Recommendation Algorithm combining the User-Based Classified Regression and the Item-Based Filtering

YU Chuan
Department of Applied Mathematics
Renmin University, Beijing, China
13810473747, 86
comtedeartois@yahoo.com

XU Jieping
Department of Computer Science
Renmin University, Beijing, China
62511256,010, 86
Jieping.xu@263.net

DU Xiaoyong
Department of Computer Science
Renmin University, Beijing, China
13501130431, 86
Duyong@ruc.edu.cn

ABSTRACT
With the expansion of the Internet services, providing personalized product recommendations has become one of the most important ways to attract customers. Especially, collaborative recommender systems have achieved widespread success on the web. Information of products is recommended to the users based on their nearest "neighbors" who have similar interests. It is widely known that there is a sparsity problem in such systems. However, according to our research, there are other problems: one is that the typical collaborative algorithm loses some important parameter when it predicts the ratings, because there might be a strong similarity between the users who give very different ratings. Another is that the classification information of resources is not used. To solve these problems, we have proposed a recommendation algorithm combining the user-based classified regression and the item-based filtering. The experiment results show that performance is improved after applying the new algorithm.

Categories and Subject Descriptors
H.3.3 [Information Systems Applications]: Information Search and Retrieval-Information filtering; I.2.6 [Computing Methodologies]: Learning-Parameter learning; J.1 [Computer Applications]: Administrative Data Processing-Business

General Terms
Algorithms, Performance, Economics, Experimentation.

Keywords
Classified regression, Collaborative recommendation, Item-based filtering, Recommender systems

1. INTRODUCTION
People are becoming increasingly dependent on the Internet to get a variety of information for there continues a vast expansion of Internet services. Consequently, how to efficiently help people obtain information that they truly need in a world of overwhelming information on the Internet is a challenging task. Being an effective tool to address this problem, the recommender system has caught increasing attention from researchers and has become an essential research program in Internet application systems such as E-commerce system (e.g., Amazon, CDNOW, eBay) and digital library system. At present, the recommender systems can be categorized as contend-based [5] or collaborative.

Collaborative filtering [7] is a technology most frequently studied and applied in current recommender systems where it can solve the problem of a contend-based algorithm. Collaborative filtering is different from the content-based algorithm in that it recommends information resource by comparing the values of the similarities among the users or resource items instead of matching customer interest profiles with the product attributes. Its advantage lies in its ability to provide new information that would arouse the user's interest. Many different techniques have been proposed for collaborative recommendation, including the most original methods [7], Bayesian learning [4], latent semantic indexing [3]and clustering.

As the contents of resources increase, the sparsity [8] becomes a serious problem. Generally speaking, the amount of information resources is immense, products with users' comments account for a small proportion of the total, which leads to a sparse user-item rating matrix. Under this circumstance, it is very difficult to accurately find similar users. To solve this problem, some people proposed a method of combination to reduce the negative effect of the sparse matrix [1]. However, there remains another problem in the user-based filtering: the value of the similarity might be high between the users who give very different ratings. This problem can not be solved by the combination.

Moreover, typical collaborative filtering does not use the classification information of the resources. The similarities of the users for partial information items have been obtained by using the item-based similarity to reduce the range of resources in [9], and the performance can be enhanced, but in the process it has lost the information about the similarity among all items.

To solve these problems, in this paper a recommendation algorithm combining the user-based classified regression and the item-based filtering is proposed. The experimental results show that performance has been improved.

2. RELATED WORK
2.1 Typical User-Based Filtering
A traditional collaborative recommender system first searches for a set of similar users who are named "similar users" for their similar interests, usually given the name "neighbors", whose past ratings had the strongest correlations, then produces a prediction or a top-N recommendation based on a combination of the ratings of nearest neighbors for the active user [6].

ICEC'06, August 14–16, 2006, Fredericton, Canada.

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u \in U} sim(a,u) \times (r_{u,i} - \bar{r}_u)}{\sum_{u \in U} | sim(a,u) |}$$

Here $p_{a,i}$ denotes the predicted rating of user a on item i, $r_{u,i}$ denotes the rating of user u on item i, $\bar{r}_a$ is the average rating of user a, U denotes the set of similar users

2.2 User-Based Similarity Computation

In order to identify similar users, we should measure the similarity between two users. There are two most popular methods: cosine-based similarity and correlation-based similarity.

