k})$, where $R_{u,k}$ defines the rating of user u for item k. As shown in the table below, the rows define users and the columns define items.

	1	…	k	…	n
1					
⋮					
u			$R_{u,k}$		
⋮					
m					

A. Item-based Collaborative Filtering

We often think a person will more likely to choose the items that are similar to the items he has liked in the past. So Item-based CF need to find out the similar items first and compute the similarity between items. Then we select the most similar items to be an item's neighbor. The main similarity measures used in experiments generally are cosine similarity and adjusted cosine similarity.

Cosine similarity: Items are thought of as vectors in the m dimensional user space and the similarity between two items defines as cosine of the angle between the two vectors.

$$sim(i,j)=\cos(\vec{i},\vec{j})=\frac{\vec{i}.\vec{j}}{\|\vec{i}\|\|\vec{j}\|} \qquad (1)$$

[*] Corresponding author, Email: lhe@cs.ecnu.edu.cn

978-0-7695-4750-3/12 $26.00 © 2012 IEEE
DOI 10.1109/BIFE.2012.25

Adjusted cosine similarity: Because of the differences in rating scale between different users, basic cosine similarity may have drawback. Adjusted cosine similarity is proposed to offset this drawback by subtracting the corresponding user average from each co-rated pair.

$$sim(i,j)=\frac{\sum_{u\in U}(R_{u,i}-\bar{R}_u)(R_{u,j}-\bar{R}_u)}{\sqrt{\sum_{u\in U}(R_{u,i}-\bar{R}_u)^2}\sqrt{\sum_{u\in U}(R_{u,j}-\bar{R}_u)^2}} \qquad (2)$$

Where U is the set of users who rated both item i and j, $\bar{R}_u$ is the average of the user u's rating.

We use adjusted cosine similarity in our experiment because it has higher prediction quality [1]. After similarity calculating, we can find out similar items for each item. Then look into the target users ratings, compute the prediction by computing weighted sum of the ratings given by the user on these similar items of the target item.

$$P_{u,i}=\frac{\sum_{j\in T_i} sim(i,j)\times R_{u,j}}{\sum_{k\in T_i}\left|sim(i,j)\right|} \qquad (3)$$

Where $P_{u,i}$ is the prediction on item i for user u, $sim(i, j)$ is the similarity between item i and j, T_i is the set of similar items of item i, $R_{u,j}$ is the rating of user u for item j.

B. User-based Collaborative Filtering

Each user belongs to a group with similar interests. Consequently, items frequently liked by the members of the group can be used to generate recommendation to other members. User-based CF operates over the entire databases to identify a set of neighbors by computing user similarity. Once the neighbors are formed, systems predict based on the interests of target user's neighbor. Cosine and Pearson correlation is often used for computing user similarity.

Cosine similarity: Each user is a vector in the space of items, the similarity between two users defines as cosine of the angle between the two vectors.

$$sim(i,j)=\cos(\vec{i},\vec{j})=\frac{\vec{i}.\vec{j}}{\|\vec{i}\|\|\vec{j}\|} \qquad (4)$$

Pearson correlation:

$$sim(i,j)=\frac{\sum_{c\in I_{ij}}(R_{i,c}-\bar{R}_i)(R_{j,c}-\bar{R}_j)}{\sqrt{\sum_{c\in I_i}(R_{i,c}-\bar{R}_i)^2}\sqrt{\sum_{c\in I_j}(R_{j,c}-\bar{R}_j)^2}} \qquad (5)$$

Where $R_{i,c}$ is the rating of user i for item c, both user i and j have rated item c, $\overline{R_i}$ and $\overline{R_j}$ represent the average rating of user i and j respectively.

We adopt Pearson correlation in experiments since it achieves higher performance. After similarity computing, system form neighbors for each user and predict by performing a weighted average of deviation from the neighbors' mean.

$$P_{i,k}=\bar{R}_i+\frac{\sum_{j\in Neighbor} sim(i,j)\times(R_{j,k}-\bar{R}_j)}{\sum_{j\in Neighbor}\left|sim(i,j)\right|} \qquad (6)$$

Where $P_{i,k}$ is the prediction of user i for item k, user j is one neighbor of target user i, $sim(i, j)$ is the similarity between user i and j, $\overline{R_i}$ and $\overline{R_j}$ represent the average rating of user i and j respectively.

