An Item-based Collaborative Filtering Method using Item-based Hybrid Similarity

Sutheera Puntheeranurak, Thanut Chaiwitooanukool
Faculty of Engineering,
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
kpsuthee@kmitl.ac.th, Orange99.ng@gmail.com

Abstract—**Item-based collaborative filtering is a preferred technique on recommender system. It uses a value of item rating similarity to predict user's preference. In this paper, we include values of item attribute similarity to adjust the predicted rating equation for target item. The results of Item-based collaborative filtering that hybrid item rating similarity and item attribute similarity techniques have Mean Absolute Error (MAE) less than a traditional Item-based collaborative filtering technique and others. The proposed algorithm is efficient to predict better than traditional algorithm as shown in our experiments.**

Keywords-recommendation system; item –based collaborative filterin; collaborative filtering

I. Introduction

Nowadays, The Internet is important of life and collects of information for system user, with the cause of community aggregation that is very important. E-commerce is a business offering their products to the user to make a purchase such as food, movies, and books, etc. There are many different methods to help referrals. Recommendation systems [2] are one of the technologies that have become to help to introduce products for meeting customer needs. There are many E-commerce systems such as Amazon, CDnow, Netflix, MovieLens, etc that use different recommendation algorithms but all of systems will offer the recommended product to the user when the user accesses the system. There are two techniques, content-based filtering and collaborative filtering that commonly used in the recommendation systems. Method of content-based filtering will analyze the content of each item that user used to provide rating. The system will recommend the item that has content match to the user's profile. The problem of this technique is the result of the recommendation will dependent on only the user's profile. Therefore the item that does not match to the profile of users is rarely selected to recommend to the user even if the item may be interesting. To solve this problem, the collaborative filtering techniques are developed and widely used now.

The collaborative filtering [3] is early implemented by Xerox Parc Institute for research, called the "Tapestry". There are two techniques, user-based collaborative filtering [4] and item-based collaborative filtering. The user-based collaborative filtering [5] has brought ratings of items that are provided by the user to compare with ratings of other users on same items. The systems will make automatic predictions about the interests of a user by collecting preferences or taste information from many users and calculating the similarity between users. However, this technique still has some problems about the amount of information in the system therefore it takes long time to process, called scalability problem. Then the Item-based collaborative filtering is proposed to solve the problem of user-based collaborative filtering. This technique resemble process step of user-based collaborative filtering but the rating of item is determined by users who used to provide it in the past. And the technique uses a similarity between items for selecting neighbor items to predict user preference on target item. The results of Item-based collaborative filtering have more effectively in predicting than user-based collaborative filtering and also commonly implement for recommendation systems. However, these techniques still have the problems as the rating of items on system is not covered, called sparsity problems. Another problem is the new item that has not been the rating score, called first-rater problem. When rating is sparse or less, it will effect to find neighbor item and the prediction rating performance is decreased.

In this paper, we propose the innovation algorithm to improve the efficiency for item-based collaborative filtering technique. Our methods will use properties of item to calculate similar values. We called item-attribute similarity, and then we use it to combine with item similarity values and predict the recommended item to the target user. The results in our prediction are more accurate after we find the group of neighbor items by checking similarity attributes and add it into the prediction equation. Therefore it shows that the item-attribute similarity, the attributes of item can indicate or define information of items, can use to improve the efficiency as shown in our experiments. The result can show that our proposed algorithm for recommendation system is more effective.

II. RELATED WORKS AND BACKGROUND

The Item-based collaborative filtering technique has various algorithms and many researchers are interested to make it much more quality. The item clustering is researched by YiBo Huang[6]. They improved item-based collaborative filtering by using k-mean clustering algorithm[7] to make item clustering for prediction and use the grouping item to find similarity of items and able to predict for target item. HengSong Tan [8] classified the items using the item attribute content to find items similarity on each item and categorized or classified items to prediction rating. Ye Zhang [9] used genre of movie dataset is imported from MovieLens dataset. They

978-1-4244-9698-3/11/$26.00 ©2011 IEEE

proposed that genre of the movie can help to find movie similarity on each movie. They implemented method to predict the rating by user-based collaborative filtering and item-based collaborative filtering separately. After that they compared the result of similarity of each movie on target item and chose the good result to present to the user. Their result increases accuracy but it still discovers the sparsity problem. SongJie Gong and HongWu Ye[10] finds neighbor items that used item rating similarity and item attribute similarity. They use linear combination to combine both similarities for prediction rating to target user. Then they have to choose the optimized weighting value between the item rating similarity and the item-attribute similarity. However, in real-world application, it is very difficult to get the optimized weighting value. Therefore, we proposed the algorithm that hybrid between the item rating similarity and the item attribute similarity in the prediction equation.

