An Effective Algorithm for Dimensional Reduction in Collaborative Filtering

Fengrong Gao[1], Chunxiao Xing[1], and Yong Zhao[2]

[1] Research Institute of Information Technology,
Tsinghua University, Beijing 100084
{gaofengrong,xingcx}@mail.tsinghua.edu.cn
[2] Department of Computer Science and Technology,
Tsinghua University, Beijing 100084
Zhaoyong04@mails.tsinghua.edu.cn

Abstract. It is necessary to provide personalized information service for users through the enormous volume of information on the web. Collaborative filtering is the most successful recommender system technology to date and is used in many domains. Unfortunately collaborative filtering is limited by the high dimensionality and sparsity of user-item rating matrix. In this paper, we propose a new method for applying semantic classification to collaborative filtering. Experimental results show the high efficiency and performance of our approach, compared with tradition collaborative filtering algorithm and collaborative filtering using K-means clustering algorithm.

Keywords: Collaborative filtering, dimensionality reduction, semantic classification.

1 Introduction

With the rapid growth of the Web, people have to spend more and more time to search what they need. To deal with this difficulty, recommender systems have been developed to provide different services for different users. Collaborative filtering is one of the most successful recommender system technologies to date and is used in many domains. It works by collecting user feedback in the form of ratings for items in a given domain and seeks similarities between user rating histories. Collaborative filtering does not consider the content of items, so it supports for filtering items whose content is not easily analyzed with computers such as video, audio, restaurants, etc. For it provides recommendations based on user's ratings to items, most users seem less reluctant to provide item-rating information. So the user-item rating matrix is very sparse. Moreover, along with users and items increase, the matrix will be very high dimensional. Both the sparsity and the high dimensionality lower the accuracy and efficiency of recommendations.

In this paper, we propose an effective method to overcome the drawbacks in collaborative filtering. Our approach uses semantic classification to divide original user-item rating matrix into several low-dimensional dense user-item rating matrices, then use low-dimensional matrices to provide recommendations.

D.H.-L. Goh et al. (Eds.): ICADL 2007, LNCS 4822, pp. 75–84, 2007.

2 Related Work

The term "collaborative filtering" was first coined by Goldberg et al on the Tapestry email system[1]. A variety of collaborative filtering algorithms have been designed and deployed henceforth. The GroupLens[2~4] system is one of the first automated collaborative filtering systems to apply a statistical collaborative filtering to the problem of Usenet news overload. It identifies advisors based on the Pearson correlation of voting history between pairs of users. This method is usually called "correlation-based collaborative filtering". It is the most popular among all the algorithms and represents in many researches.

Besides correlation-based methods, some model-based algorithms [5~7] appeared which adopt data mining techniques such as clustering, classification, and Bayesian networks. These methods pre-compute a model based on a training data set and then use the model for predictions. Once the clustering process is complete, the efficiency of the recommendations can be very good, since the size of the data is much smaller than original user-item rating matrix. In section 5 besides comparing our approach to traditional collaborative filtering, we also experiment by comparing our approach to a collaborative filtering algorithm using clustering technique.

To reduce the dimensionality of data and tackle the sparsity problem, Singular value decomposition [8] is used to produce a low-dimensional representation of the original user-item space and a list of recommendations will be generated using low-dimensional space. Our method uses the idea "dimensionality reduction" of this paper. But our approach of dimensionality reduction is quite different from SVD.

Popescul et al [9] presented probabilistic mixture models for recommending items based on collaborative and content-based evidence merged in a unified manner. The model builds on two-way co-occurrence models and collaborative filtering. It incorporates three-way co-occurrence data (users, items and item content) presuming that users are interested in a set of latent topics which in turn generate both items and item content information. The approach of building probabilistic model mixing content data with collaborative data is also proposed in Ref. [10~11].

Breese et al [12] performed an empirical analysis of several collaborative filtering algorithms. Experiments show that the recommendations are correlative with the nature of the dataset, nature of the application, and the availability of votes with which to make predictions.

3 Semantic Classification-Based Collaborative Filtering

3.1 Problem Description

User ratings for items are described in traditional collaborative filtering algorithm as a matrix:

$$R_{m\times n}=\begin{bmatrix} r_{11} & \cdots & r_{1j} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ r_{a1} & \cdots & r_{aj} & \cdots & r_{an} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mj} & \cdots & r_{mn} \end{bmatrix}_{m\times n} \qquad (1)$$

where r_{aj} denotes the score of item j rated by user a. If user a had not rated item j, r_{aj} =0. m denotes the total number of users, and n denotes the total number of items. The prediction of user a to an unseen item j, i.e. p_{aj} is done based on the average ratings of user a and a weighted sum of co-rated items between user a and all his similar users based on $R_{m\times n}$.

$$p_{aj} = \overline{r_a} + k\sum_{i=1}^{m_a} w(a,i)(r_{ij} - \overline{r_i}) \tag{2}$$

where m_a is the number of users, similar to user *a*, who have rated item *j*. The weight $w(a,i)$ expresses the similarity between user a and user *i*. *k* is a normalizing factor such that the absolute values of the weights sum to unity.