III. Collaborative Filtering Based on User Activity Level (UACF)

A. User Activity

We assume that users' interests can be reflected on their activities in systems, such as login times, the percentage of the operation on each type of items, the number of times operating on a certain type of items, the duration of the operation on each item, etc. We use Movielens Dataset in our experiments. According to the feature of the dataset, here we only consider two factors of user activity—rating times and the percentage of the rating on each type.

Assume such a situation as Figure 1 shown, there are three users' rating on six items of two types. If we use traditional method to calculate similarities between user A and B, user A and C, while *sim*(A, B) = 0.854, *sim*(A, C)=0.865. In comparison with user B, user C is more similar to user A. We can easily find that user A and B have seen action movies for 4 times, but user C has only seen action movie once and seen more romance movies. From the rating history we're not sure whether user C like action movie or not. Compared with user C, maybe user A and B prefer action movie. In this case, it is difficult to explain which one is more similar than another or which one is the accurate nearest neighbor.

	Action				Romance	
	Item1	Item2	Item3	Item4	Item5	Item6
User A	4	1	1	1	0	0
User B	4	3	2	4	0	0
User C	4	0	0	0	1	1

Figure 1

In this paper, we propose an approach in order to improve the accuracy of recommendation. For this purpose, we take rating times and the percentage of the rating on each type into consideration, so that the user activity weight is defined as:

$$\text{if } R(u,i)=0,\ Act(u,i)=\lambda\cdot\sqrt{\frac{T_u}{\bar{T}}}\cdot\sqrt{\frac{T_i}{T_{total}}}$$

$$\text{else } \ Act(u,i)=1$$

Where $Act(u,i)$ represents user activity weight of user u for item i. T_u is rating time of user u, $\bar{T}$ is average rating time of all users, T_i is the rating time on the type of item i (If item i belong to type action, T_i represents the rating time on type action of user u), T_{total} is the sum of rating times on all types of user u, λ is a parameter, we can adjust it in the experiments.

Adjusted similarity formula:

$$sim(i,j)=\frac{\sum_{c\in I_{ij}} Act(i,c)\cdot(R_{i,c}-\bar{R}_i)\times Act(j,c)\cdot(R_{j,c}-\bar{R}_j)}{Act(i,c)\cdot\sqrt{\sum_{c\in I_i}(R_{i,c}-\bar{R}_i)^2}\times Act(j,c)\cdot\sqrt{\sum_{c\in I_j}(R_{j,c}-\bar{R}_j)^2}} \quad (7)$$

Adjusted prediction formula:

$$P_{i,k}=\bar{R}_i+\frac{\sum_{j\in Neighbor} sim(i,j)\times(R_{j,k}-\bar{R}_j)\cdot Act(j,k)}{\sum_{j\in Neighbor}\left|sim(i,j)\right|} \quad (8)$$

B. *Sparsity in CF*

Sparsity is an unavoidable problem in all recommender systems. There are many vacant ratings in real data set, this will make low prediction quality. In respect to an item which is rated by only few users, computing prediction become difficult, so it can hardly be recommended to users. In respect of a user who only rates few items, systems can't find his interests correctly and the prediction is inaccurate.

Many approaches have been proposed to alleviate sparsity, such as Combining Memory-Based CF, dimensionality reduction technique, etc. [2, 6]. We adopt Memory-Based CF in our experiments. We first use formula (1) and (3) to compute the prediction of all the empty rating in the original matrix, and then we generate a new matrix R' which is filled with these predicted rating. After alleviating sparsity, we use formula (7) and (8) to do subsequent experiments on the new matrix.

IV. Experiment

A. *Data Set*

We use MovieLens 100K dataset in our experiments. This dataset consists of three subsets. Rating sets contains user, item and rating information, there are 100,000 ratings of 943 users on 1682 items. Each user has rated 20 times at least. User sets contains all user id and user attributes. Item sets contains all item id and item attributes, including the type information of each item. Each item belongs to several types such as action, comedy, crime, etc. There are 19 types in total. The datasets used were divided into 80% of the training set and 20% of the test set respectively.

B. *Metrics*

We use the Mean Absolute Error (MAE) metrics to measure the predict quality of our proposed approach and compare our UACF algorithm with traditional collaborative filtering algorithms.

MAE is defined as:

$$MAE=\frac{\sum_{i\in test}\left|P_{u,i}-R_{u,i}\right|}{n} \quad (9)$$

Where $R_{u,i}$ denotes the rating that user u give to item i, $P_{u,i}$ denotes the rating that user u gave to item i which is predicted by our approach, n is the number of test ratings.