Correlation-based similarity: I denotes the set of items which are rated by both the active user a and user u. Then the correlation similarity between a and u is given by [7]:

$$sim(a,u) = \frac{\sum_{i \in I} (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i \in I} (r_{a,i} - \bar{r}_a)^2} \sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2}}$$

Here $r_{a,i}$ denotes the rating of user a on item i, $\bar{r}_a$ is the average rating of user a, and all summations over i are over the items that have been rated by both a and u.

Cosine-based similarity: In this case, two users are thought of as two vectors in the item-space. The similarity between these two users is measured by computing the cosine of the angle between them. Formally, the similarity between users a and u, denoted by $sim(a, u)$ is given by:

$$sim(a, u) = \frac{\sum_{i \in I} r_{a,i} r_{u,i}}{\sqrt{\sum_{i \in I} r_{a,i}^2} \sqrt{\sum_{i \in I} r_{u,i}^2}}$$

Most recommender systems apply the correlation similarity since it outperforms the cosine-based similarity in most situations [1].

2.3 Problems of Typical User-Based Collaborative Filtering Algorithm

2.3.1 Problem1:

Cosine-based similarity is equal to the cosine of the angle between two vectors. When the angle between vector X and vector Y is zero, there exists a relationship between X and Y that can be denoted by such a simple formula: $Y = \beta X$. Thus when the cosine-based similarity is close to 1, in another word: when the angle between X and Y is close to 0: $Y \approx \beta X$ .There is a positive relationship between the fitting degree and the cosine-based similarity.

When it comes to the user-based collaborative filtering, we will discover the correlation-based similarity is rightly the cosine-based similarity between $(R_a - \bar{R}_a)$ and $(R_u - \bar{R}_u)$.

$$R_a = (r_{a,1} ... r_{a,n})^T \ , \bar{R}_a = (\bar{r}_a ... \bar{r}_a)^T$$

So when the value of the similarity between these two users is high, $R_a - \bar{R}_a \approx \beta (R_u - \bar{R}_u)$ or $r_{a,i} \approx \bar{r}_a + \beta (r_{u,i} - \bar{r}_u)$. It is clear that if $(R_a - \bar{R}_a)$ and $(R_u - \bar{R}_u)$ are very different when the value of the similarity is high, the value of β cannot approximates 1, the prediction should be given by such a formula after weights are included:

$$p_{a,i} = \frac{\sum_{u \in U} sim(a,u) \times [\bar{r}_a + \beta (r_{u,i} - \bar{r}_u)]}{\sum_{u \in U} | sim(a,u) |} .$$

Thus if we make prediction via the traditional formula for this case, then we will lose the information of β .

2.3.2 Problem2:

In many recommender systems, information resources usually can be classified into different sets of information items according to their contents. For example, if the information resources are all movies, then each set might denote a film genre. The existing way of computing the similarity only computes the similarity between users among all items. However, if the interests of two users are not similar among all items, but they do take a similar interest in one of the genres, and the item on which we have to predict its rating is just in the genre. In this case, computing the similarity among all items of an inappropriate categorical level will have a bad performance.

Table 1: A simple example of the situation described above.

	Genre1		Cenre2				Genre3			Genre4		
	Item	Item…	Item i						…			…
a	**1**	**5**	**?**	**2**	**4**			**1**	**4**		**2**	**5**
u	**4**	**2**	**1**	**2**	**4**			**5**	**1**		**5**	**1**

Table 1 is an example. The value of the similarity (correlation-based) of user a and user u is very low among all items for the value=—0.7, but the value of the similarity in genre2 which the target item i belongs to is very high for the value=1.

2.3.3 Problem3:

The user-based collaborative filtering has been known to be problematic when the available ratings are sparse. In such a situation, it is very difficult to find any similar users accurately, which would result in the reduction of accuracy of the recommendation engine in general.