TABLE I. ITEM ATTRIBUTE MATRIX

Attribute Item	A_1	A_2	A_3	A_4	A_5
Item_1	1	0	1	1	0
Item_2	0	1	0	0	1
Item_3	1	0	1	1	1
Item_4	0	1	1	1	1

In our proposed algorithm, we create relationship table between attributes of item as TABLE I. We classified each item that have the same content attributes. For example, item1 has a value '1' in column A1, A3 and A4. It means the item1 has related to attributes A1, A3 and A4. And if item1 that has not related to attributes A2 and A5, then we filled the values '0' in column of attributes A2 and A5. After we classified all items, we will get the result as TABLE I. We will use all data in TABLE I to calculate a similarity of each item using properties of attribute items to predict preference rating to target user.

III. METHOD AND COMPUTATION

A. Item Rating Similarity Method

TABLE II. RATIING MATRIX

Item User	Item_1	Item_2	Item_3	Item_4
User_1	3	2	5	4
User_2	?	5	1	?
User_3	2	5	??	3
User_4	2	1	3	2
User_5	2	?	5	5

We calculate the item rating similarity by comparing the rating of target item to rating of all items. We find a group of items that have similar rating or most similar base on preference rating. We called the group of items as co-rated as shown in TABLE II. This method will take only co-rated rating to compute to find the similarity that matches between two items. In this paper, we use cosine correlation method [4] to find similarity of items as (1).

$$RSim(t,c) = \frac{\sum_{u\epsilon U} R_{u,t} * R_{u,c}}{\sqrt{\sum_{u\epsilon U} R_{u,t}{}^2} * \sqrt{\sum_{u\epsilon U} R_{u,c}{}^2}} \qquad (1)$$

Where *RSim(t,c)* denotes the similarity between item *t* and *c* ; $R_{u,t}$, $R_{u,c}$ mean the rating of user *u* on item *t* and *c* respectively and *U* is the set of users who rated on both item *t* and *c*. The circle in TABLE II represents co-rated rating.

B. Item Attributes Similarity Method

This method is used to find the items similarity by using attributes items. At First, we bring information of all items in the system to compare with each item. The item that has a same category or a same attribute will be used to create Item attribute matrix. In our proposed algorithm, we adjust Hamming distance technique [11] to apply for comparing each item and finding similarity by attribute of items. We give examples item3 and item4 to compare to find similar as follows.

Item_3 [0 1 1 1 1]
Item_4 [1 0 1 1 1]

Typically Hamming Distance method calculate from a number of different attribute of items that is compared in each position respectively between Item3 and Item4 then the result is equal to 2. But we use a number of same attribute of items instead. Because we want to indicate the similarity of items and show that two items are closely and have same attribute of items. We proposed (2) as.

$$Sim(A_t, A_c) = \frac{\sum(A_t = A_c)}{A_n} \qquad (2)$$

Where *$Sim(A_t,A_c)$* denotes the attribute similarity between item *t* and *c*. A_t, A_c mean the value of attributes *A* on item *t* and *c* respectively, and A_n is number of all attributes.

In this example, when we count the pair of same items attribute that its result equals 3, and we do the normalization by divide by total number of all attributes. Then we can create relationship table of all items as following.

TABLE III. THE SIMILARITY OF ITEMS BY ATTRIBUTES

Attribute Item	Item_1	Item_2	Item_3	Item_4
Item_1	1	0	0.8	0.4
Item_2	0	1	0.2	0.6
Item_3	0.8	0.2	1	0.6
Item_4	0.4	0.6	0.6	1

In TABLE III, we show that the results is equal to 1 if the item is compared by itself. And if we compared with other items, the attribute of items are not exactly. Then the similarity of items is less than 1. In our proposed method, we define *Sim($A_{t,k}$)* that means the similarity of item attribute between item *t* and other items in system.