For a target user (in subsequent sections, user *a* denotes the target user.), the collaborative system provides a list of unseen items descending ordered by predicted values.

We have known that both the total number of items and the total number of users are very large. And generally each user will only have rated a small percentage of the total number of items. So $R_{m\times n}$ in equation (1) is high dimensional sparse. The weakness of the original matrix led us to explore alternate methods for low dimensional representation.

3.2 Dimension Reduction Based on Semantic Classification

Domain-specific classification is used to organize web resources. Each item can be assigned to one or more classifications. For example, in the domain of movies, every movie can be classified according to the attribute "genre" of each item (the values of genre include Action, Adventure, Drama, and so on). In the domain of books, an attribute "category" of items is used to classify books. We use domain-specific classification information to partition original user-item rating matrix for dimensionality reduction. Each item belongs to one or more classes. Each class has at least one item. The following presents main process of dimensionality reduction method.

Step 1: Reduce the items in original matrix $R_{m\times n}$.

For each class *p*, find all the items belong to *p* from $R_{m\times n}$, $R_{m\times n}$ is converted to $R^p_{m\times n_p}$:

$$R^p_{m\times n_p} = \begin{bmatrix} r_{11} & \cdots & r_{1j} & \cdots & r_{1n_p} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ r_{a1} & \cdots & r_{aj} & \cdots & r_{an_p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mj} & \cdots & r_{mn_p} \end{bmatrix}_{m\times n_p} \tag{3}$$

where n_p is the total number of items that belong to class *p*. Experiments show that $n_p << n$. Thus from $R_{m\times n}$ to $R^p_{m\times n_p}$ ($p = 1,\cdots,C$), the reduction of dimension "item" is successfully completed. If an item belongs to one more classes, it will be assigned to each of the classes it belongs to. For example, the genres of the movie “Toy Story” are “Animation”, “Children’s” and “Comedy”. So the movie is assigned to each of the three genres.

Step 2: Reduce users in every $R^{p}_{m\times n_p}$.

In $R^{p}_{m\times n_p}$, if the number of items rated by user a is less than a threshold ϖ , the system considers user a is uninterested in this class of movies. So user a will be removed from the matrix $R^{p}_{m\times n_p}$. Then $R^{p}_{m\times n_p}$ is converted to $R^{p}_{m_p\times n_p}$:

$$R^{p}_{m_p\times n_p} = \begin{bmatrix} r_{11} & \cdots & r_{1j} & \cdots & r_{1n_p} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ r_{a1} & \cdots & r_{aj} & \cdots & r_{an_p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{m_p1} & \cdots & r_{m_pj} & \cdots & r_{m_pn_p} \end{bmatrix}_{m_p\times n_p} \tag{4}$$

where m_p is the total number of users within class *p*. Also our experiments show that $m_p << m$. We completed the reduction of dimension “user” from $R^{p}_{m\times n_p}$ to $R^{p}_{m_p\times n_p}$ for each *p*. In the next section we present how to generate recommendations with $R^{p}_{m_p\times n_p}$ (*p=1*, …, *C*) for user *a*.

3.3 Modeling User Preferences

Traditional collaborative filtering algorithm represents user preferences as a list of <item, rating>. In semantic classification-based collaborative filtering, we model user preferences of a particular user (e.g. user a) as

$$U^{(a)} = \{R_{ap} \mid p = 1, \cdots, C\} \tag{5}$$

where R_{ap} denotes the set of <item, score> whose items belong to class *p*, and they had been rated by user *a*. If user *a* rated no item within class *p*, R_{ap} will be NULL.