3. The Recommendation Algorithm Combining the User-Based Classified Regression and the Item-Based Filtering

3.1 User-Based Regression:

3.1.1 Typical Regression Model:

$Y = \alpha + \beta X + \varepsilon$, here Y and X denote two vectors, $a = \hat{\alpha}$ and $b = \hat{\beta}$, the parameters of which can be figured out with Least Square Equation method (LSE).Thus $Y = a + bX + e$, $a = \bar{Y} - b\bar{X}$ and $Y - \bar{Y} = b(X - \bar{X}) + e$. When R^2 (the coefficient of determination) is higher, the residual error is smaller and the regression level of the regression model is higher. There is a positive relationship between R^2 and $|r|$ since $r^2 = R^2$. Therefore, when the value of $|r|$ is high, we can deduce the formula:

$$(Y - \bar{Y}) \approx b(X - \bar{X}) .$$

3.1.2 User-Based Regression Algorithm:

When it comes to the user-based recommendation algorithm, if we define that $R_a = Y$ and $R_u = X$, then when the absolute value of r is high, we may estimate that $R_a - \bar{R}_a \approx b(R_u - \bar{R}_u)$; where

$$|r| = \left| \frac{\sum_{i \in I}(r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i \in I}(r_{a,i} - \bar{r}_a)^2}\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r}_u)^2}} \right| = |sim(a,u)|$$

. The value of b is figured out with LSE method and regarded as the estimation value of β in Section 2.3.1. We further conclude:

$r_{a,1} \approx \bar{r}_a - b\bar{r}_u + br_{u,1} \ldots r_{a,n} \approx \bar{r}_a - b\bar{r}_u + br_{u,n}$. Thus if we wish to predict the rating of user a on item i, we may apply this formula: $p_{a,i} = \bar{r}_a - b\bar{r}_u + br_{u,i}$. Notes: this formula is applied in a set of neighbors that only includes one similar user. For a set which includes more than one similar user, we compute the summation with weights, and the prediction is given by :

$$p_{a,i} = \frac{\sum_{u \in U} W_{a,u}(a + br_{u,i})}{\sum_{u \in U} W_{a,u}}$$

. Here U denotes the set of similar users whose weights are higher than the value of correlation threshold T_1. Furthermore, since there is a positive relationship between the regression level of the regression model and R^2, the regression model is efficient only at the time that R^2 is high. Therefore, we define another threshold T_2. If $|r| = |sim(a,u)| >= T_2$, b is figured out with the LSE method $a = \bar{r}_a - b\bar{r}_u$ $W_{a,u} = r^2 = R^2$; otherwise when $|r| = |sim(a,u)| < T_2$, as typical formula does, $a = \bar{r}_a - \bar{r}_u$, $b = 1$, $W_{a,u} = r$, where $\bar{r}_a$ is the average rating of user a on all items he has rated.

Note: Normally, to determine the applicability of a regression model, besides an observation of the coefficient of determination, an F-test is also necessary. However, for our approach, the efficiency of the algorithm will decrease rapidly once we apply an F-test to every regression model of two users since too much storage space and time would be required to complete the additional computing. Thus an F-test is not taken.

3.2 User-Based Classified Regression Algorithm:

First the items are classified based on their contents:

$$I = S_1 \cup S_2 \cup \ldots \cup S_n ; S_1 = \{i_{11} \ldots i_{1k_1}\}, \ldots, S_n = \{i_{n1} \ldots i_{nk_n}\}$$

$$|S_1| = k_1, \ldots, |S_n| = k_n ; \; k_1 + k_2 + \ldots + k_n = n$$

Here I denotes the set of all items. For all S_1 to S_n each denotes one set of items. For a particular set S_1, from i_{11} to i_{1k_1} each one denotes one item that is classified into S_1.

The second step is to define a similarity named the similarity in a particular set. Different from the similarity computed among all items, the similarity in a particular set is computed only among those items in this set that have been rated by both a and u. At the same time, we should figure out the two parameters a and b attached to this set. These two parameters are also computed among those items in this particular set with LSE method.

The last step is to predict the rating of the active user. When we predict the user a's rating through item i, first we compute the similarity among all items between a and the user who has rated item i, and the two parameters defined in Section 3.1. Second, we find out which set item i belongs to. Once we find this set, we compute the similarity in it and the two parameters attached to it as well. Finally the two similarities are compared. If the value of the similarity in the particular set is higher than the one computed among all items, the similarity in this set and the attached two parameters are retained for subsequent use, otherwise, we apply the typical similarity among all items and the two parameters. Therefore we have applied the classification information without losing the information of the similarities among all items.