C. Prediction by using Item-based Hybrid Similarity

In our research, we adjust the traditional method to predict rating as equation:

$$P_{u,t} = \overline{R_t} + \frac{\sum_{k=1}^{n} RSim(t,k) * Sim(A_{t,k}) * \left(R_{u,k} - \overline{R_k}\right)}{\sum_{k=1}^{n} |RSim(t,k) * Sim(IA)|} \quad (3)$$

Where $P_{u,t}$ means the prediction for the user *u* on item *t* based on item-based collaborative filtering; *RSim(t,k)* means the item similarity of co-rated between item *t* and *k* ; *Sim($A_{t,k}$)* denotes the item-attribute similarity between item *t* and *k* ; $R_{u,k}$ is the user rating on item *k;* $\overline{R_t}, \overline{R_k}$ is the average rating on item *t* and *k* respectively, *n* is the total of neighbors of item *t*;

From (3), we use item similarity *RSim(t,k)* and item-attribute similarity *Sim($A_{t,k}$)* to improve the accuracy of the prediction. The result is preference rating of user that provides on target items and is able to recommends item to users.

IV. EXPERIMENTAL AND RESULTS

Dataset that used in the experiments is from the MovieLens website. This site keeps data to use for recommending a movie with voted movie ratings by member users. MovieLens's datasets were collected by the GroupLens Research Project at The University of Minnesota. The historical dataset consists of 100,000 ratings from 943 users on 1682 movies with every user having at least 20 ratings. Ratings follow 1 to 5 numerical scales. The number of valuable means users like that movie at most. There are 19 genres as unknown |Action |Adventure |Animation |Children's |Comedy |Crime |Documentary |Drama |Fantasy |Film-Noir |Horror |Musical |Mystery |Romance |Sci-Fi |Thriller |War |Western. Then we use this information to create an Item attribute matrix. Each movie has 19 genres then a matrix has 19 columns. After that we compute item-attribute similarity. .

We evaluate our algorithm by comparing an accuracy of prediction. We use the Mean Absolute Error (*MAE*) [13]; The *MAE* returns the error result of prediction rating between real rating or hiding rating (testing data) and new rating of prediction. The *MAE* is represented as follow.

$$MAE = \frac{\sum_{i=1}^{n} |p_i - q_i|}{n} \quad (4)$$

Where, p_i is a predicted rating, q_i is the actual ratings and *n* is the number of actual ratings in the testing data. The lower of *MAE* show that the more accurate the predictions are permitting to provide better recommendations.

A. Experiment 1

We test the performance by adjusting the value of *k* to have the number of items that are neighbors 2, 5, 10, 15, 20, 25, 30, 35, and 40 respectively. Then we choose movies that have similar to target movie. The results are shown in Fig. 1.

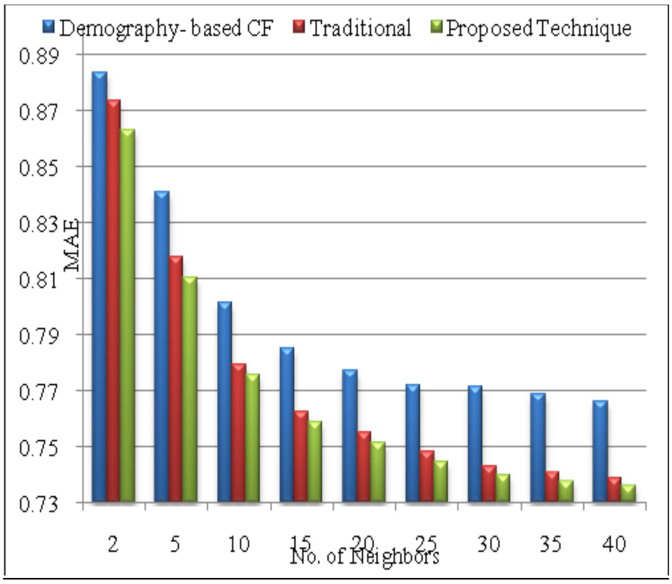


Figure 1. The results of mean absolute error (*MAE*) by adjusting the value of k-neighbors