Example: Table 1 shows user 1 rated movies. Table 2 shows the relations between items and classes. In the cell of table 2, “1” represents the item belongs to the Class. From the data of table 1 and table 2, we model user preferences for user 1:

$U^{(1)} = \{R_{11}, R_{12}, R_{13}, R_{14}, R_{15}, R_{16}, R_{17}, R_{18}, R_{19}\}$, where
$R_{11} = \{<2{,}3>, <4{,}3>\}$, $R_{12} = \{<2{,}3>\}$, $R_{13} = \{<1{,}5>\}$, $R_{14} = \{<1{,}5>, <5{,}3>, <8{,}1>\}$,
$R_{15} = \{<1{,}5>, <4{,}3>, <8{,}1>\}$, $R_{16} = \{5{,}3\}$, $R_{17} = \phi$, $R_{18} = \{<4{,}3>, <7{,}4>, <8{,}1>, <9{,}5>\}$,
$R_{19} = \{<3{,}4>, <11{,}2>\}$.

3.4 Computing Similarity

There are many similarity measures including vector similarity, Pearson correlation coefficient, entropy-based uncertainty measure, and mean square difference to weight all users with respect to similarity with the target user. Researchers found that Pearson

Table 1. Examples of user rated items

User	Item	Rating
1	1	5
1	2	3
1	3	4
1	4	3
1	5	3
1	7	4
1	8	1
1	9	5
1	11	2

Table 2. Examples of item-class matrix

Ite m	Class								
	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9
1			1	1	1				
2	1	1							
3									1
4	1				1			1	
5				1		1			
6								1	
7				1	1			1	
8								1	
9						1			1
11	1				1			1	

correlation performs better than other similarity measures. In our algorithm, we use Pearson correlation coefficient to compute the similarity weight between user a and user i ($i=1,\cdots,m_p$) within class p ($p=1,\cdots,C$), i.e. $w^p(a,i)$:

$$w^p(a,i)=\frac{\sum_j (r_{aj}-\overline{r_{ap}})(r_{ij}-\overline{r_{ip}})}{\sqrt{\sum_j (r_{aj}-\overline{r_{ap}})^2 \sum_j (r_{ij}-\overline{r_{ip}})^2}} \tag{6}$$

where $j\in I_{ap}\cap I_{ip}$, I_{ap} is the set of items rated by user a within class p , I_{ip} is the set of items rated by user i within class p . Thus the summations over j are over the items for which both user a and user i have given ratings within class p . $\overline{r_{ap}}$ is the average rating of user a for items, which belong to class p .

$$\overline{r_{ap}}=\frac{1}{|I_{ap}|}\sum_{j\in I_{ap}} r_{aj} \tag{7}$$

If the total number of items rated by both user *a* and user *i* is under a threshold, it is considered that the two users have no common preferences for class *p*, i.e. $w^p(a,i)=0$.

4 Recommendations Generation

4.1 Predicting Unseen Items

After the similarity weight is computed between user a and each user in class p , the prediction of unseen items can be computed using the following equation:

$$p_{aj}^p=\overline{r_{ap}}+k\sum_{i=1}^{m_{ap}} w^p(a,i)(r_{ij}-\overline{r_{ip}}) \tag{8}$$

where p_{aj}^{p} represents the prediction for the target user a for item j within class p. m_{ap} is the number of users, similar to user a, which have rated item j in class p. Synthetically, we compute the prediction for user a for unseen item j whenever the item belongs to more than one class or belongs to just one class.

If item j belongs to more than one class, then for user a the prediction for item j, i.e. p_{aj} is assigned to the maximum value among the classes that item j belongs to.

$$p_{aj} = \max_{j \in p} p_{aj}^{p} \tag{9}$$

If item j belongs to just one class, then $p_{aj} = p_{aj}^{p}$.

4.2 Recommendation for a List of Items

In semantic classification-based collaborative filtering, unseen items for user a are descending sorted by the predicted value. Then the algorithm provides a list of highest predicted items for user a. This is commonly known as top-N recommendation. If user a prefers C semantic classes, for each class p about $\lceil N/C \rceil$ highest predicted items are selected for the recommendation. N is the total number of recommended items for user a, and C is the total number of semantic classes in user preference model of user a.

5 Experiments and Evaluation

In order to evaluate the quality of semantic classification-based collaborative filtering, we experimentally evaluate the performance among semantic classification-based collaborative filtering (SCF), traditional collaborative filtering (Tradtional CF)and collaborative filtering using K-means clustering (KCF). Some researchers proposed that data mining techniques could be applied to collaborative filtering systems. One of popular clustering algorithms used in model-based collaborative filtering is K-means. We use K-means clustering algorithm to partition the train data subset into K clusters. Then predict each item in test data subset, determining which cluster the item belongs to. In the algorithm, we assign the value K equals to C, i.e. the number of clusters equals to the number of genres.