3.3 Combining a User-Based Classified Regression and an Item-Based Filtering:

Now we address the problem of the sparse rating matrix. In order to solve this we combine a user-based classified regression and an item-based filtering. As Deng et.al [1] points out: making this combination is a way to lessen the negative effect of the sparsity.

3.3.1 Item-Based Collaborative Filtering:

Unlike the user-based collaborative filtering algorithm, the item-based approach fixes the attention on the similarity of items according to the ratings on them. This approach looks into the set of items the target user has rated and computes the similarity between them and the target item i. Then the most similar items are selected to form a list I.

Since the formula in Section 2.1 is used to give the prediction in a typical user-based filtering when the correlation-based similarity is applied to measure the similarity between users, we change the formula that is used to predict ratings in item-based filtering [2] to a new formula to give a pseudo-score when we use the correlation-based similarity to measure the similarity between items. The new formula is defined as:

$$p_{a,i} = \bar{r}_i + \frac{\sum_{j \in I} sim(i,j) \times (r_{a,j} - \bar{r}_j)}{\sum_{j \in I} |sim(i,j)|}$$

Here $sim(i,j)$ denotes the similarity between item i and item j. $r_{a,j}$ denotes the rating of user a on item j, $\bar{r}_i$ denotes the average rating on item i by the users who have rated it.

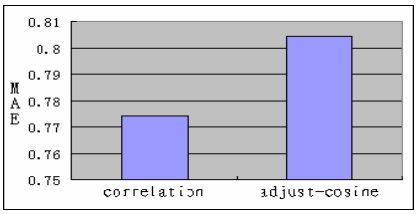

Figure 1. Comparison between the Item-Based Adjust-cosine Similarity and the Item-Based Correlation Similarity after the predictive formula is changed.

Figure 1 shows that once the formula is changed, the MAE of the correlation-based similarity becomes lower than that of the

adjusted cosine similarity which had a lowest MAE in Sarwar et.al's work [2]. So we apply the new formula and the correlation-based similarity of items to produce a pseudo-score. In addition the item-based approach in Section 4.3.3 is also referred to the algorithm of the correlation-based similarity only.

It should be noted that all the results of Figure 1 were run at the value 0.1 of the correlation-based similarity threshold as MAE increased with the threshold, and the lowest MAE consistently resulted with a threshold value of 0.1,

3.3.2 The Combining of the Two Approaches:

Following the algorithm as described by Deng et.al [1], when we compute the similarity of two users, we find out the union set of rated items, in which each item is rated by user a or user u. Then we fill out the un-rated items in this set with pseudo-scores generated by the item-based collaborative filtering. Hence the union set becomes the set of items rated by both users.

Subsequently, we compute each similarity between users in such a union set and utilize the user-based classified regression later. As a result, the regression model should become more accurate, and the results should be better.

Note: To match the similarity in a particular set and the similarity among all items, each pair of users has two union sets. One is extracted from all items; the other is extracted from a particular set where the target item belongs.

4. Experimental Results

4.1 Experimental Data

We used the data from MovieLens recommender system. MovieLens is a web-based research recommender system. The data was collected through the MovieLens web site. The full data set has 100000 ratings by 943 users on 1682 items. Each user has rated at least 20 movies.

The 1682 items, in this case 'movies', are classified into 18 genres. Some movies belong to more than one genre. Such movies were dropped from this study. Therefore the data set used has 30052 ratings on 831 items and 942 users.

The data were divided into a training set and a test set. The training set has 24144 ratings, 80.34% of the full data set. The same factor-sparsity level was used as defined in [2]. The sparsity level of therefore is 1-30052/942*831, which is 0.9616.

4.2 Evaluation Metrics

We used MAE to evaluate our prediction experiment. Mean Absolute Error (MAE) between ratings and predictions is a widely used metric to evaluate the quality of recommender system [2]. Let $\langle p_i, r_i \rangle$ represent the rating-prediction pair, p_i is the prediction of one rating, r_i is the actual user rating.

$$MAE = \sum_{i=1}^{N} |p_i - r_i| / N$$

The lower the MAE is, the more accurately the recommendation engine predicts the ratings.