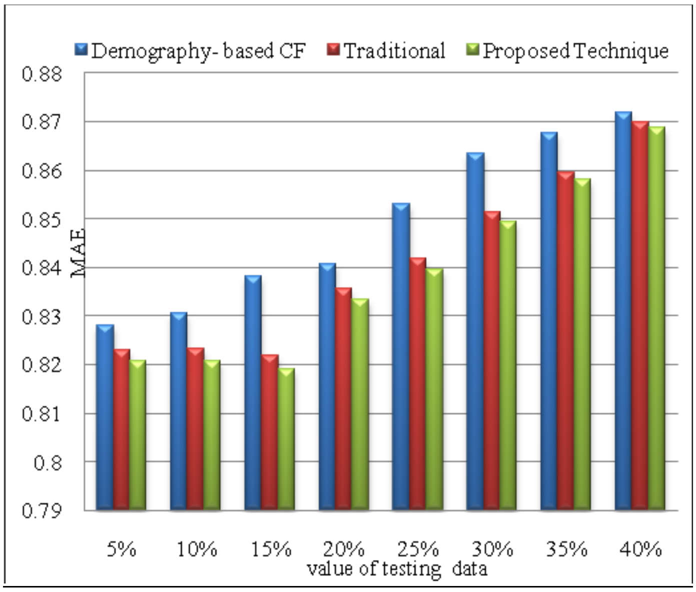


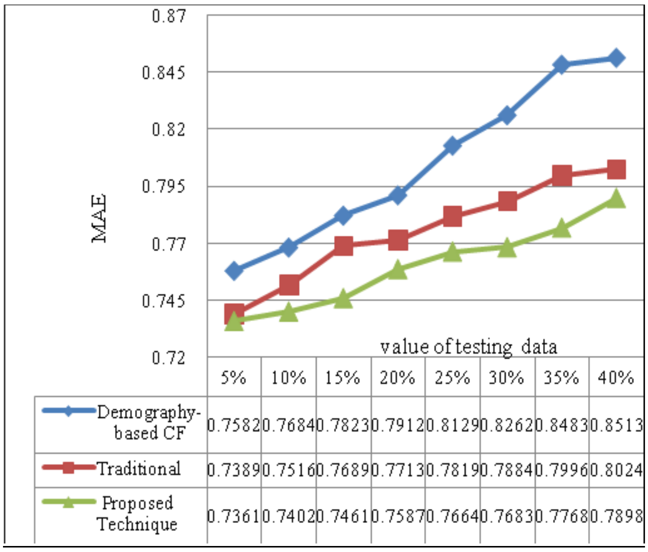
Figure 2. The results of mean absolute error (*MAE*) by adjusting the value of testing data

In Fig. 1, we evaluate by increasing *k* value, size of neighbors items, is used for prediction. Our experiments compare Traditional algorithm, Demography-based collaborative filtering [14], with our proposed algorithm. Demography-based CF method includes demography of users to find similarity that increases accuracy for prediction. We observe the results of evaluation. It can show that if we increase k neighbor values, the *MAE* will be decreased and the result from our proposed algorithm has performed better than other algorithms at all *k* neighbors.

B. *Experiment 2*

We test the performance by increasing a number of randomly in the testing data 5%, 10 %, 15%, 20%, 25%, 30%, 35% and40 % respectively, and defining of *k* neighbors is equal 5. It means there are 5 movies that take on a neighborhood movie. The results are illustrated in Fig. 2. We can show that when we increased percent of testing data, the MAE of predictions is increased because the ratings of each item are sparse. It affects to compute selecting items for the neighbors and find the values of similarity neighbor items, so the results of the prediction preference rating are decreased. Our proposed algorithm makes more accuracy than all.

The results of the two experiments can be used to try to compare three algorithms are shown in Fig. 3. We determine the number of neighbor item as 40. This number of neighbor movies has the lowest *MAE* as shown in experiment 1. And we calculated by adjusting the percent of testing data respectively. All algorithms have the same trend of the results of *MAE* as when the numbers of testing data are small, the MAE is low. In all experiments have made the confident that our proposed algorithm offers more accuracy of prediction than other algorithms.