5.1 Data Set

In our experiments, we use MovieLens data set contains 100,000 ratings of 1682 movies rated by 943 users. There are 18 genres in the data set, and each movie belongs to one or more genres. The data set is divided into 80% training set and 20% test set. In our approach, "genre" is used as semantic classification to partition user-item matrix of the training set.

5.2 Evaluation Metrics

The effectiveness of collaborative filtering algorithms has traditionally been measured by statistical accuracy and decision-support accuracy metrics. Statistical accuracy metrics evaluate the accuracy of a system by comparing the numerical recommendation scores against the actual user ratings for the user-item pairs in the test dataset. We use Mean Absolute Error (MAE), a statistical accuracy metrics, to report prediction experiments for it is most commonly used and easy to understand

$$MAE = \frac{\sum_{j=1}^{N} \left| p_j - r_j \right|}{N} \tag{10}$$

where $\{p_1,\dots, p_N\}$ are predicted values in the target set, and $\{r_1,\dots,r_N\}$ are all the real values for the same items. N is the total number of items in the target set.

Decision support accuracy metrics evaluate how effective a prediction engine is at helping a user select high-quality items from the set of all items. The ROC (receiver operating characteristic) sensitivity is an example of decision-support accuracy metrics. The metric indicates how effectively the system can steer users towards highly-rated items and away from low-rated ones. We use ROC-4 measure as the evaluation metric.

$$ROC-4 = \sum_{j=1}^{N} w_j \Big/ \sum_{j=1}^{N} u_j \tag{11}$$

where $w_j = \begin{cases} 1 & r_j \ge 4 \text{ and } p_j \ge 4 \\ 0 & otherwise \end{cases}, \quad u_j = \begin{cases} 1 & p_j \ge 4 \\ 0 & otherwise \end{cases}$

5.3 Experimental Results

5.3.1 Dimensionality reduction

Table 3 shows the number of items that each genre has. From Table 3 and Fig. 1, we can see the largest number is 725, 43.1 percent of original 1682 items, and the minimum number is 22, 1.3 percent of the original. The total number of items grouped by genres are larger than 1682 because many items belong to more than one genre.

Table 3. Number of items in semantic classification

Genre	Number of Items	Genre	Number of Items
1	251	10	24
2	135	11	92
3	42	12	56
4	122	13	61
5	505	14	247
6	109	15	101
7	50	16	251
8	725	17	71
9	22	18	27

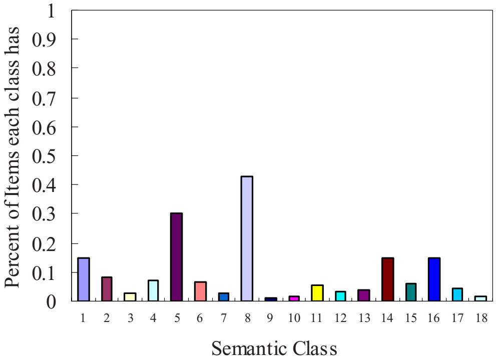

Fig. 1. Percent of items in semantic classification

As the same, users are partitioned into semantic classes. Table 4 shows the number of users that each genre has, and Fig. 2 shows the percent of users. We can see the difference of the number in each genre is very large. Some genres such as genre 8 interest almost all the users, and few people like genre 9. The percent averages at 0.38.

Table 4. Number of users in semantic classification

Genre	Number of Users	Genre	Number of Users
1	828	10	88
2	470	11	75
3	39	12	74
4	125	13	141
5	873	14	836
6	216	15	481
7	30	16	832
8	933	17	314
9	25	18	38

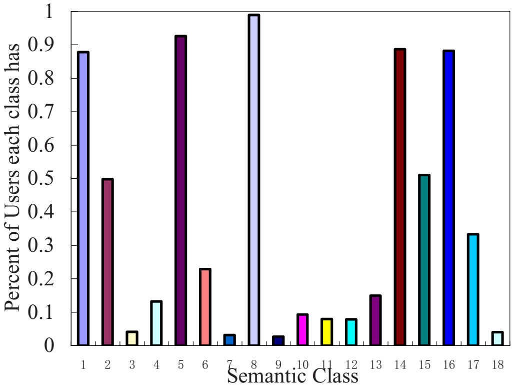

Fig. 2. Percent of users in semantic classification

5.3.2 Statistical Accuracy

When a user rates an item, the number of nearest neighbors affects the MAE for the algorithm. Figure 3 shows the dependence of the MAE on the number of nearest neighbors. SCF algorithm performs better than both traditional CF and KCF. On average, the SCF algorithm performs 7.3% better than the traditional CF algorithm, and it performs little difference between traditional CF algorithm and KCF algorithm.