4.3 Results and Discussions

All of the experiments made use of the method of the correlation threshold to select the most similar users, the value of which was increased from 0.1 to 0.9(A negative number in a correlation-based similarity means that interests are in some sense opposite).

4.3.1 Determination of Values of Some Threshold:

Since the lowest MAE value consistently resulted with a threshold value of 0.1, the threshold value for similar items was set equal to 0.1 when a combination of the item-based filtering and the user-based classify regression was made. The value of threshold T_2, which is described in Section 3.1 is settled equal to 0.5 in this case as it was presumed that the value of the similarity between users is high when it exceeded 0.5.

4.3.2 Dealing with the Problem of Overflow:

During the application of the proposed algorithm, the prediction: $p_{a,i} = \bar{r}_a - b\bar{r}_u + br_{u,i}$ might exceed 5 or fall below 1. If this should occur, and if $p_{a,i}$ >5 we set $p_{a,i}$ equal to 5, else if $p_{a,i}$ falls below 1 we set $p_{a,i}$ equal to 1 (the values of the ratings vary from 1 to 5).Experimental Results and Discussion:

4.3.3 Experimental Results and Discussion:

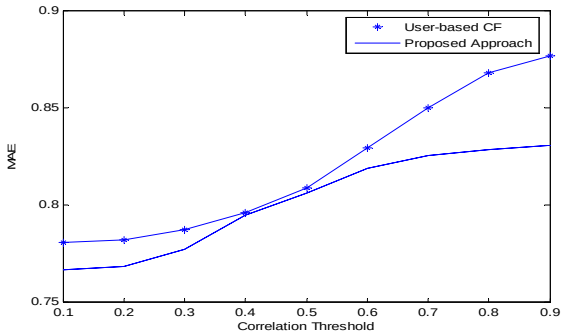

Figure 2. Comparison between User-based CF and Proposed approach.

Figure 2 demonstrates that the proposed approach outperforms the typical user-based CF as the MAE of the new algorithm is much lower than that of the typical approach when the correlation threshold T_1 varies from 0.1 to 0.4 as well as for the range 0.5 to 0.9. For example, at T_1=0.1, the new approach shows a MAE of 0.766632 and the typical user-based approach shows a MAE of 0.780989. Similarly, at T_1=0.8, the new approach shows a MAE of 0.828749 and the typical one shows a MAE of 0.868157. However, when we set the value of T_1 to 0.4 or to 0.5, although the MAE of the new approach is lower, the advantage is not significant.

In the first situation when the values of the correlation threshold T_1 are held below 0.4, they are also below T_2 (0.5), as a result, the approach takes both the benefit of regression and of the typical approach: as for a rating of user a on item i, when $sim(a,u) < T_2$ but> T_1, part of the typical formula is applied; when $sim(a,u) >= T_2$, the regression model is utilized. Having the benefit of both of the two methods, the new approach certainly performs better in this situation. In the second situation, when $T_1 > T_2$ (0.5), since the values of the similarities of the users

are high, the new approach purely applies the regression model to make a prediction and outperforms the typical approach significantly. In the third situation, when T_1 is set to 0.4 or 0.5, the number of neighbors whose similarities with the active user are below 0.5 is much fewer, the element of typical formula is seldom used and the regression method can not perform well under such a low value of similarity. In such a case, the new approach performs only slightly better.

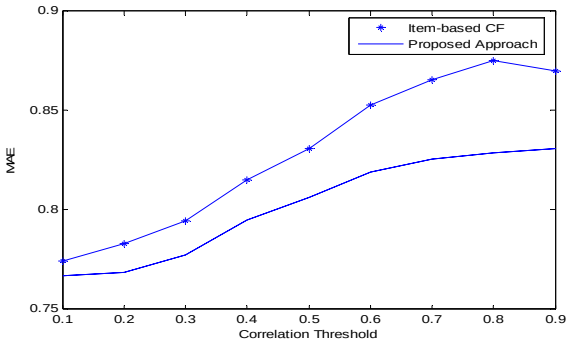

Figure 3. Comparison between Item-based CF and Proposed approach.