	5%	10%	15%	20%	25%	30%	35%	40%
◆ Demography-based CF	0.7582	0.7684	0.7823	0.7912	0.8129	0.8262	0.8483	0.8513
■ Traditional	0.7389	0.7516	0.7689	0.7713	0.7819	0.7884	0.7996	0.8024
▲ Proposed Technique	0.7361	0.7402	0.7461	0.7587	0.7664	0.7683	0.7768	0.7898

Figure 3. The average of *MAE*

V. CONCLUSIONS

Item-based collaborative filtering finds similar items to target item and predict preference rating. In generally, it finds the group of items that are closed by using a rating data of each item. In our research, we proposed the algorithm that includes attribute of item to compute similarity of each items, in order to be effective on prediction preference rating more accurately in experiments when we take the similar values obtained genre of movie to compute. When we predict the preference rating to target items, the result of MAE is less than others. This result shows that our proposed algorithm is more accurate when we compare with Traditional and Demography-based Collaborative Filtering algorithms and our proposed algorithm can be implemented to the current recommender systems.

REFERENCES

[1] H.J. Scholl, K. Barzilai-Nahon, J.H. Ahn, O.H. Popova, and Barbara, “E-Commerce and e-Government: How Do They Compare? What Can They Learn From Each Other?," Proceedings of the 42nd Hawaii International Conference on System Sciences, pp 1-10, 2009.

[2] J. Ji, Z. Sha, C. Liu,N. Zhong, "Online Recommendation Based on Customer Shopping Model in E-Commerce," Proceedings of the IEEE/WIC International Conference on Web Intelligence, pp 68-74, 2003.

[3] D.B. Terry, "Replication in an Information Filtering System," pp 66-67, 1992.

[4] J. Herlocker, “Understanding and Improving Automated Collaborative Filtering Systems,” Ph.D. Thesis, Computer Science Dept., University of Minnesota, pp 34-52, 2000.

[5] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, “Item-Based collaborative filtering recommendation algorithms,” Proceedings of the 10th International World Wide Web Conference, pp285-295 2001.

[6] Y. Huang, "An Item Based Collaborative Filtering Using Item Clustering Prediction," ISECS International Colloquium on Computing, Communication, Control, and Management, pp54-56, 2009.

[7] J. Wu, W. Yu,"Optimization and improvement based on K-Means Cluster algorithm," Second International Symposium on Knowledge Acquisition and Modeling , pp 335-339, 2009.

[8] H. Tan, H. Ye, "A Collaborative Filtering Recommendation Algorithm Based on Item Classification," Pacific-Asia Conference on Circuits,Communications and System, pp 694-697, 2009.

[9] Y. Zhang ,W. Song, "A Collaborative Filtering Recommendation Algorithm Based on Item Genre and Rating Similarity," International Conference on Computational Intelligence and Natural Computing, pp 72-75, 2009.

[10] S. Gong ,H. Ye, X. Shi, "A Collaborative Recommender Combining Item Rating Similarity and Item Attribute Similarity," 2008 International Seminar on Business and Information Management, pp 58-60, 2008.

[11] F. Pappalardo, M. Pennisi, S. Motta, C. Calonaci, E. Mastriani, "Fast Hamming Distance Computation," World Congress on Computer Science and Information Engineering, pp 569 – 572, 2009.

[12] P. Resnick, N. Iacouvou, N. Suchak, P. Bergstrom, and J. Riedl, “GroupLens: an poen architecture for Collaborative filtering of Netnews,” Proc. of ACM CSCW’94, pp.175-186, ChapelHill, NC, 1994.

[13] T. Hofmann, “Collaborative Filtering via Gaussian Probabilistic Latent Semantic Analysis, ” Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 259-266, 2003.

[14] Y. Dai, H. Ye,S. Gong, “Personalized Recommendation Algorithm Using User Demography Information,” 2009 Second International Workshop on Knowledge Discovery and Data Mining, pp 694-697, 2009.