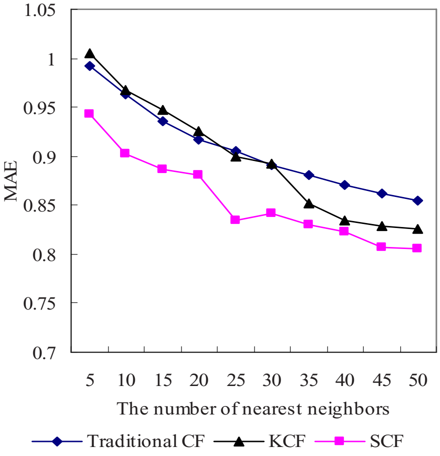

Fig. 3. MAE results

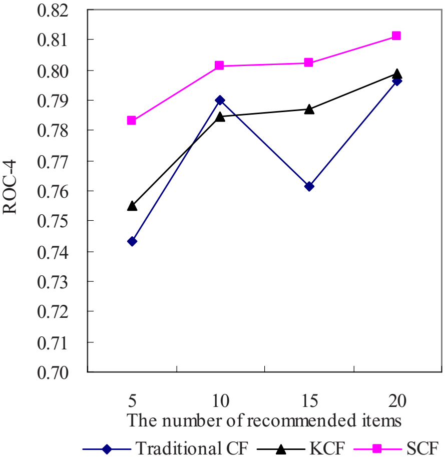

Fig. 4. ROC results

6 Conclusions and Future Work

In this paper, according to the attribute of items, we proposed a collaborative filtering based on semantic classification. Experiments shows it works well in dimensionality reduction and performs better quality of recommendations in the domain of movie recommendations. In the paper we suppose one genre is independent of another. But in many domain-specific classifications, the classes are correlative. For example, in the domain of book recommendations, the attribute "category" of books can be used as semantic classification. Different from movies, the category has multi-levels. Categories can be subdivided into sub-categories, and sub-categories are subdivided into sub-subcategories, and so on. We will apply other techniques such as association analysis to expand semantic classification-based collaborative filtering in the future research.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No. 60473078 and National Science and Technology Support Plan of China (No. 2006BAH02A12). We also thank GroupLens research group for allowing us to use the data set they had developed.

References

1. Goldberg, D., Nichols, D., Oki, B., Terru, D.: Using collaborative filtering to weave an information tapestry. Communications of the ACM 35(12), 61–70 (1992)
2. Resnick, P., Iacovou, N., Sushak, M., et al.: GroupLens: An open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 Computer Supported Cooperative Work Conference, ACM Press, New York (1994)
3. Sarwar, B.M., Konstan, J.A., Borchers, A., et al.: Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. In: Proceedings of 1998 Conference on Computer Supported Collaborative Work (November 1998)
4. Rashid, M., Lam, K., Karypis, G., Riedl, J.: ClustKNN: A Highly Scalable Hybrid Model- & Memory-Based CF Algorithm. In: proceedings of WEBKDD 2006, Philadelphia, Pennsylvania, USA (August 20, 2006)
5. Xue, G.R., Lin, C.X., Yang, Q., et al.: Scalable collaborative filtering using cluster-based smoothing. In: Proceedings of SIGIR 2005, Salvador, Brazil, August 15-19, pp. 114–121 (2005)
6. Aggarwal, C.C., Yu, P.S.: Data mining techniques for personalization. IEEE Bulletin of the Technical Committee on Data Engineering - Special Issue on Database Technology in E-Commerce 23(1), 4–9 (2000)
7. Kohrs, A., Merialdo, B.: Clustering for collaborative filtering applications. In: Computational Intelligence for Modeling, Control & Automation (CIMCA 1999), Vienna, IOS Press, Amsterdam (1999)
8. Zhang, S., Wang, W.H., Ford, J., et al.: Using singular value decomposition approximation for collaborative filtering. In: Proceedings of the Seventh IEEE International Conference on E-Commerce Technology (CEC 2005), pp. 257–264 (2005)

9. Popescul, A., Ungar, L.H., Pennock, D.M., et al.: Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In: Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI-2001), pp. 437–444. Morgan Kaufmann, San Francisco (2001)
10. Basilico, J., Hofmann, T.: A joint framework for collaborative and content filtering. In: Proceedings of SIGIR, pp. 550–551 (2004)
11. Basilico, J., Hofmann, T.: Unifying collaborative and content-based filtering. In: Proceedings of the twenty-first international conference on Machine learning (ICML 2004) (2004)
12. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithm for collaborative filtering. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, Morgan Kaufmann, San Francisco (1998)