C. *Comparison*

In order to verify the effectiveness of our approach, we compare our UACF algorithm with traditional CF algorithms. We use (5) and (6) compute prediction in traditional CF. We adjusted parameter λ in our experiments at first, figure 2 shows that the result is optimal when λ is 2.0.

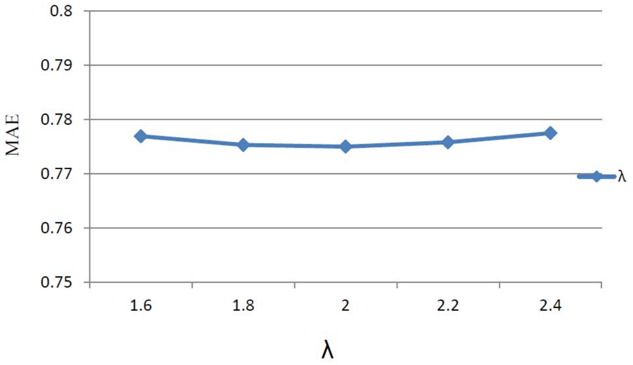

Figure 2

The number of neighbor increases from 30 to 80 in our experiments, the result is shown in figure 3. We can find that the prediction is more accurate after alleviating sparsity, and our UACF performs best in experiments. The difference between UACF and traditional CF is that user activity is considered in user similarity calculating and prediction in UACF. And the result shows our UACF is more accurate than traditional CF.

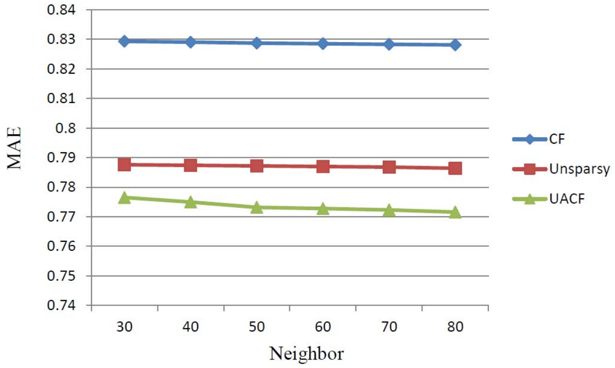

Figure 3

V. Conclusions and Future Work

In this paper, we propose an effective collaborative filtering algorithm based on user activity level, our approach takes advantage of user activity which represents user behavior and habit and implies user preference. And our approach is easy to be integrated with existing system, it only uses the type information. Experiments results show the UACF is more accurate than traditional CF. We only consider two factors in user activity, so in the future work, we can take more factors of user activity into account. We can classify the users based on user activity, and then choose the most appropriate recommendation methods for corresponding users.

Acknowledgment

This work is supported by a grant from the National High Technology Research and Development Program of China

(863 Program) (No.2011AA01A107) and the Shanghai Science and Technology commission Foundation (No. 10dz1500103 and 10dz150010B and 11511502203).

REFERENCES

[1] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” Proceedings of the 10th international conference on World Wide Web, ACM, 2001, pp. 285–295.

[2] R. Hu and Y. Lu, “A hybrid user and item-based collaborative filtering with smoothing on sparse data,” Proceedings of the 16th International Conference on Artificial Reality and Telexistence, IEEE Computer Society Press, 2006, pp. 184−189.

[3] S. Gong, H. Wu Ye, and H. S. Tan, “Combining memory-based and model-based collaborative filtering in recommender system,” Recommender System. Pacific-Asia Conference on Circuits, Communications and System, 2009, pp. 690−693.

[4] S. Deng, L. He, and W. Xia, “A collaborative filtering algorithm based on rating distribution,” Proceedings of 2008 IEEE International Symposium on IT in Medicine and Education, 2008, pp. 1118 – 1122.

[5] Y. Zhang and Y. Liu, “A collaborative filtering algorithm based on time period partition,” Third International Symposium on Intelligent Information Technology and Security Informatics, 2010, pp. 777 – 780.

[6] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Application of dimensionality reduction in recommender system-a case study,” Proc. of the ACM WebKDD Workshop, 2000.

[7] G.R. Xue, C. Lin, Q. Yang, W. Xi, H.J. Zeng, Y. Yu, and Z. Chen, “Scalable collaborative filtering using cluster-based smoothing,” Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2005, pp. 114–121.