Figure 3 shows: the MAE of the new approach is lower than that of the item-based approach at all values of T_1 .

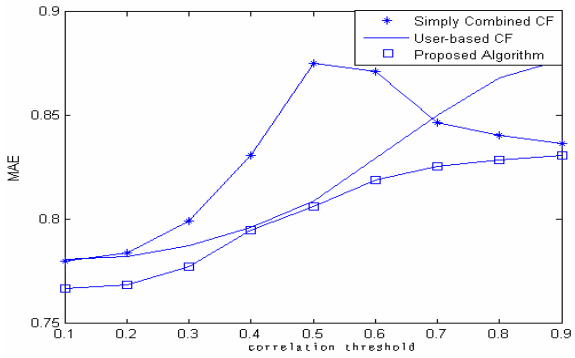

Figure 4. Comparison among User-based CF, Simply Combined CF and Proposed approach.

Figure 2 and Figure 3 do prove that the new approach proposed by this paper is effective for improving the quality of prediction. However, since we also apply a method of combination in our algorithm, the advantage might be the result of combination and not the result of classified regression. To test this possibility, it is necessary to compare our algorithm and an algorithm that is a simple combination of the item-based collaborative filtering and the user-based collaborative filtering. The results are shown in Figure 4. Figure 4 clearly shows that our approach performs better than the simply combined CF, which means the classified regression is an effective component. Furthermore, though the simply combined CF does not perform better than our approach, it performs better than the typical user-based CF when the values of the correlation threshold are high, which means combination is indeed an effective way to reduce the negative effect of the sparse rating matrix in this case.

5. CONCLUSION

In conclusion, using classified regression helps to avoid the following problems: a missing important parameter in the typical collaborative algorithm caused by the strong similarities between the users who give very different ratings; and the inadequate use of the classification information. In addition, the incorporation of the item-based algorithm lessens the negative effect of the sparse user-item rating matrix[1].

However our experiments were run only on the movie items that each belongs to one specific genre. As for those movies (items) that each belongs to more than one genre (set of items), how to utilize our algorithm remains an unsolved problem.

6. ACKNOWLEDGMENTS

The authors wish to acknowledge Movielens that supplied the data from its web-based research recommender system.

7. REFERENCES

[1] Ailin Deng, Yangyong Zhu and Bole Shi, “A Collaborative Filtering Recommendation Algorithm Based on Item Rating Prediction”, 2003 Journal of Software , vol.14, No.9.

[2] Badrul Sarwar etc., “Item-Based Collaborative Filtering Recommendation Algorithms”, n Proceedings of the Tenth IWWWC, 2001, pages: 285-295.

[3] J. Jiang, M. Berry, J. Donato, G. Ostrouchov, and N. Grady, “Mining consumer product data via latent semantic indexing,” Intelligent Data Analysis, vol. 3, pp. 377–398, 1999.

[4] L. Ungar and D. Foster, “Clustering methods for collaborative filtering,” in Recommender Systems - Papers from the AAAI Workshop, Madison,WI, July 1998.

[5] Pazzani, M. and Billsus, D.: Learning and Revising User Profiles: The identification of interesting Web sites. *Machine Learning* 27, 313-331 (1997).

[6] Prof. Dr. Erich J. Neuhold. Personalization and User profiling & Recommender Systems. WI/IM Information management Proseminar SS 2003 LVA_Nr.:4002988

[7] Resnick, P., Neophytos, I., Mitesh, S., Bergstrom, P., and Riedl, J.” GroupLens: An open architecture for collaborative filtering of netnews.” In Proceedings of CSCW94, 175-186, Chapel Hill, Addison-Wesley (1994).

[8] Sarwar, B. M., Konstan, J. A., Borchers, A., Herlocker, J., Miller, B., and Riedl, J. (1998). Using Filtering Agents to Improve Prediction Quality in the GroupLens Reseach Collaborative Filtering System. In Proceedings of CSCW ’98, Seattle, WA.

[9] YU Li etc., “Research on personalized recommendation algorithm for user’s multiple interests”, Computer Integrated Manufacturing Systems,Vol.10, No.12,2004.

[1] This work is partially supported by China NSFC (National Natural Science Foundation of China) Research Programs, Index Number: 60573092.